National Institute of Technology
Karnataka, Surathkal

NITK-IEEE Student Branch

Branch Code: 98631

Tmix - Traffic Generator

By:
Isha Tarte
Sophomore year
Computer Science

Sheetal Shalini
Sophomore Year
Computer Science

Abstract

Simulation of Tmix, realistic synthetic traffic generator in ns3, as well as understanding and realization of Tmix topology in ns.

1 Project Activities

- Understanding Tmix topology.
- Realization of the traffic generated by Tmix.
- Simulation of Tmix in ns3.

2 Introduction

In order to perform realistic network simulations, one needs a traffic generator that is capable of generating realistic synthetic traffic in a closed-loop fashion that "looks like" traffic found on an actual network. Tmix is a realistic synthetic traffic generator. It is an application which converts a cvec(connection vector) or trace file to traffic, and emulates socket-level behaviour of the source. It generates traffic in a closed-loop fashion, which looks like traffic found on actual network. It reproduces pattern of socket reads and writes that the original application performed. The Tmix system takes as input a packet header trace file captured from a network link of interest (such as the link between the UNC campus and the rest of the internet). This trace is reverse-compiled into a connection vector (or cvec) file, which is a source-level characterization of each TCP connection present in the trace. Tmix then uses this information to emulate the socket-level behavior of the source application that created the corresponding connection in the trace. The resulting traffic generation is statistically representative of the traffic measured on the real link.

You may not need to capture a new trace file in order to use Tmix. You can use an existing trace file and use the compilation tool to make it compatible for use in ns-2. The average traffic load can be adjusted either up or down using a scaling tool.
Input: Cvec file.
Output: Traffic generated.

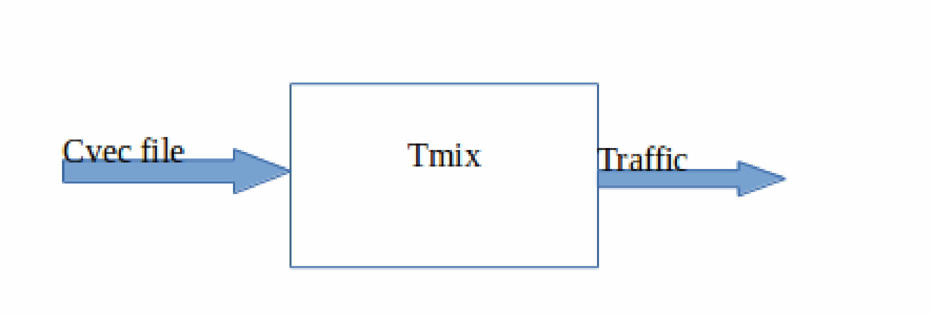

Figure 1: flow diagram

3 Tmix Setup

A typical Tmix topology includes a pair of delay box nodes, an outbound initiator/acceptor node pair, and an inbound initiator/acceptor node pair: Each Tmix initiator node may represent many individual TCP connections as defined by outbound and inbound connection vector (cvec) files. Application traffic flows from the initiator to the acceptor within each pair. The delay box pair allows for simulation of RTTs, bottleneck links, and loss rates for each connection. It is not strictly necessary to use delay boxes; if you don't to simulate these individual connection properties, then you may connect Tmix nodes to normal everyday ns-2 nodes. To use Tmix in ns-2, you may use an existing cvec file pair, or use the tools necessary to convert a network trace into a cvec file and/or to scale the average amount of traffic offered by the Tmix connections.

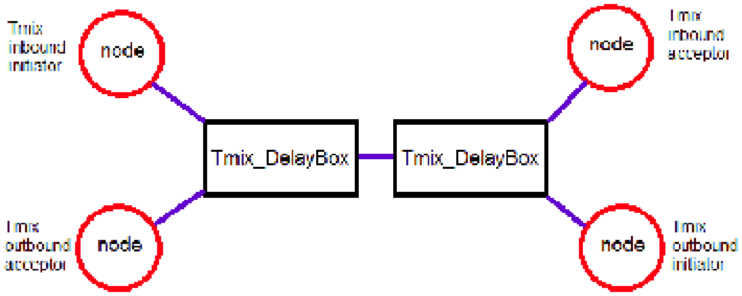

Figure 2: Tmix Topology

4 IMPLEMENTATION OF TMIX IN NS3

Tmix is an ns object that drives the generation of TCP traffic. Each initiator or acceptor cloud is represented by a single ns node that can produce and consume multiple TCP connections at a time. For each TCP connection, Tmix creates initiator and acceptor applications and their associated TCP Agents. Tmix DelayBox: To add the ability to specify delays and losses for each connection using a connection vector file. When we run ns3, 2 trace files are produced.

- Inbound trace file – It is responsible for the traffic generated outside the network.
- Outbound trace file – It is responsible for the traffic generated within the network.

