Dynamic Forwarding Table Management for High-speed GPU-based Software Routers

Joongi Kim¹, Keon Jang¹, Sangjin Han¹, KyoungSoo Park², and Sue Moon¹
¹Department of Computer Science, KAIST, {joongi, keonjang, sangjin}@an.kaist.ac.kr, sbmoon@kaist.edu
²Department of Electrical Engineering, KAIST, kyoungsoo@ee.kaist.ac.kr

Motivation

- Software routers are gaining momentum
  - in favor of extensibility & flexibility in network packet processing.
- PacketShader achieves 40Gbps. [SIGCOMM ’10]
  - Currently the fastest software router (data-plane speed 40Gbps on a single x86 machine)
  - Next step: control-plane integration
- Will PacketShader keep up?
  - Bursting routing table updates (50-150 times/sec)
  - Large routing & forwarding tables (more than 320,000 entries and a few hundreds MB)
  - Updating forwarding tables in GPU similar to FIB updates in high-end routers

Our key insight on GPU-based software routers:
“Bursty routing table updates hurt the performance of GPU-based software routers”

Design & Ideas

- Software Architecture

  Existing software routing frameworks (e.g. XORP, Quagga)
  Forwarding engine (using GPUs)
  Forwarding table manager
  Packet API
  Linux TCP/IP stack
  Packet I/O driver

  Inside GPU
  Updating forwarding tables
  Uploading continues...

- Idea #1: Double-buffering
  - Modern GPUs have enough memory (1.5GB for GTX480) to store multiple instances of the forwarding table.

- Idea #2: Incremental FIB updates
  - They reduce bandwidth and update time.
  - The data structure for forwarding table is critical.
  - We are considering a few known methods. ([Gupta98], [Basu05], [Zhao10], [Liu10])