GASPP: A GPU-Accelerated Stateful Packet Processing Framework

Giorgos Vasiliadis, *FORTH-ICS, Greece*

Lazaros Koromilas, *FORTH-ICS, Greece*

Michalis Polychronakis, *Columbia University, USA*

Sotiris Ioannidis, *FORTH-ICS, Greece*

Network Packet Processing

- Computationally and memory-intensive
- High levels of data parallelism
  - Each packet can be processed in parallel
- Poor temporal locality for data
  - Typically, each packet is processed only once

GPU = **G**raphics **P**rocessing **U**nits

- Highly parallel manycore devices
- Hundreds of cores
- High memory bandwidth
- Up to 6GB of memory

GPUs for Network Packet Processing

- Gnort [RAID’08]
- PacketShader [SIGCOMM’10]
- SSLShader [NSDI’11]
- MIDeA [CCS’11], Kargus [CCS’12]
- …

GPUs for Network Packet Processing

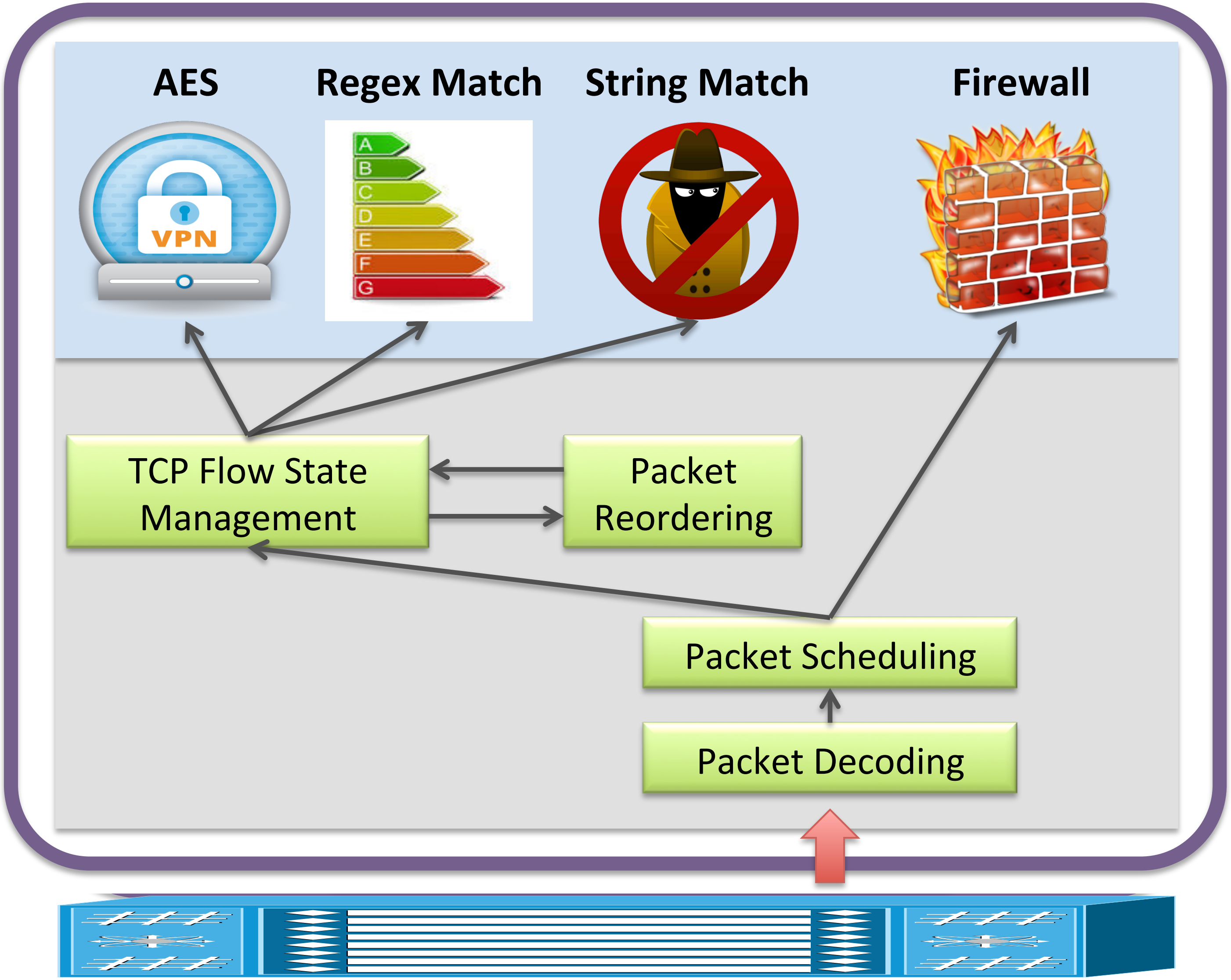
- Gno
- Pac
- SSL
- MID
- …

Independent/Monolithic Designs

1. **A lot of CPU-side code even for simple apps**
2. **Explicit batching**
3. **Explicit data copies and PCIe transfers**

Need a **framework** for developing **GPU** accelerated **packet processing** applications

GASPP Framework



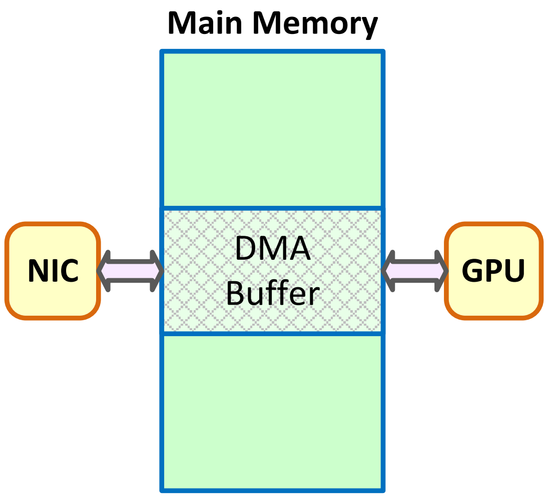
GASPP Framework

- Fast user-space packet capturing
- Modular and flexible
- Efficient packet scheduling mechanisms
- TCP processing and flow management support

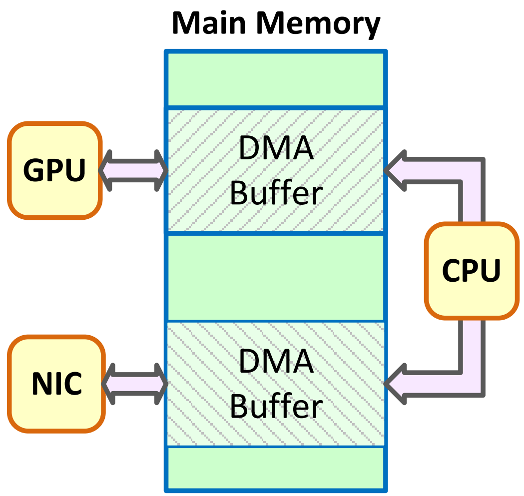
GASPP Framework

- Fast user-space packet capturing
- Modular and flexible
- Efficient packet scheduling mechanisms
- TCP processing and flow management support

