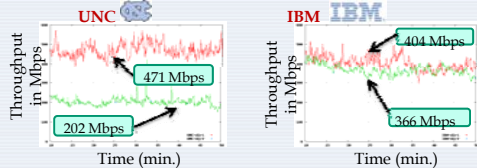# Tools for GENI Experimenters

## Collect Production Network Traffic

As examples, we show below two very different network traces collected at two diverse locations on the Internet used in some of our studies. The first one from UNC was taken on the border link connecting the campus to the Internet service provider network. The second trace was taken at an aggregation switch for four internal networks, connecting one of IBM Corporation's largest development sites to the Internet. Both were 1-hour traces taken at peak times.
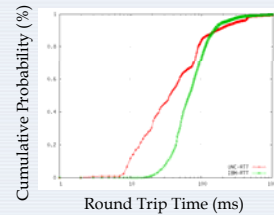
**Traffic Data for Experiments**

**UNC**
Throughput in Mbps
471 Mbps
202 Mbps
Time (min.)

**IBM**
Throughput in Mbps
404 Mbps
366 Mbps
Time (min.)

4.7 million TCP connections carrying 290 GB of application data

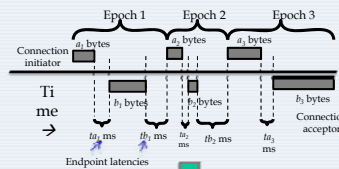2.8 million connections carrying 320 GB of application data

**Round Trip Time of Connections**

Cumulative Probability (%)
Round Trip Time (ms)

- UNC
- **80 ms** – average RTT
- **36 ms** – median RTT

- IBM
- **92 ms** – average RTT
- **68 ms** – median RTT

## Run Experiments on GENI

Anonymized Packet Header Trace → Processing → Source-level Trace: Set of $a$-$b$-$t$ Connection Vectors

Workload Partitioning

**GENI TESTBED**
Traffic Generators ⟷ Traffic Generators

The synthetic trace is comparable to the original
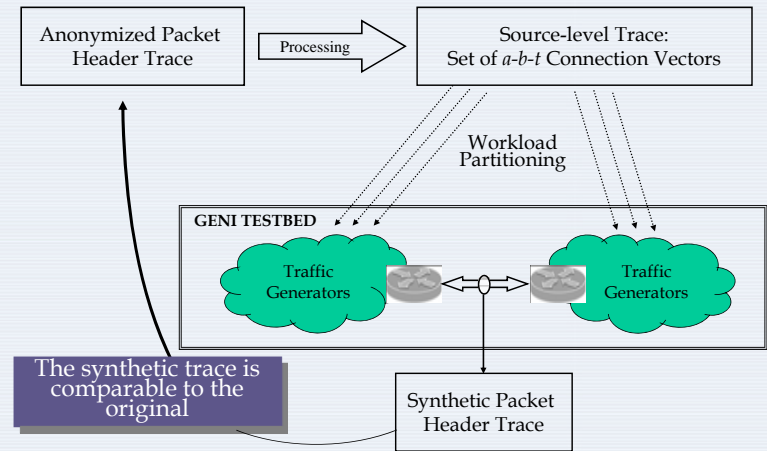
Synthetic Packet Header Trace

## Convert TCP Traces to Connection Vectors

We analyze every connection found in a trace of TCP/IP headers to produce a "connection vector" for each connection. A connection vector includes the connection's start time relative to the beginning of the trace and a descriptor of each request-response exchange in the connection. A request-response exchange, called an "epoch," is described by a 4-tuple consisting of the request size (called the "$a$" unit size), the response size (called the "$b$" unit size) and two endpoint latency values (called the "$t$" values); one for the server-side "think time" between a request and its response and one for the user/client "think time" between successive requests. Unidirectional transfers have only an $a$ or $b$ value depending on the direction of transfer.
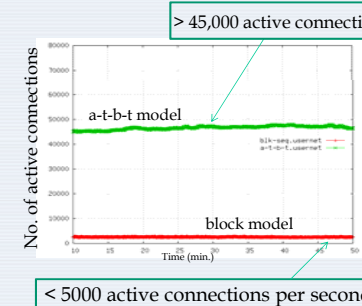
Epoch 1   Epoch 2   Epoch 3

Connection initiator
$a_1$ bytes   $a_2$ bytes   $a_3$ bytes

Time →
$b_1$ bytes   $b_2$ bytes   $b_3$ bytes
Connection acceptor

$ta_1$ ms   $tb_1$ ms   $ta_2$ ms   $tb_2$ ms   $ta_3$ ms

Endpoint latencies

Epoch1:
> a1 bytes
ta1 ms
< b1 bytes
tb1 ms

Epoch2:
> a2 bytes
ta2 ms
< b2 bytes
tb2 ms

Epoch3:
> a3 bytes
ta3 ms
< b3 bytes
tb3 ms

## Why does this matter?

Experiment to test router queuing policy - keep state of the active connections

> 45,000 active connections per second

No. of active connections
Time (min.)

a-t-b-t model

block model

< 5000 active connections per second

- Same traffic input
- Same path characteristics
- Different application models
- *Drastically different outcomes!*

*Note: The block model emulates the same number of bytes in each direction, but omits the epoch structure and the endpoint latencies*

*Jay Aikat, Kevin Jeffay*
*Department of Computer Science*
*University of North Carolina – Chapel Hill*

**Networking Research Group -- http://www.cs.unc.edu/dirt**