TCPivo: A High-Performance Packet Replay Engine

Ashvin Goel, Wu-chang Feng, Abdelmajid Bezzaz, Wu-chi Feng, Jonathan Walpole

My router needs to perform so many functions!

Motivation: need a fast, inexpensive, reproducible, realistic method to test my router

Simulation is slow, doesn't capture timing

Emulation is expensive, unrealistic

Question: what if a trace-driven packet generator is used?

It could be fast, inexpensive, produce reproducible and realistic loads, address mixes, packet timing

Trace-driven packet generator requirements:

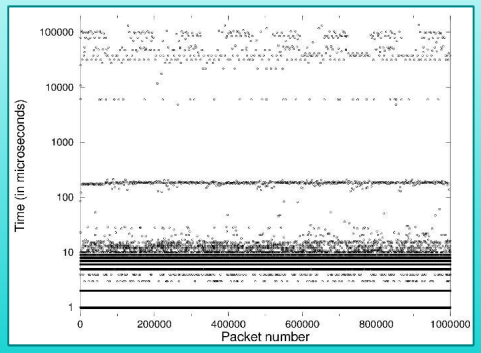
High throughput
- Well understood

High timing accuracy
- For testing router's traffic modulation, buffer provisioning, packet dropping
- For determining router's packet jitter, especially useful for interactive gaming

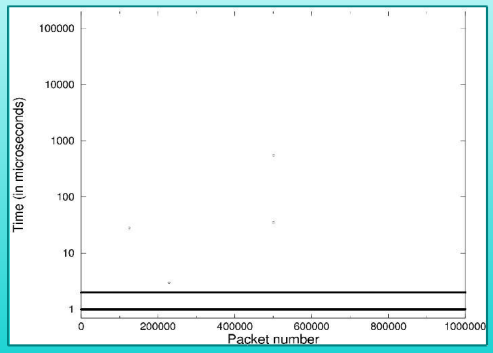
Challenges:
- High throughput
  - I/O management of traces
  - Transmission efficiency
- High timing accuracy
  - Timing of packet send events
  - Scheduling

I/O management of traces

- User-driven prefetching
- Double buffering
- Improves throughput and timing accuracy



`fread()` standard latency



`fread()` latency with `mmap()/madvise()`

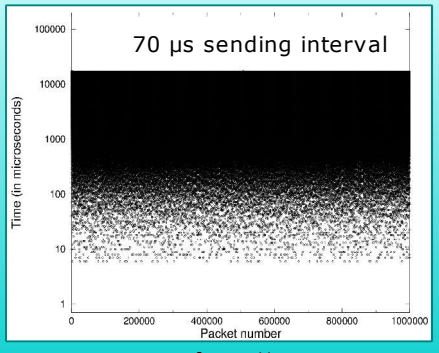
Transmission efficiency

- Zeroed user-level payload
- Kernel/driver level payload
- Allows close to gigabit speeds

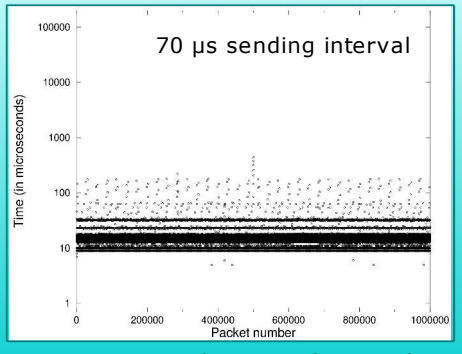
Task	Average time spent
Main loop	**9.38µs**
Data padding	1.45µs
Checksum	1.25µs
`sendto()`	5.16µs
Trace read	1.30µs

Timing of packet send events

- Use one-shot timers
- Use soft timers



`usleep()`
CPU usage: 40%



Firm timers: (APIC, soft timers)
CPU usage: 19%

TCPivo replays traffic at high speed with tight timing accuracy

Uses combination of
- User-driven prefetching
- Kernel support for efficient transmission
- Firm timers for precise, low-overhead timing
- Low-latency Linux for accurate scheduling

OGI SCHOOL OF SCIENCE & ENGINEERING
OREGON HEALTH & SCIENCE UNIVERSITY
http://www.cse.ogi.edu/sysl

Sponsored by intel