Fast user-space packet capturing



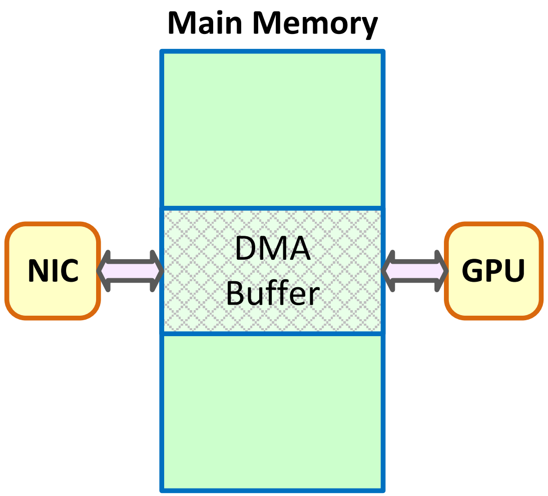
versus



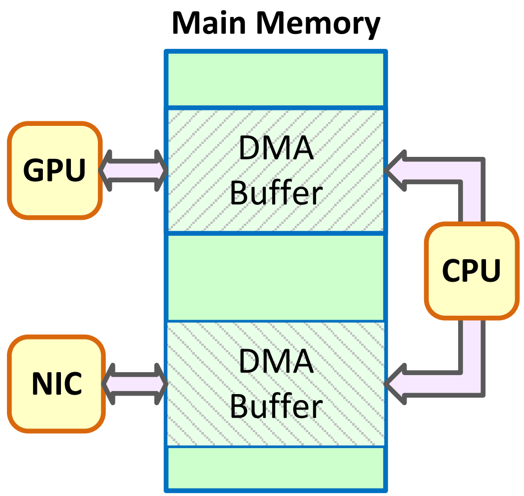
Use a single user-space buffer between the NIC and the GPU

Stage packets back-to-back to a separate buffer

Fast user-space packet capturing



versus

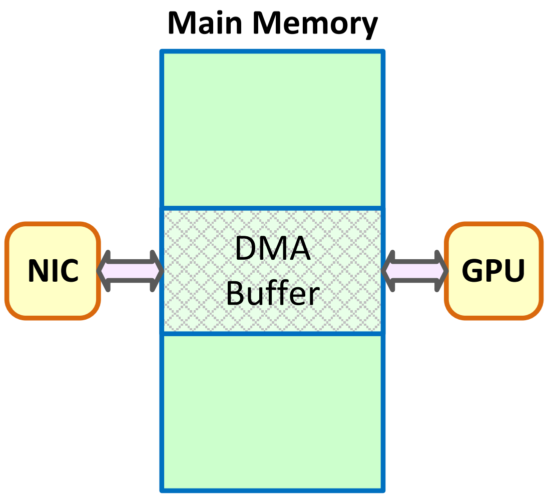


Packets size (#bytes)	Gbit/s
64	**2.06**
128	**4.03**
256	**8.07**
512	**16.13**
1024	**32.26**
1518	**47.83**

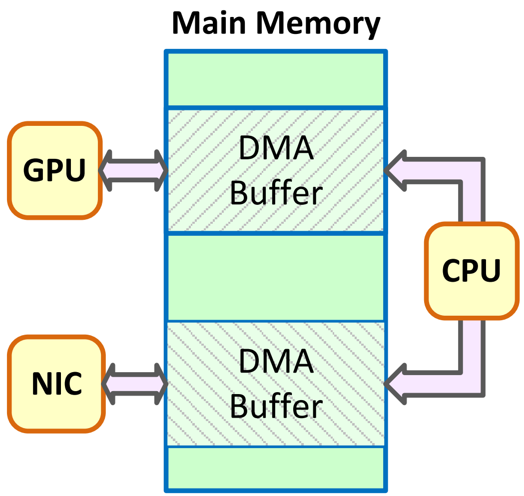
>

Packets size (#bytes)	Gbit/s
64	**13.76**
128	**18.21**
256	**20.53**
512	**19.21**
1024	**19.24**
1518	**20.04**

