High–Fidelity Link Shaping

Roman Chertov
roman@santabarbaralabs.com
Santa Barbara Labs

1. INTRODUCTION

Network testbeds are an invaluable tool for conducting experiments with real hardware and software in a controlled and easily manageable environment. Typically, testbeds employ an emulation component to control the delay and bandwidth of experimental links. On most testbeds, the nodes are only a few milli-seconds away from each other and the link speed is either 100 or 1000 Gbps. Such an arrangement is acceptable for LAN only experiments; however, networks, which have long distance links, experience much higher delays. For instance, networks with Geostationary Earth Orbit (GEO) satellite links can experience a delay of 125 ms per hop. Additionally, networks with satellite links can have a significant variation of speed between the links. Hence, providing a variable delay and bandwidth is paramount, when conducting Wide Area Network (WAN) experiments.

Link shaping over Ethernet can be provided with three methods: rate limiting at the network node, transparent bridging, and IEEE 802.3x flow–control. In this report, we will investigate which of the link shaping approaches will provide the best performance without inducing emulation artifacts.

Transparent bridging is a popular link shaping approach. In such a case, an extra delay node (link emulator) is used to pass Ethernet frames from one node to another, while performing some rate scheme. When the delay node receives a packet, it first delays the packet by a specified amount, then forwards the packet to a rate limiter. If the rate on the link exceeds the specified bandwidth, the delay node first queues up frames to ensure that the egress interface maintains the desired rate. If the ingress rate does not decrease, packets get dropped as the queue overflows. From the point of view of non–delay network nodes, the packets are either lost or delayed. Examples of such link emulators are DummyNet [Rizzo], Click modular router [Kohler et al. 2000], and *LinkEm*. Dummynet is a very popular link emulator and is frequently used in networking research. Another popular link emulator is a Click modular router which can be configured to provide link shaping [Kohler et al. 2000; Agarwal et al. 2005]. Additionally, Click can be configured to utilized multiple CPUs to ensure that flows in one direction do not affect the flows in the other direction. The major drawback of link emulators is the fact that they require an additional node, and hence an increase in configuration complexity, as transparent bridges have to be created.

Another way to provide the desired link characteristics is to rate–limit the output interface itself. For example, Linux supports shaping via Class Based Queuing (CBQ) [Hubert and et al 2002] and Cisco routers provide a *rate-limit* command [Cisco Systems 2008]. The obvious drawback of this approach is the fact, that the node must provide the rate limiting facility. Additionally, providing rate limiting taxes the system resources. The rate limiting functionality does not imply that the packet delay functionality exists, meaning that it might be impossible to create long propagation delays.

Finally, the IEEE 802.3x standard, which specifies flow control for Ethernet, can be used for rate limiting and delaying a link. The IEEE standard specifies that control Ethernet pause frames can be sent out to the sender from the receiver to temporarily halt transmissions. Hence, the rate of the pause frames and their duration dictate the resulting bandwidth and delay. Obviously, the pause frame approach relies on the fact that the transmitting hardware is capable of dealing with micro–second precision timers and can support flow–control.

The remainder of this report is organized as follows. Section 2 provides the details of our experimental tools and the testbed. Section 3 provides the calibration results of our measurement tool. Section 4 discusses the results obtained with various link shaping methods. Finally, Section 6 concludes this report and summarizes our findings.

2. EXPERIMENTAL SETUP

To test which link shaping approach will produce the least amount of emulation artifacts, we have set up a small testbed, as shown in Figure 1. The testbed has one Cisco 2851 router and two Dell SMP PCs. One PC acts as a traffic source/sink and has two quad–core 1.86 GHz Intel Xeon processors. The other PC acts as a bridge/packet counter and has two dual–core AMD Opteron 2212 processors. Additionally, the PC with the AMD processors has the node interleaving memory option enabled. The equipment is directly connected as shown in Figure 1, using Gigabit Ethernet. In the experiments, we were interested in shaping the output link of the Cisco router, using several methods in order to make a direct comparison.

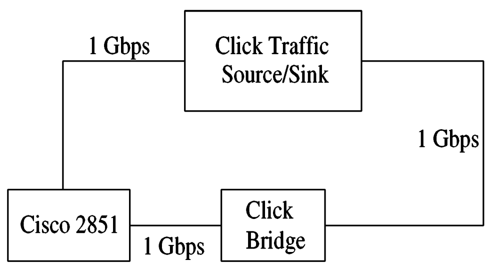

Fig. 1. Testbed topology with a commercial Cisco router and SMP Dell PCs.

2.1 Traffic Generator/Analyzer

To gauge the effectiveness of each method, we have used a high–precision packet generator/analyzer. The tool is called a Black Box Profiler (BBP) [Chertov et al. 2007]. The layout of the BBP is demonstrated in Figure 2. BBP is configured to act as a traffic source and as a traffic sink. The device driver was also modified to embed timestamps into packets as they are sent and received. The traffic flow over the network was configured such that the packet path would originate and finally terminate at the BBP. As the packets originate and terminate at the BBP node, no clock synchronization is required to obtain the packet time in the network with microsecond–level precision. Additionally, it is possible to compute inter–packet gap as the packets leave and enter the system.