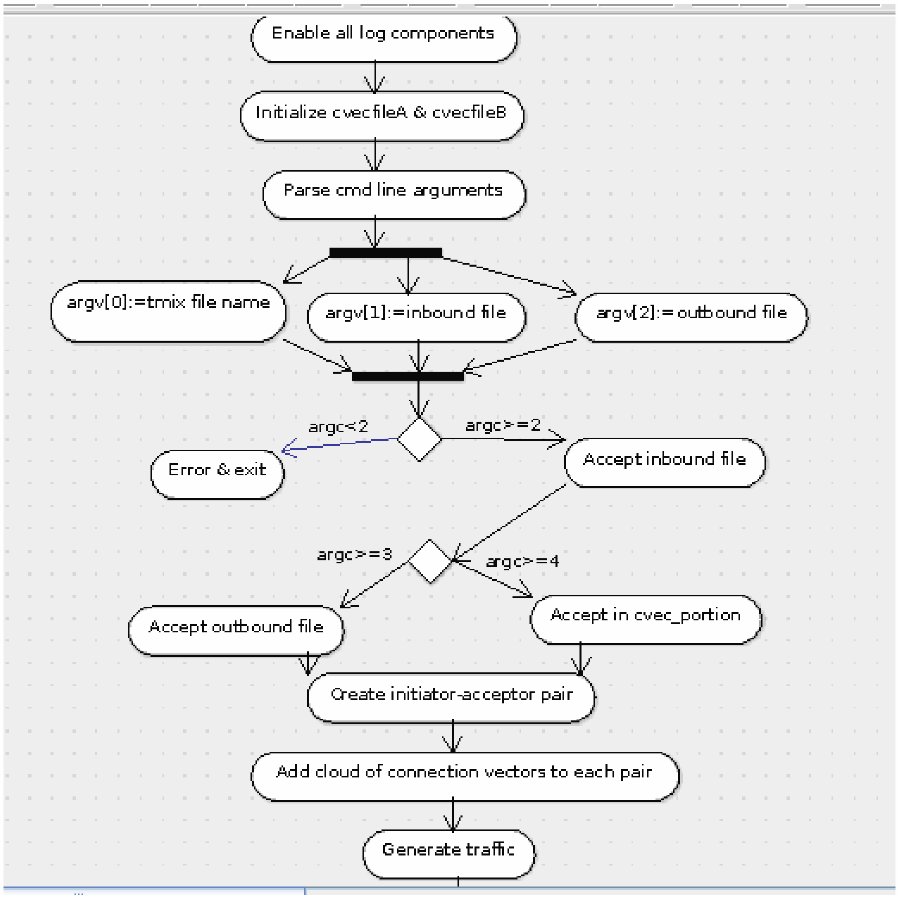

Figure 3: Flowchart

5 Trace File

Trace files reverse-compiled into cvec file format are given as input to tmix. Every trace file contains trace strings. The Format of a trace string is as follows:

- Event: '+' represents enqueue a packet, '-' represents dequeue a packet, 'd' represents drop a packet, 'r' represents reception of a packet.
- Time: indicates the time at which the trace string was generated. From and to nodes
- Packet type: it can be ack(acknowledgement) or tcp.
- Packet size: mentioned in bytes.
- flags: it is a 7-digit flag string.
- fid: flow id
- Source and destination address: they are in the format a.b where a is the address and b is the port number.
- Sequence number and packet id.

6 CVEC File

Connection vector(cvec) is a representation of a TCP connection. The fields specified are:

- Type of connection(sequential or concurrent).
- Connection start time.
- Loss rate.
- Window size.

Each cvec file contains arbitrary number of ADU's(Application data units). Each ADU represents data to be sent over the network by eitherinitiator or acceptor of the TCP connection. They also specify when the data has to be sent.

7 Conclusion

Simulation is the dominant method for evaluating most networking technologies. However, it is well known that the quality of a simulation is only as good as the quality of its inputs. An overlooked aspect of simulation methodology is the problem of generating realistic synthetic workloads to drive a simulation or laboratory experiment. We have developed an empirically-based approach to workload generation. Starting from a trace of TCP/IP headers on a production network, a model is constructed for all the TCP connections observed in the network. The model, a set of a-b-t connection vectors, can be used in the workload generator tmix to replay the connections and reproduce the application-level behaviors observed on the original network. Moreover, by combining set of connection vectors or sub-sampling vectors according to any number of heuristics, a wide range of meaningful departures from reality are possible. This enables researchers to perform “what if” experiments where the synthetic traffic can be manipulated in controlled ways. We believe this approach to source-level modeling, and the tmix generator, are significant contributions to network evaluations, specifically because of their ability to automatically generate valid workload models representing all TCP applications in a network with no a priori knowledge about them. Our work therefore serves to demonstrate that researchers need not make arbitrary decisions when performing simulations such as deciding the number of flows to generate or the mix of “long-lived” versus “short-lived” flows. Given an easily acquired TCP/IP header trace, it is straightforward to populate a workload generator and instantiate a generation environment capable of reproducing a broad spectrum of interesting and important features of network traffic. For this reason, we believe this work holds the potential to improve the level of realism in network simulations.

8 References and Work

- Original paper (2006): Tmix: A Tool for Generating Realistic TCP Application Workloads in ns-2
- ns-2 documentation for Tmix
- Codebase for Tmix in ns-2 (includes example files)