Fast user-space packet capturing



versus

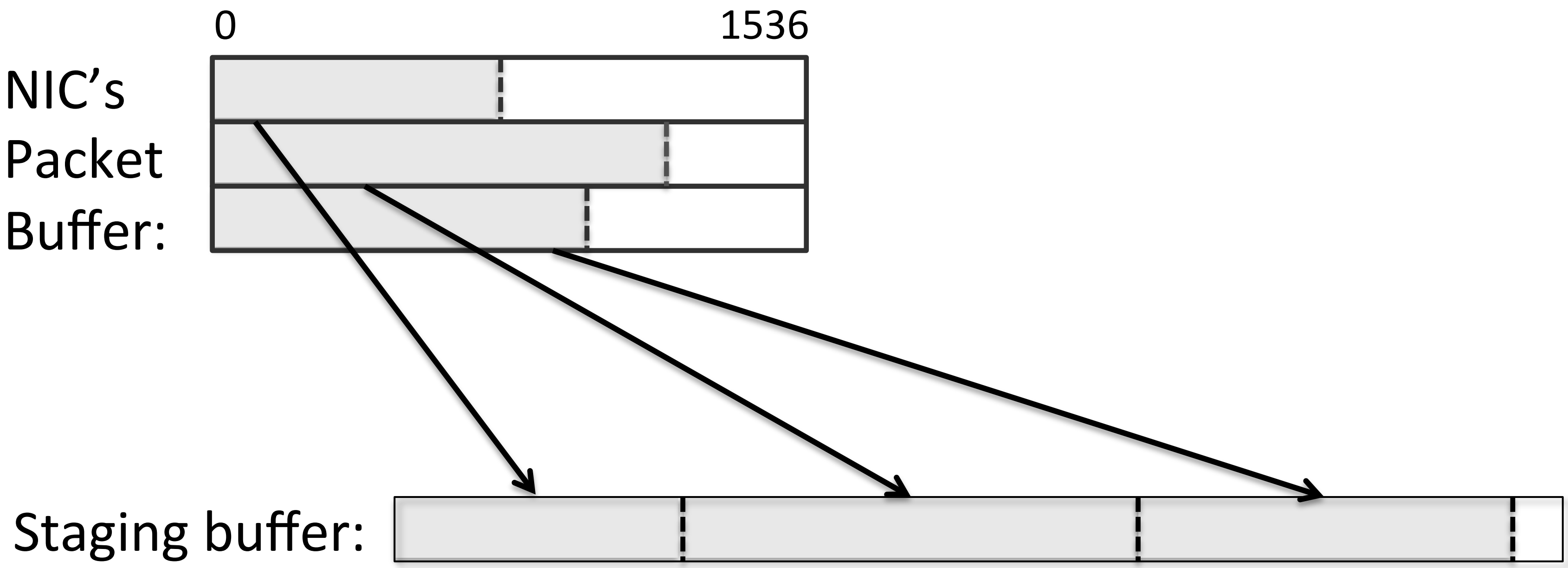


Packets size (#bytes)	Gbit/s
64	**2.06**
128	**4.03**
256	**8.07**
512	**16.13**
1024	**32.26**
1518	**47.83**

<

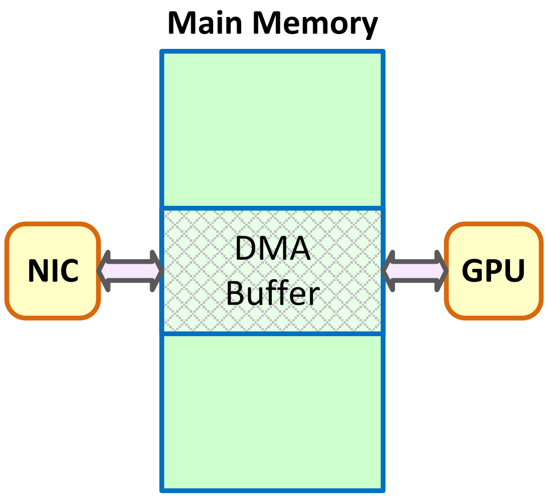
Packets size (#bytes)	Gbit/s
64	**13.76**
128	**18.21**
256	**20.53**
512	**19.21**
1024	**19.24**
1518	**20.04**