The ability of BBP to compute the delay of packets in the network and inter–packet gaps allows us to gauge the performance of a link shaping scheme. For instance, if the overall

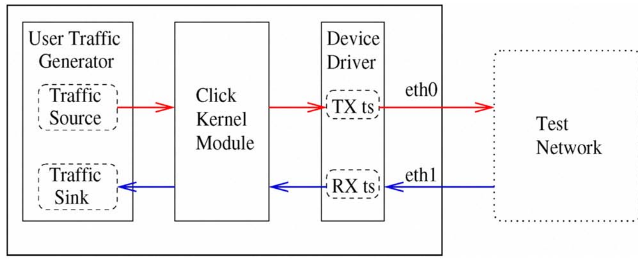

Fig. 2. Layout of the traffic generator/analyzer.

packet delay is much larger than expected or there is a significant variation, then the link shaping scheme is not performing well. In addition, an inter–packet gap is an indicator if the desired rate is achieved or not, and how much jitter an emulation tool induces. Ideally, the inter–packet gap is a constant for a Constant Bit Rate (CBR) flow.

In our experiments, we created a CBR stream of 1,000,000 UDP packets. Using same sized packets makes it easier to test the fidelity of a link emulation method, as there is no variance in packet delays due to different sizes. We also repeat the same experiment five times and report the results across all of the experimental runs.

2.2 Shaping Methods

In this Section we will provide an overview and configuration of the link shaping methods that we have evaluated.

2.2.1 *Transparent Delay Node.* A node acting as a transparent Ethernet bridge can delay and rate limit packets from one interface, before passing them to the other interface. Figure 3 demonstrates such functionality. In our experiments, we have configured the Click modular router to perform this task. We have also configured Click to utilize two CPUs, such that each port gets a dedicated CPU. Assigning a CPU per port removes the possibility of heavy congestion on one port affecting traffic on the other port. To provide link shaping in Click we have used the *LinkUnqueue* element.

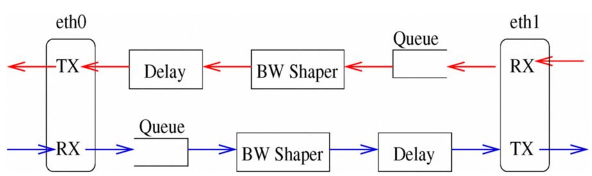

Fig. 3. Transparent bridge link shaper.

Besides using Click to perform the shaping, we were interested in comparing it to *LinkEm*, a link emulation utility. Unlike Click, *LinkEm* runs in user–level mode and bridges two network interfaces by relying on a raw socket, to capture Ethernet frames. Running a bridge in user–level can be detrimental to high speed packet forwarding, as packets have to be moved from kernel space to user space, hence sacrificing efficiency. However, since *LinkEm* provides a significant library of satellite link shaping models, we were interested in including it in the evaluation study.

The main drawback of a transparent delay node is the fact that an extra delay node is needed and that packet drops occur at a delay node and not at the Cisco router. In a heavy congestion scenario it does not matter if either the router of the delay node drops the packets. However, in a low–load but bursty scenario the delay node might drop more packets if it has smaller buffers than the router. Alternatively, the delay node can drop fewer packets if its buffers are larger than that of the router. This can possibly be mitigated by profiling a router first to ascertain its buffer sizes [Chertov et al. 2008].

2.2.2 *Router Rate Limiting.* The Cisco 2851 router has an ability to limit the output and input rates of an interface via the *rate-limit* command. The router does not have a feature to add delay to a packet. Hence, an additional delay node is still required. However, in this case, packets will get dropped on the router if the rate is exceeded. In our experiments, we have tested the router's ability to rate limit its output without using the delay node and also conducted experiments with the delay node. Testing without the delay node is necessary to ascertain the performance of a rate limiter, without any additional measurement noise.

2.2.3 *Ethernet Pause Frames.* The IEEE 802.3x standard proposes flow–control via control Ethernet frames. One node can send a frame to another node to pause its transmission for a specified duration of time. The pause frames specify a duration in 512 bit increments after which the transmission can resume. For example, if a receiver on 1 Gbps link sends a frame with a duration of 195, the sender must block for 100 micro–seconds[1]. Changes in duration and rate of the pause frames can be used to achieve a desired link bandwidth. Additionally, the pause duration can be set to the desired link delay. Using pause frames can induce burstiness into the packet flow, as the link operates in the on/off mode; however, on the aggregate, it might be possible to achieve the desired delay and bandwidth.

Pause frames require no changes in the router, and all packet drops will occur at the router. However, a pause frame generating node is required. Figure 4 demonstrates the layout of the generator we have used. The pause generator is labeled as Click Bridge in Figure 1.

Just as in the transparent delay node case, we have used Click and configured it to act as a bridge. In addition to bridging, we have created a pause frame generator that emits Ethernet pause frames of a specific quanta, at a given constant rate. The pause frame generator emits pause frame in the opposite direction of the measured packet flow, hence switching off the transmitting interface on the router for a specified duration.

3. CALIBRATION

Prior to conducting the experiments, we performed a calibration of BBP on our traffic generation PC. To perform calibration, we have connected two Intel Pro cards with a single cable and ran a series of tests, where we varied packet size and rate. The purpose of the test is to ascertain how much overhead the system adds to overall packet delay. Additionally, it was of interest to determine if the noise varied significantly or not. If the noise variation is low, then the noise can be treated as a constant delay.

