Public review for
Tmix: A Tool for Generating Realistic TCP Application Workloads in ns-2
Michele C. Weigle, Prashanth Adurthi, Félix Hernández-Campos,
Kevin Jeffay and F. Donelson Smith

When doing simulations, we are confronted with the problem of choosing a good workload model. Often, realistic workload models are difficult to come up with. The paper by Weigle et al. proposes a tool, called Tmix, that allows to automatically extract communication workloads from packet traces and then to replay those workloads in the ns simulator or in test beds.

Interesting features of Tmix are

• A generic and application independent model for capturing the behavior of single TCP connection that also includes connection specific properties such as RTT, window size or loss in order to be able to realistically reproduce TCP streams on the packet level.

• Per-connection behavior can either be extracted from packet level traces or generated from probability distributions.

• An abstract per-connection description that can be used to generate equivalent connections in ns simulators or in test lab environments.

The majority of the reviewers found that Tmix makes an important step in the right direction by creating an abstraction of the real world that reproduces important features of traffic. At the same time, they raised some concerns that the method used does not allow to capture all the essential features:

To describe the behavior of a connection, Tmix extracts a connection vector that essentially consists of a sequence of so called “epochs,” where each epoch $E$ is a triplet $E=(a,b,t)$ with $a$ and $b$ being the amount of data transmitted in each direction and $t$ being the “think time” until the start of the next transfer period. However, today a large amount of the traffic in the Internet is due to peer-to-peer applications for which the rate of transfer may not be determined by the network conditions (available bottleneck bandwidth) but rather by the application itself. Examples are Skype that produces fixed data rate transfers of Edonkey where the application paces the packet transfer as to respect user set rate limitation. The notion of an epoch, as defined in the paper fails to capture this behavior of an application.

Another feature not captured by the current definition of an epoch is the application behavior across connections. However, modern web-browsers often multiplex request/responses onto multiple TCP connections (to the same website) and peer-to-peer applications such as Bittorrent have a large number of simultaneous connections that are choked/un-choked in a coordinated way.

In summary, this paper proposes an interesting approach to workload synthesis. It also raises important questions such as how to extract realistic workloads from packet traces and more generally about the limitations of using ns for realistic performance evaluation.

Public review written by
Ernst Biersack
Institut Eurecom