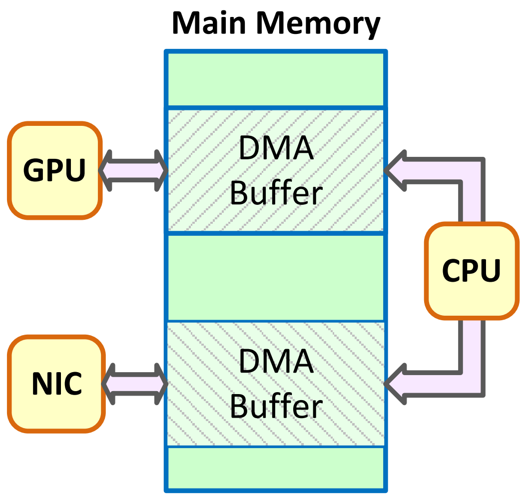
Why staging is better than zero-copy
(for small packets)



Better space utilization => No redundant transfers

Selective scheme





- Packets are are *copied back-to-back* to a separate buffer, if the buffer occupancy is sparse
- *Otherwise*, they are transferred *directly* to the GPU

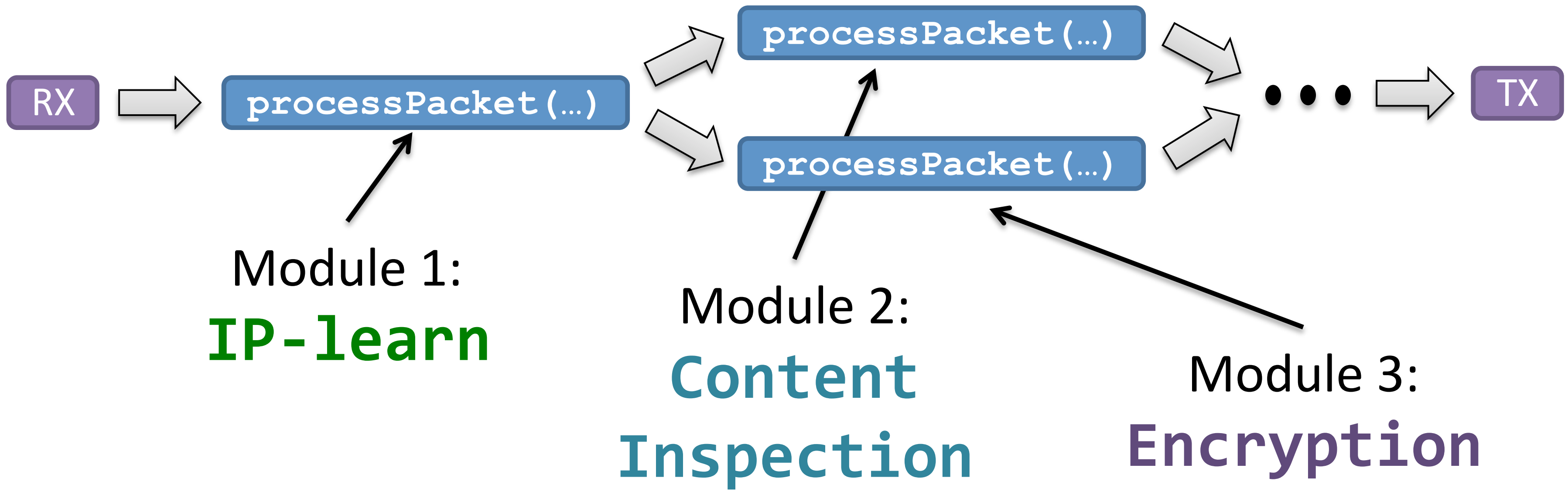
GASPP Framework

- Fast user-space packet capturing
- Modular and flexible
- Efficient packet scheduling mechanisms
- TCP processing and flow management support