In [Chertov et al. 2007; 2008], it was shown that BBP can perform well under a wide

[1] $\frac{195*512b}{1e9\ bps} \times \frac{1e6\ \mu s}{sec} = 100\ \mu s$

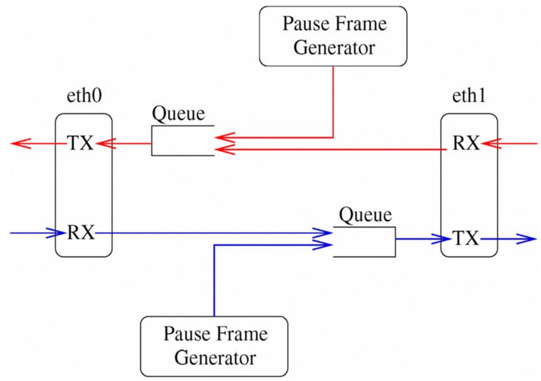

Fig. 4. Transparent bridge and pause frame generator.

variety of loads, without inducing a significant level of measurement noise. In addition, the previous results have shown that variance is low. However, since the experimental platform in the previous studies had different hardware, it was of interest to perform calibration on the new platform.

To conduct the calibration, we have created UDP packets such that resulting Ethernet frame sizes were 64-, 800-, and 1518-bytes. For each packet size, we have set the packet rate to 8000, 40000, 80000, 120000, and 200000. In cases where the resulting bandwidth is greater than the available bandwidth, we do not use such rate. Such an arrangement allows us to explore a wide variety of byte and packet rates.

Table I. NIC–to–NIC: inter–packet gaps (μs) for 64-, 800- and 1518-byte Ethernet frames

	64 bytes			800 bytes			1518 bytes		
Rate	10th	50th	90th	10th	50th	90th	10th	50th	90th
8000	123	125	127	123	125	127	123	125	127
40000	23	25	27	19	25	31	20	25	31
80000	7	12	18	8	12	16	11	12	14
120000	5	8	12	6	8	11			
200000	4	5	6						

Table II. NIC–to–NIC: packet delays (μs) for 64-, 800- and 1518-byte Ethernet frames

	64 bytes			800 bytes			1518 bytes		
Rate	10th	50th	90th	10th	50th	90th	10th	50th	90th
8000	6	7	8	14	15	16	22	23	24
40000	6	7	8	14	15	20	22	23	28
80000	6	7	10	15	16	17	24	29	34
120000	6	7	10	15	16	18			
200000	6	7	8						

Tables I and II respectively show the inter–packet gaps and packet delays in μs for 10th, 50th, and 90th percentiles. In the cases when the byte rate exceeded link capacity, we have left the entries in the tables blank.

Inter–packet gaps shown in Table I represent the time between packets as they enter BBP. Ideally, the difference between the 10th and 90th percentiles must be small, and the mean must be the same as the computed theoretical μs inter–packet gap for a given rate, which is computed as $\frac{1}{packet\_rate} \times \frac{1e6\ \mu s}{sec}$. The data indicates that the system performed exceptionally well, as the difference between the 10th and 90th percentiles is very small. Additionally, the 50th percentile for inter–packet gaps is the same as the theoretical value.

The delay, that packets experience when going over the cable between two network cards, should ideally be equal only to the transmission delay over a Gigabit link. Any additional delay besides that implies noise in the system. In the case of the BBP, the additional delay arises from the fact that packet delays include the transmission over the PCI-E bus and the network card. However, if the difference between the 10th and 90th percentiles is small, then the additional noise can be treated as a constant. The data in Table II implies that the additional delay in the system does not vary significantly and can be treated as a constant.

The results described above demonstrate that the BBP system provides an adequate precision level, to conduct evaluation studies of various link shaping techniques. Additionally, no packet loss was detected, meaning that the system is capable of maintaining very high packet rates with full packet logging loss–free.

4. EXPERIMENTAL RESULTS

In this section we will provide the experimental comparison of the link shaping methods.

4.1 Transparent Delay Node

As mentioned in Section 2.2.1 we have chosen two methods to emulate a link, *LinkEm* and Click. We have used the topology as shown in Figure 1, except without the Cisco router. Removing the Cisco router was necessary to measure the delay node only without the additional noise introduced by the router. For both tools, we have conducted two sets of tests. In the first test, the node performs no delay and purely bridges. The purpose of the test is to ascertain the level of additional packet delay due to bridging and to determine variability of inter–packet gaps due to the delay node. In the second test, the node is configured to provide a link delay of 125 ms and maintain a 1 Gbps bandwidth. We have chosen 125 ms delay, as it is a typical delay from ground to a Geostationary communications satellite in orbit. The test is aimed at determining if the desired link delay has been maintained, and if the variance of inter–packet gaps has increased. Finally, both tests must produce no packet loss, as packet loss is considered an artifact in such a scenario.

