Harpoon: A Flow-Level Traffic Generator for Router and Network Tests

Joel Sommers
Computer Science Department
University of Wisconsin
jsommers@cs.wisc.edu

Hyungsuk Kim
Electrical and Computer
Engineering Department
University of Wisconsin
hyungsuk@cae.wisc.edu

Paul Barford
Computer Science Department
University of Wisconsin
pb@cs.wisc.edu

ABSTRACT

We describe Harpoon, a new application-independent tool for generating representative packet traffic at the *IP flow level.* Harpoon is a configurable tool for creating TCP and UDP packet flows that have the same byte, packet, temporal, and spatial characteristic as measured at routers in live environments. We validate Harpoon using traces collected from a live router and then demonstrate its capabilities in a series of router performance benchmark tests.

Categories and Subject Descriptors

C.2.6 [**Computer-Communicaton Networks**]: Internetworking—*Routers*; C.4 [**Performance of Systems**]: [Measurement techniques, Modeling techniques, Performance attributes]

General Terms

Measurement, Performance

Keywords

Traffic Generation, Network Flows

1. INTRODUCTION

The network research community has a persistent need to evaluate new algorithms, systems and protocols using tools that create a range of test conditions similar to those experienced in live deployment. Such tools are critical in laboratory facilities where they are used to evaluate behavior and performance of new systems using real networking hardware. They are also essential in emulation environments and simulation environments where representative background traffic is required. Without the ability to create an appropriate array of traffic conditions, new systems run the risk of unpredictable behavior and unacceptable performance when deployed in live environments.

Current best practices for traffic generation have focused on either simple packet streams or recreation of a single application-specific behavior. Packet streaming methods such as those used in tools like `iperf`[1] consist of sequences of packets separated by a constant interval. Another example is an infinite FTP source which is commonly used for traffic generation in simulations. While these approaches provide some insight into network system capabilities, they lack nearly all of the richness and diversity of packet streams observed in the live Internet.

A number of successful application-specific workload generators have been developed, *e.g.*, [2]. These tools focus on generating application-level request sequences that result in network traffic that has the same statistical properties as live traffic from the modeled application. While these tools are useful for evaluating the behavior and performance of host systems, they are often cumbersome for use in router or switch tests and obviously only reconstruct one kind of application traffic, not the diversity of traffic observed in the Internet. This observation suggests the need for a tool capable of generating a range of network traffic such as might be observed either at the edges or core of the Internet.

We developed Harpoon as a tool for generating packet traffic that has the same characteristics as traffic observed at live routers. Our approach in Harpoon is to abstract flow-level traffic generation into a series of application independent file transfers that use either TCP or UDP for transport. Our model also includes both temporal (diurnal effects associated with traffic volume) and spatial (vis-à-vis IP address space coverage) components. The result is a constructive model with ten distinct components that can be parameterized using NetFlow [3] data collected from live routers. The model is summarized in Figure 1.

2. ARCHITECTURE

Harpoon's architecture begins with the notion of unicast file transfers using either TCP or UDP. Harpoon does not address the packet level dynamics of TCP file transfers. Rather, it relies on the version(s) of TCP running on end hosts to transfer the requested files. Modeling UDP traffic is complicated by the fact that packet emission behaviors are largely application-specific. While Harpoon currently contains three simple models of UDP packet transfer, development of a model with a more diverse set of UDP traffic sources is left for future work.

We refer to the lowest level of the Harpoon model as the *file level.* It is made up of two components that have measurable distributional properties. The first component is the *size* of the file transferred, and the second component is the time interval between consecutive file transfer requests (*inter-file request time*). Harpoon makes requests for files with sizes drawn from a modeled file size distribution P_{Size} and requests are separated by time intervals drawn from a

SIGMETRICS/Performance '04, June 12–16, 2004, New York, NY, USA.
ACM 1-58113-664-1/04/0006.

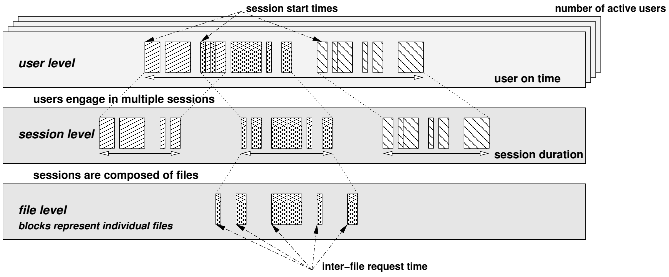

Figure 1: Harpoon's flow-based hierarchical traffic model. Canonical five-tuple flows are analogous to Harpoon users. Users engage in sessions, which are composed of individual files. A heavy-tailed file size distribution and an ON/OFF transfer model generates self-similar packet-level behavior.

modeled inter-file request distribution $P_{InterFile}$.

