Modularity Meets Batching: Towards an Experimental Platform for High-speed Software Routers

Joongi Kim[1] , Seonggu Huh[1], Sangjin Han[2], Keon Jang[1], KyoungSoo Park[3], and Sue Moon[1]
[1]Department of Computer Science, KAIST, {joongi, seonggu, keon}@an.kaist.ac.kr, sbmoon@kaist.edu
[2]Computer Science Division, U.C.Berkeley, sangjin@eecs.berkeley.edu
[3]Department of Electrical Engineering, KAIST, kyoungsoo@ee.kaist.ac.kr

Motivation

Challenges all software routers share:

- **Batch processing** is the key for *performance*.
  - PacketShader showed I/O batching is essential.
- **Modularity** is the key for *programmability*.
  - More routers keep adding new functions, which are difficult to integrate with existing systems.

Hardware development trends:

- Next-generation hardware technology is proceeding to **massively parallel processors**.
  - Example: Tilera's many core processors, AMD's APUs, and AMD/NVIDIA's GPGPUs
- High-performance software routers can benefit from **GPUs**, which parallelizes batch processing.
  - Our prior work PacketShader showed GPUs can boost common packet processing operations.

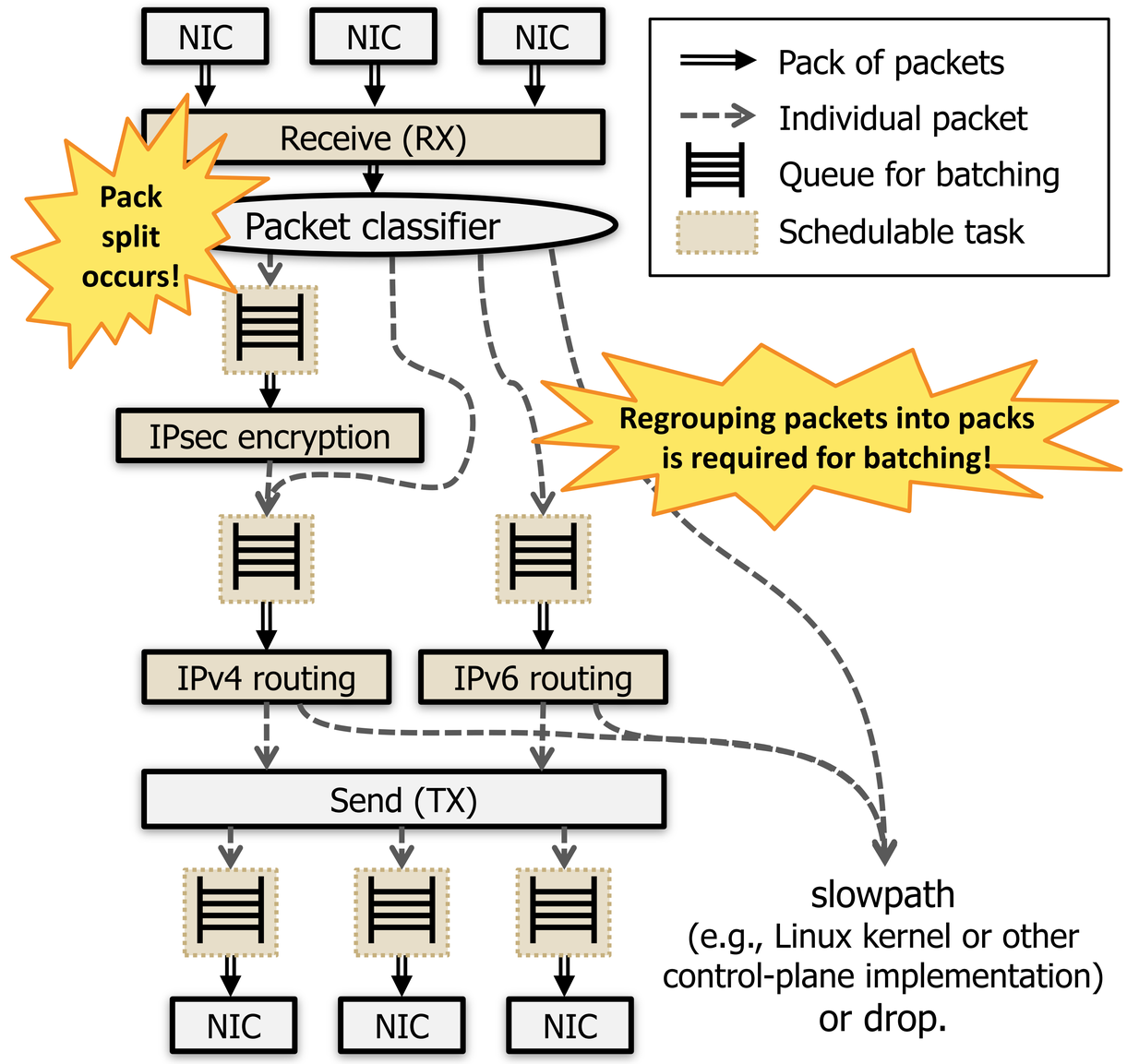
Technical Challenges

- **Per-packet path diversity within a pack of packets**
- **Copy overheads between the host & GPU memory**
- **Load balancing for overloaded cases**

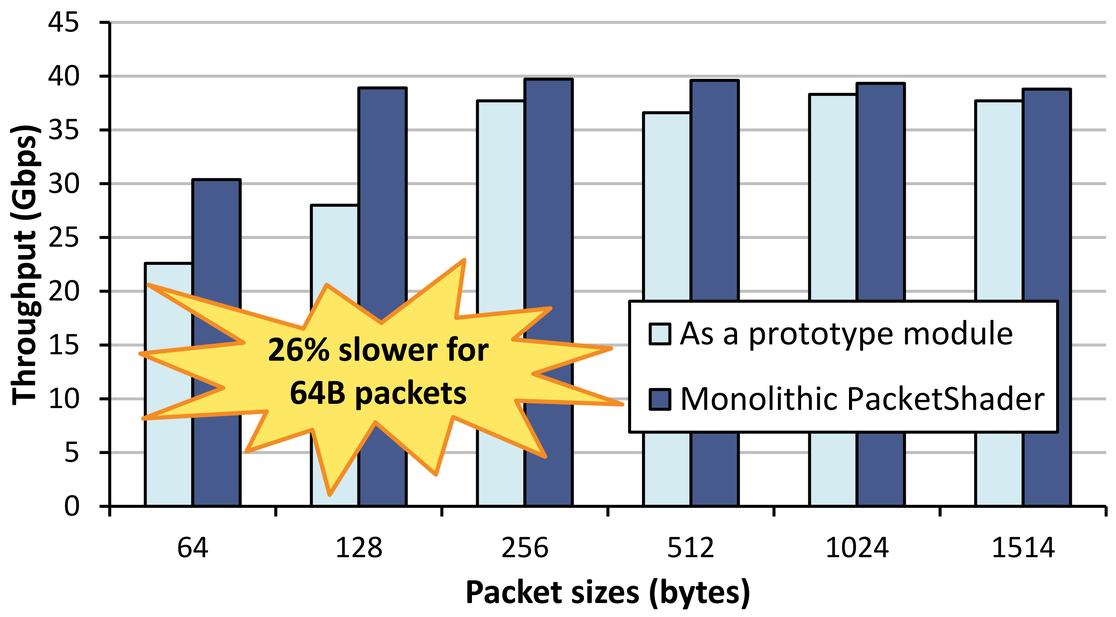
Our Strategic Bullets

- **Efficient pack split/merge mechanisms**
  - Use of **zero-copy pack data structures**
- **Abstraction of memory resources**
  - **Table buffers & packet buffers** for sharing data between the host/GPU memory, differentiated by updating mechanisms
- **Load balancing techniques**
  - **Opportunistic off-loading** of computations to decrease latency of batching (small data → CPU, large data → GPU)
  - **Dynamic module-to-processor assignment** depending on traffic patterns and processor usage

A diverse processing path example



Overhead of preliminary implementation of a packet classifier and input queuing (IPv4 routing)



Accommodating both batching & modularity is ***not trivial.***

This work was supported by NAP of Korea Research Council of Fundamental Science & Technology (KRCF).