4.1.1 *LinkEm.* At first, we have experimented with *LinkEm.* The tool has a wide variety of satellite link models, and we were interested if it could perform at Gigabit rates without introducing artifacts. At first, we configured *LinkEm* to produce zero delay and ran a single UDP flow of varying packet sizes and rates. To ensure that *LinkEm* would not suffer from scheduling artifacts, we have configured it to run at priority level -20 via the "nice" command. Table III, Table IV, and Table V show the results for loss ratios, inter–packet gaps, and packet delays. It is interesting to note that at packet rates larger than 40 Kpps, *LinkEm* starts to lose packets. We used "ifconfig -s" to determine if the losses were occurring on the ingress or the egress interfaces. The ingress interface was reported to have zero loss. On the other hand, the egress interface did not transmit all of the packets that were received, meaning that the packets were lost in transit between the interfaces. The data also reveals a large degree of packet jitter (variance in inter–packet gaps) and variance in packet delays. Delay data for 120 Kpps rate and above indicates a significant increase in delay compared to lower packet rates. The additional delay is is most likely due to queuing delays, since the tool cannot forward packets at full rate, hence leading to queue buildups. The experimental date as a whole indicates that *LinkEm* adds a significant level of emulation artifacts.

Table III. LinkEm 0 ms delay: packet loss ratios (%) for 64-, 800- and 1518-byte Ethernet frames

Rate	64	800	1518
8000	0.0000	0.0000	0.0000
40000	0.0000	0.0000	0.0000
80000	2.4797	1.3296	3.4842
120000	11.6460	12.0584	
200000	37.6603		

Table IV. LinkEm 0 ms delay: inter–packet gaps (μs) for 64-, 800- and 1518-byte Ethernet frames

	64 bytes			800 bytes			1518 bytes		
Rate	10th	50th	90th	10th	50th	90th	10th	50th	90th
8000	99	145	151	99	145	151	99	144	151
40000	5	22	45	7	22	43	11	12	49
80000	5	10	30	6	9	30	11	12	14
120000	4	7	20	6	7	15			
200000	4	6	12						

Table V. LinkEM 0 ms delay: packet delays (μs) for 64-, 800- and 1518-byte Ethernet frames

	64 bytes			800 bytes			1518 bytes		
Rate	10th	50th	90th	10th	50th	90th	10th	50th	90th
8000	31	51	71	47	67	87	64	85	104
40000	34	49	62	57	72	86	178	228	284
80000	41	56	5108	65	78	89	246	289	1056
120000	45	60	4276	77	90	799			
200000	2101	2880	3613						

Next, we have configured *LinkEm* to emulate a 125 ms delay and no loss. As before, we ran the same set of experiments. Tables VI, VII, VIII show the results for packet loss ratios, inter–packet gaps, and delays. It is interesting to note, that every single experimental run resulted in some level of packet loss. The inter–packet gap values are similar to the previous experiments when zero delay was used. The packet delays, are close to the desired 125 ms delay for packet rates under 120 Kpps. For rates 120 Kpps and above, the packet delay is larger than 125 ms. Just as before, the increase in delay is most likely due to queueing delay.

The experiments with 0 ms and 125 ms delays have revealed that *LinkEm* adds a significant level of emulation artifacts, which include packet loss, jitter, and delay. Hence it is not a good choice for high–fidelity emulation experiments.

4.1.2 *Click.* Since *LinkEm* runs in user–space, we were interested in using the Click modular router kernel module. To take advantage of the multiple cores available on the node, we configured Click to assign each packet path, as shown in Figure 3, to a separate CPU. Because the packet handling code operates in the kernel, there is no overhead of copying data from the kernel to the user–space. Hence, we expect the tool to provide low jitter, low packet delay, and no loss.

Tables IX and Table X show the results with Click bridging with zero delay. As there was no packet loss, we did not include a loss ratio table. The data for inter–packet gaps

Table VI. LinkEm 125 ms delay: packet loss ratios (%) for 64-, 800- and 1518-byte Ethernet frames

Rate	64	800	1518
8000	1.1798	1.1767	1.1807
40000	0.7923	0.7986	0.2206
80000	10.8434	15.4696	21.2297
120000	28.9757	41.1499	
200000	57.9171		

Table VII. LinkEm 125 ms delay: inter–packet gaps (μs) for 64-, 800- and 1518-byte Ethernet frames

	64 bytes			800 bytes			1518 bytes		
Rate	10th	50th	90th	10th	50th	90th	10th	50th	90th
8000	98	144	152	98	144	152	98	144	152
40000	4	23	46	7	25	44	11	13	48
80000	4	12	30	6	12	28	11	13	23
120000	4	10	19	6	12	25			
200000	3	10	27						

Table VIII. LinkEm 125 ms delay: packet delays (μs) for 64-, 800- and 1518-byte Ethernet frames

	64 bytes			800 bytes			1518 bytes		
Rate	10th	50th	90th	10th	50th	90th	10th	50th	90th
8000	125040	125060	125080	125059	125079	125097	125073	125093	125115
40000	125040	125060	125078	125063	125081	125098	125190	125247	125298
80000	125051	125074	130723	125080	125150	126242	125275	125995	126623
120000	128122	128343	130169	125745	125887	126296			
200000	128105	129286	131500						

and delays reveals that the difference between the 10th and 90th percentiles has slightly increased from the calibration values in Section 3. However, the level of noise induced by Click is significantly smaller than the noise induced by *LinkEm*.

