

Precise and Closed-loop Traffic Generation with Caliper

Monia Ghobadi, Geoffrey Salmon,
Yashar Ganjali

Department of Computer Science
University of Toronto
{monia, geoff, yganjali}@cs.toronto.edu

Martin Labrecque, J. Gregory Steffan

Department of Electrical and Computer
Engineering
University of Toronto
{martinl, steffan}@eecg.toronto.edu

There are many challenges associated with performing valid experiments in network testbeds. Generating realistic and responsive traffic that reflects different network conditions and topologies is one of such key challenges. To perform network experiments, researchers often use a collection of commodity Linux machines as traffic generators. However, creating a large number of connections in order to accurately model the traffic shape in networks with thousands of flows is difficult for several reasons: (i) it is not always possible to use real network traces as they do not maintain the feedback loop between the network and traffic sources, (ii) the complexities of traffic generation increase when trying to capture the heterogeneity of link capacities using only a limited number of physical machines, and (iii) commodity hardware does not guarantee the precision of generated traffic, which is bounded by the system timer's resolution.

In this work, we present Caliper, a precise and responsive traffic generator based on the NetFPGA platform with highly-accurate packet injection times that can be easily integrated with various software-based traffic generation tools. Caliper has the same accuracy level as the NetFPGA-based packet generator by Covington *et al.* [1], but provides a key additional feature that makes it useful in a larger range of network experiments: Caliper injects dynamically created packets, and thus, it can react to feedback and model the closed-loop behavior of TCP and other protocols. The ability to produce *live* traffic makes Caliper useful to explore a variety of what-if scenarios by tuning user, application, and network parameters.

Caliper is built on *NetThreads* [2], a platform we have created for developing packet processing applications on FPGA-based devices and the NetFPGA in particular. Caliper's main objective is to precisely control the transmission times of packets which are created in the host computer, continually streamed to the NetFPGA, and transmitted on the wire. The generated packets are sent out of a single Ethernet port of the NetFPGA, according to any given sequence of requested inter-transmission times. Unlike previous works that replay packets with prerecorded transmission times from a trace file, Caliper generates live packets and supports closed-loop traffic. Therefore, Caliper is easily coupled with existing traffic generators (such as Iperf) to improve their accuracy at small time scales.

A packet's journey through Caliper can be summarized as the following stages: (1) First, a user space process or a kernel module on the host computer determines when a packet should be sent. A description of the packet, containing the transmission time and all the information necessary to assemble the packet is sent to the NetFPGA driver. (2) In the driver, multiple packet descriptions are

combined and copied to the NetFPGA card. Combining descriptions reduces the number of separate transfers required and is necessary for sending packets at the line rate of 1 *Gbps*. (3) This last part of Caliper runs as software on the NetThreads platform inside the NetFPGA. The driver sends its messages containing the headers of multiple packets and their corresponding transmission times. Then, Caliper prepares these packets for transmission and sends them at the appropriate times.

Caliper is an extended version of PTG [3] with two main additional features: (i) Caliper is integrable with existing Linux based traffic generators and hence generates responsive and live TCP traffic. For example, we used Caliper to produce *paced* TCP traffic for a research project in routers with small buffers. (ii) Caliper is integrable with existing network simulators and can be used as a precise network emulator. In particular, we develop a prototype that allows packets from the *ns-2* network simulator to be sent on a real network using Caliper.

Our evaluations demonstrate that Caliper is able to reproduce packet inter-transmission times from a given arbitrary distribution with high accuracy while capturing the closed-loop feedback of TCP sources. We measure the accuracy of Caliper with various packet arrival rates and observe that with requested traffic rates up to the line rate and MTU sized packets, the maximum error that Caliper incurs is 8 *ns* which is the resolution of the measuring system clock.

Sending packets to the NetFPGA over the 33-MHz 32-bit PCI bus introduces some challenges. Specifically, the number of DMA transfers between the driver and NetFPGA is limited such that the total throughput is only 260 *Mbps* when individually transferring MTU sized packets. We work around this problem by sending only the packet headers across the PCI bus and rebuilding the packets inside the NetFPGA. For NetThreads applications in general, the maximum achievable packet rate depends on the amount of computations done per packet and hence it also is a function of the packet size. The shortest packets are the worst case leaving less cycle budget per packet: the 125MHz clock allows for 1 cycle per packet byte per processor.

1. REFERENCES

- [1] G. A. Covington, G. Gibb, J. W. Lockwood, and N. Mckeown, "A packet generator on the NetFPGA platform," in *FCCM '09*, 2009, pp. 235–238.
- [2] M. Labrecque, J. G. Steffan, G. Salmon *et al.*, "NetThreads: Programming NetFPGA with threaded software," in *NetFPGA Developers Workshop*, August 2009.
- [3] G. Salmon, M. Ghobadi, Y. Ganjali *et al.*, "NetFPGA-based precise traffic generation," in *NetFPGA Developers Workshop*, August 2009.