The middle level of the Harpoon model is referred to as the *session level*. Sessions consist of sequences of file transfers between two distinct IP addresses. The session level has three components, the first is the *IP spatial distribution*, the second is the *inter-session start times*, and the third is *session duration*. Harpoon picks source and destination addresses from ranges of available addresses to make a series of file transfer requests. The address selection is made preferentially using weights drawn from a modeled distribution $P_{IPRange}$. The series of file transfer requests then takes place between the source and destination for a session duration that is drawn from a modeled distribution $P_{SessionOn}$. Consecutive sessions are separated in time by an interval drawn from a modeled distribution $P_{InterSession}$.

The highest level of the Harpoon model is referred to as the *user level*. Harpoon "users" are divided into either "TCP" or "UDP" types which then conduct consecutive sessions using that protocol during the time that they are active. The user level has two components, the *user ON time* and the number of *active users*. Harpoon starts "users" which remain active for a period of time drawn from a modeled distribution P_{UserOn}. By modulating the number of users that are active at any point in time, Harpoon can realize the temporal (diurnal) traffic volumes that are a common characteristic of the Internet. The number of users of each type (TCP/UDP) that are active at any point in a day is taken from a flow data time series. The average number of users is then calculated from flow data to create an empirical model for $P_{ActiveUsers}$ for both TCP and UDP types.

3. VALIDATION

We validated the ability of Harpoon to generate traffic that qualitatively exhibits the same statistical signatures as the distributional models used as input. These input distributions were derived from NetFlow logs captured at our university border router from January 13 to 17, 2003.

Our validation testbed is comprised of three workstations and a Cisco 6509 router. Two workstations act as Harpoon servers (data sources) and one workstation acts as a Harpoon client (data sink.) Our results show that Harpoon faithfully reproduces each distribution given as input. We also validated that as a result of adhering to its input distributions, Harpoon generates traffic volumes consistent with the original flow traces. Our results show that while Harpoon introduces roughly the same number of packets and flows per time interval, it sends many more bytes than were measured in the original flow records. This discrepancy is an effect of using a file size distribution that is not derived from our flow records, which is an area for future work.

4. COMPARISON WITH PACKET-ORIENTED TRAFFIC GENERATORS

In addition to validating the ability of Harpoon to scalably and accurately generate background traffic based on measured flow data, we compare router performance using workloads produced by Harpoon with workloads generated by a standard packet-level traffic generator. Our motivation for comparing Harpoon with a standard traffic generator system is both to demonstrate the differences between the two approaches, and to consider how the standard tools might be tuned to exercise routers more comprehensively.

Our system under test is a Cisco 6509 router, to which we connect workstations running Harpoon and a Spirent AX/4000 traffic generator. While our tests contrast Harpoon with the AX/4000, the model of traffic generation in the AX/4000 is representative of many tools that generate packets with constant spacing, such as `iperf`.

In separate series of experiments, we generate load over two 1 Gbps links with each traffic generator. For each experiment, we measure forwarding rates, CPU, memory, and switching fabric utilization at the router. We choose two different levels of load for Harpoon and tune the constant spacing packet generator and to match Harpoon's average output. We perform experiments with different routing table sizes at the 6509, ranging from 2^5 entries to 2^{16} entries. With the constant spacing packet generator, we perform experiments using a range of burst (a series of packets separated by the minimum inter-frame gap) sizes and packet sizes.

It is clear from our results that precisely controlled traffic streams are useful for Internet RFC conformance testing and for subjecting network systems to extreme conditions along certain dimensions. However, our experiments demonstrate that a workload based on measured characteristics of real Internet traffic generates a fundamentally different and more variable load on routers. Our results suggest ranges of behaviors that can be expected for given average loads. These ranges could be used to tune constant bit rate streams to explore an appropriate operational space. Finally, the subsystem load variably imposed by Harpoon should provide insights to system designers on the stresses that these systems will be subjected to under real operating conditions. This could inform the allocation of resources in future system designs.

5. REFERENCES

[1] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs. Iperf 1.7.0 – the TCP/UDP bandwidth measurement tool. http://dast.nlanr.net/Projects/Iperf, 2004.

[2] P. Barford and M. Crovella. Generating representative workloads for network and server performance evaluation. In *Proceedings of ACM SIGMETRICS '98*, pages 151–160, Madison, WI, June 1998.

[3] Cisco's IOS Netflow feature. http://www.cisco.com/warp/public/732/netflow, 2004.