Table IX. Click 0 ms bridge: inter–packet gaps (μs) for 64-, 800- and 1518-byte Ethernet frames

	64 bytes			800 bytes			1518 bytes		
Rate	10th	50th	90th	10th	50th	90th	10th	50th	90th
8000	121	125	129	121	124	129	121	126	129
40000	21	25	30	12	26	31	13	25	33
80000	4	12	16	7	12	19	11	13	14
120000	3	8	14	6	7	12			
200000	2	5	9						

Table X. Click 0 ms bridge: packet delays (μs) for 64-, 800- and 1518-byte Ethernet frames

	64 bytes			800 bytes			1518 bytes		
Rate	10th	50th	90th	10th	50th	90th	10th	50th	90th
8000	34	35	38	56	58	60	78	79	82
40000	34	36	41	57	70	85	79	93	108
80000	22	27	32	48	55	62	90	96	105
120000	20	23	29	47	53	61			
200000	22	26	31						

The results for packet inter–packet gaps and packet delays with Click, configured to provide 125 ms of delay, can be found in Tables XI and XII. There was no loss of packets.

The data for inter–packet gaps and packet delays indicates relatively small differences between the 10th and 90th percentiles. Also, it can be seen that the packet did experience the 125 ms delay with a variance of a fraction of a milli–second.

Click performed significantly better than *LinkEm*, by providing the desired link delays and not inducing a loss of packets. Hence, in the remainder of this report, we will only use Click for link delay and shaping.

Table XI. Click 125 ms bridge: inter–packet gaps (μs) for 64-, 800- and 1518-byte Ethernet frames

	64 bytes			800 bytes			1518 bytes		
Rate	10th	50th	90th	10th	50th	90th	10th	50th	90th
8000	111	126	141	111	127	141	111	126	141
40000	21	25	29	20	24	29	13	24	34
80000	2	12	22	7	12	20	11	13	14
120000	2	8	16	6	7	12			
200000	2	4	10						

Table XII. Click 125 ms bridge: packet delays (μs) for 64-, 800- and 1518-byte Ethernet frames

	64 bytes			800 bytes			1518 bytes		
Rate	10th	50th	90th	10th	50th	90th	10th	50th	90th
8000	125083	125152	125224	125107	125176	125247	125127	125196	125267
40000	125034	125047	125061	125058	125071	125084	125081	125093	125106
80000	125029	125035	125043	125052	125058	125066	125094	125101	125110
120000	125026	125031	125038	125052	125057	125065			
200000	125030	125036	125043						

To further ascertain the applicability of Click as a link shaper, we have conducted bandwidth shaping experiments. We have modified the configuration file to provide a rate limit of 155 Mbps. This equivalent to an OC-3 link speed. To gauge the effectiveness of bandwidth limiting, we have ran the same set of experiments as before.

Table XIII lists the inter–packet gaps. The theoretical minimum inter–packet gaps corresponding to using the entire link capacity are 3.30, 41.28, and 78.34 for 64-, 800- and 1518-byte Ethernet frames respectively. The maximum packet rate for 64-byte frames at 155 Mbps could not be achieved, as it is over 300 Kpps. However, for 800- and 1518-byte frames, the maximum rate has been achieved. The data in the table indicates that the desired link shaping was achieved. Table XIV shows that for 800- and 1518-byte packet delays are larger than the specified 125 ms. This is because the packets have experienced queueing delay due to a reduction of bandwidth, as shown in Figure 3. The size of the queue can be adjusted to any value. If the queue is very small, then the total delay would be close to target value. However, if the queue is small then, the link shaper might drop packets that come in bursts. Ideally, *the queue size should be the same capacity as on the device whose link is being emulated.*

Table XIII demonstrates the percentile statistics for inter–packet gap data. However, it is sometimes valuable to look at the immediate data. Figure 5 demonstrates a small window of sequential packets and their inter–packet gap values, after link shaping has been applied to a 1518-byte flow at 80 Kpps. The plot looks like a scatter plot, as no packets with consecutive sequence numbers were received (two adjacent sequence numbers cannot be joined to form a line). The inter–packet gaps in the graph have a maximum variation of

Table XIII. Click 125 ms 155 Mbps bridge: inter–packet gaps (μs) for 64-, 800- and 1518-byte Ethernet frames

	64 bytes			800 bytes			1518 bytes		
Rate	10th	mean	90th	10th	mean	90th	10th	mean	90th
8000	111	127	141	112	126	141	14	135	152
40000	20	25	30	38	42	46	72	80	86
80000	3	12	22	37	42	48	71	80	85
120000	2	8	16	36	41	48			
200000	2	3	10						

Table XIV. Click 125 ms 155 Mbps bridge: packet delays (μs) for 64-, 800- and 1518-byte Ethernet frames

	64 bytes			800 bytes			1518 bytes		
Rate	10th	50th	90th	10th	50th	90th	10th	50th	90th
8000	125084	125156	125230	125142	125209	125280	125204	125272	125343
40000	125038	125051	125065	135574	135594	135615	145105	145138	145172
80000	125032	125038	125047	135577	135594	135612	145112	145143	145176
120000	125028	125033	125040	135579	135595	135612			
200000	125034	125041	125048						

10 μs and are roughly centered around the desired 78 μs line, meaning that the link shaper has performed an adequate task of evenly spacing the packets in time.

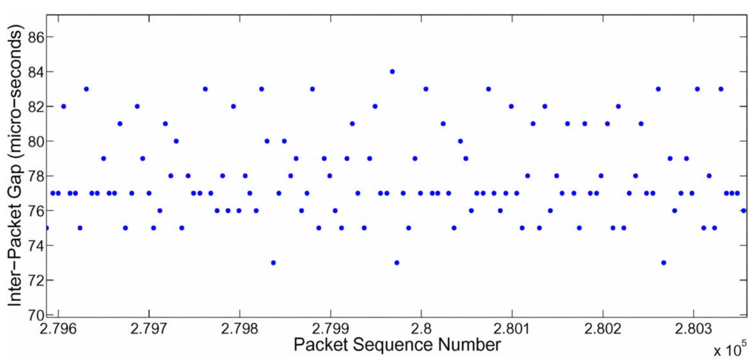

Fig. 5. Inter–packet gaps

4.2 Router Rate Limiting

Link shaping can be performed on a router. This can potentially reduce complexity as no extra link shaping node is required, and there is no need to match the size of the link shaping node's queue to that of the router.

Prior to conducting an experiment with a rate limiter, we have conducted a test where we measured inter–packet gaps and packet delays with a base–line configuration. The router under test is a Cisco 2851, as demonstrated in Figure 1. For the experiments in this section we removed the link shaping node, and connected the Click traffic source/sink to the router directly. Table XV and Table XVI present the results for inter–packet gaps and packet delays. The results indicate that the router does not add a significant amount of delay when compared to calibration data. Also, there is very little variance in inter–packet gaps for packets coming from the router, meaning that packet forwarding functions well.

Table XV. Cisco 2851 1 Gbps: inter–packet gaps (μs) for 64-, 800- and 1518-byte Ethernet frames

	64 bytes			800 bytes			1518 bytes		
Rate	10th	50th	90th	10th	50th	90th	10th	50th	90th
8000	123	125	127	123	125	127	123	125	127
40000	21	25	29	23	25	27	18	25	31
80000	5	12	18	8	12	17	11	13	14
120000	4	9	12	6	7	12			
200000	2	4	11						

Table XVI. Cisco 2851 1 Gbps: packet delays (μs) for 64-, 800- and 1518-byte Ethernet frames

	64 bytes			800 bytes			1518 bytes		
Rate	10th	50th	90th	10th	50th	90th	10th	50th	90th
8000	16	17	18	33	33	35	48	49	50
40000	16	17	21	33	33	34	49	50	53
80000	15	17	21	33	35	39	56	60	348
120000	14	16	19	35	38	43			
200000	15	19	23						

Next, we have enabled a rate limiter on the interface via the *rate-limit* command. We have specified an output limit of 155 Mbps and set the burst size to the minimum allowed setting of 77500 bytes. We opted for the smallest burst size to ensure that inter–packet gaps have the smallest degree of variance. Table XVII and Table XVIII demonstrate the results for inter–packet gaps and packet delays. As was stated in Section 4.1.2, the theoretical minimum inter–packet gaps corresponding to using the entire link capacity are 3.30, 41.28, and 78.34 for 64-, 800- and 1518–byte Ethernet frames respectively. It is interesting to note, that even the 90th percentiles are below the target values. However, the mean values for the inter–packet delays are close to the ideal values. Figure 6 demonstrates the reason for such statistics. Even though the mean inter–packet gaps are close to the ideal target, the analysis of individual packets shows a high degree of burstiness by the shaping mechanism. The router drops a sequence of packets and then transmits a burst of queued packets. This results in a very large gap between the first packet of a current burst and the last packet of a previous burst. Hence, this behavior explains the L–shaped curves seen on the graph.

Even though the router has achieved the desired link rate–limits, it has induced a large degree of burstiness in the traffic. Such burstiness can be detrimental for studies of real–time protocols such as VoIP. Additionally, the router does not provide a mechanism to delay the packets, thus requiring an additional delay node.

4.3 Ethernet Pause Frames

The final shaping method available to us is the IEEE 802.3x pause frames approach. As was discussed in Section 2.2.3, pause frames can be used to induce delays and limit the traffic rate. Because of how pause frames operate, there is a limit as to how much of packet delay they can induce. On a 1 Gbps link, the maximum achievalbe delay is 33.553 ms [2].

[2] $\frac{65535 \times 512\ b}{1e9\ bps} \times \frac{1000\ ms}{1sec} = 33,553\ ms$

Table XVII. Cisco 2851 155 Mbps rate limit: inter–packet gaps (μs) for 64-, 800- and 1518-byte Ethernet frames

	64 bytes			800 bytes			1518 bytes		
Rate	10th	50th	90th	10th	50th	90th	10th	50th	90th
8000	123	125	127	122	125	128	122	125	128
40000	20	25	30	22	25	28	20	25	31
80000	6	12	19	6	14	16	11	13	14
120000	4	7	14	6	7	14			
200000	4	6	8						

Table XVIII. Cisco 2851 155 Mbps rate limit: packet delays (μs) for 64-, 800- and 1518-byte Ethernet frames

	64 bytes			800 bytes			1518 bytes		
Rate	10th	50th	90th	10th	50th	90th	10th	50th	90th
8000	19	20	21	35	36	37	51	52	53
40000	19	20	24	35	36	38	53	54	57
80000	18	20	25	34	38	41	58	63	68
120000	17	21	25	37	40	43			
200000	418	434	471						

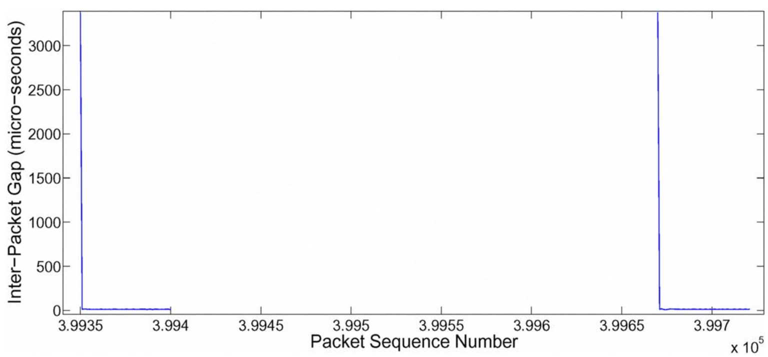

Fig. 6. Inter–packet gaps

Since pause frames are sent periodically, there can be cases when packets do not experience the desired delay. Such situations arise when packets are sent to the router between two sucessive pause frames (i.e. when the link is not paused). Additionally, using pause frames induces heavy jitter. This is due to the fact that the link operates in on/off states, hence making the inter–packets gaps non–constant for a CBR flow. When conducting this experiment, we have configured Click to generate pause frames with a quanta delay value of 390. This is identical to a 200 μs delay on a 1 Gbps link. We have also set the pause frame rate to 4231 packets per second. This rate is necessary to pause a 1 Gbps link to produce a 155 Mbps bandwidth limit. Since we had to use a hidden node to generate pause frames, we configured it to perform a 125 ms packet delay, as the pause frames induced a delay of at most 200 μs.

The results for inter–packet gaps and packet delays are in Table XIX and Table XX. It is interesting to note that the inter–packet gaps exhibit a very large degree of variance and that the 50th percentile is much smaller than the ideal inter–packet gap. Also, the average inter–packet gap values are smaller than the ideal, meaning that the desired rate limit was not achieved. The delays on the other hand are satisfactory.

Table XIX. Cisco 2851 125 ms 155 Mbps pause frames: inter–packet gaps (μs) for 64-, 800- and 1518-byte Ethernet frames

	64 bytes			800 bytes			1518 bytes		
Rate	10th	50th	90th	10th	50th	90th	10th	50th	90th
8000	2	174	238	6	171	227	13	163	220
40000	1	2	162	6	8	186	12	21	194
80000	2	2	18	6	8	186	12	21	194
120000	2	2	22	6	8	186			
200000	2	2	20						

Table XX. Cisco 2851 125 ms 155 Mbps pause frames: packet delays (μs) for 64-, 800- and 1518-byte Ethernet frames

	64 bytes			800 bytes			1518 bytes		
Rate	10th	50th	90th	10th	50th	90th	10th	50th	90th
8000	125120	125235	125355	125167	125276	125390	125209	125312	125421
40000	125104	125187	125269	131439	131596	131670	135971	136179	136392
80000	125156	125224	125283	131457	131647	131686	135992	136210	136434
120000	125210	125255	125298	132119	132317	132369			
200000	129110	129203	129314						

In addition, Figure 7 sheds more light on inter–packet gaps. The figure shows a series of 1518-byte packets with the source rate of 80 Kpps. As expected, the inter–packet gaps vary from a low value, being 20 in this case, to 200 μs (pause frame delay). Even though the data exhibits a large degree of jitter, the pause frame method produces less jitter than the router rate–limit approach.

Ultimately, the pause frame method has not provided us with the desired results, as the bandwidth reduction did not meet our specified target. The complexity of the pause frame method, rivals the complexity of the delay node method, as in both cases, an intermediate node is required. A potential drawback of the pause frame method is the fact that the router has to constantly process pause frames, thus potentially taking away resources from regular data traffic processing. The applicability of this method only makes sense, if the router can correctly process pause frames to meet the desired bandwidth limit, and if the experiment requires the packets to be dropped on the router instead of at the delay node.

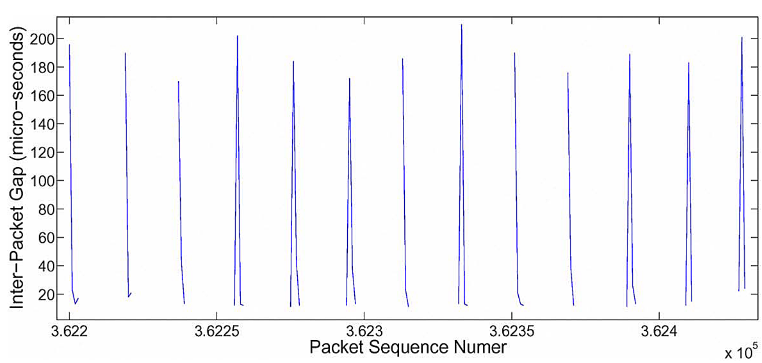

Fig. 7. Inter–packet gaps

5. FINAL SHAPING METHOD

In this section, we present the results with using the Click shaper, as it proved to be superior over the others. Figure 1 demonstrates the experimental topology. The Click shaper was configured to add 125 ms of delay and reduce the link bandwidth to 155 Mbps. The Cisco router was configured not to perform any rate limiting. Additionally, the Click delay node was configured to use 256–slot based output queue.

Tables XXI and Table XXII demonstrate the results for inter–packets gaps and packet delays. The data indicates that the desired inter–packet gaps are achieved, and the difference between the 10th and 90th percentiles is small. Additionally, the packet delays are 125 ms except for cases when queueing delay due to bandwidth shaping has results in additional delays.

Table XXI. Click-Cisco 2851 155 Mbps rate limit: inter–packet gaps (μs) for 64-, 800- and 1518-byte Ethernet frames

	64 bytes			800 bytes			1518 bytes		
Rate	10th	50th	90th	10th	50th	90th	10th	50th	90th
8000	111	127	141	111	127	142	14	138	151
40000	20	24	32	38	42	46	72	79	86
80000	2	12	22	36	42	48	72	79	90
120000	2	7	18	36	42	49			
200000	2	3	11						

Table XXII. Click-Cisco 2851 155 Mbps rate limit: packet delays (μs) for 64-, 800- and 1518-byte Ethernet frames

	64 bytes			800 bytes			1518 bytes		
Rate	10th	50th	90th	10th	50th	90th	10th	50th	90th
8000	125095	125165	125237	125162	125230	125302	125231	125299	125370
40000	125047	125060	125074	135591	135612	135633	145133	145165	145199
80000	125041	125047	125056	135596	135613	135632	145144	145180	145236
120000	125038	125044	125052	135601	135618	135637			
200000	125045	125052	125064						

Figure 8 shows the inter–packet gaps for a flow of 1518-byte packets with a source rate of 80 Kpps. Even though, there are some packets that have a 140+ μs inter–packet gap, the majority of the packets are around the desired 78 μs line. This indicates that the delay node was successfully used in a conjuction with a Cisco 2851 router to emulate a link delay and rate limit the bandwidth. Unlike the previous methods, this method did not produce undesired artifacts and kept jitter to a minimum.

6. CONCLUSION

In this report, we have focused on three link shaping methods: hidden delay bridge, router rate limiting, and pause frames. The focus of the study was to determine which method produced the specified delay and bandwidth limit. In addition, the study also took into consideration variance of inter–packet gaps (jitter). To obtain the results, we have created a variety of constant UDP flows and compared the performance of the link shaping methods with each other. The results revealed that the delay bridge using the Click modular router is superior to the other two methods.

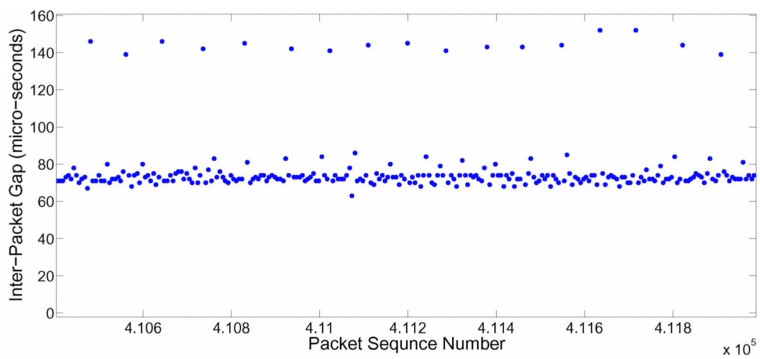

Fig. 8. Inter–packet gaps

REFERENCES

AGARWAL, S., SOMMERS, J., AND BARFORD, P. 2005. Scalable network path emulation. In *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*.

CHERTOV, R., FAHMY, S., AND SHROFF, N. B. 2006. Emulation versus simulation: A case study of TCP-targeted denial of service attacks. In *Proc. of TridentCom*. Extended version accepted to appear in ACM Transactions on Modeling and Computer Simulation (TOMACS).

CHERTOV, R., FAHMY, S., AND SHROFF, N. B. 2007. A black-box router profiler. In *Proc. of Global Internet*.

CHERTOV, R., FAHMY, S., AND SHROFF, N. B. 2008. A device-independent router model. In *Proc. of IEEE INFOCOM*.

CISCO SYSTEMS. 2008. Commands: rate-limit – set qos-group. `http://www.cisco.com/en/US/docs/ios/12_2/qos/command/reference/qrfcmd8.html`.

HUBERT, B. AND ET AL. 2002. Linux Advanced Routing & Traffic Control HOWTO. `http://www.tldp.org/HOWTO/Adv-Routing-HOWTO/lartc.qdisc.classful.html`.

KOHLER, E., MORRIS, R., CHEN, B., JANNOTTI, J., AND KAASHOEK, M. F. 2000. The Click modular router. *ACM Transactions on Computer Systems 18,* 3 (August), 263–297.

RIZZO, L. DummyNet. http://info.iet.unipi.it/∼luigi/ip_dummynet/.