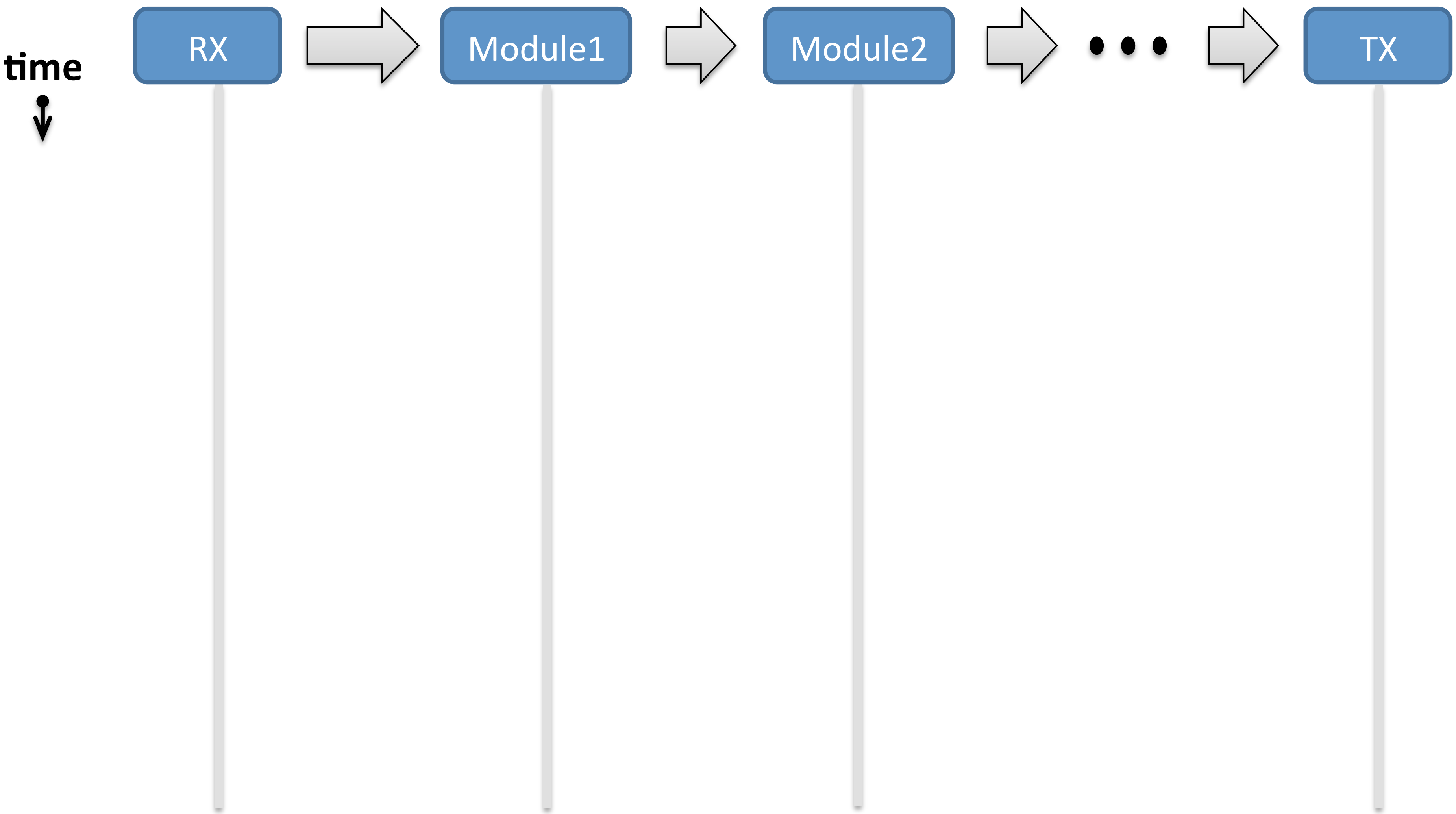
Modular and Flexible

- Basic abstraction of processing: ``modules''
  - `processPacket(packet){ ... }`

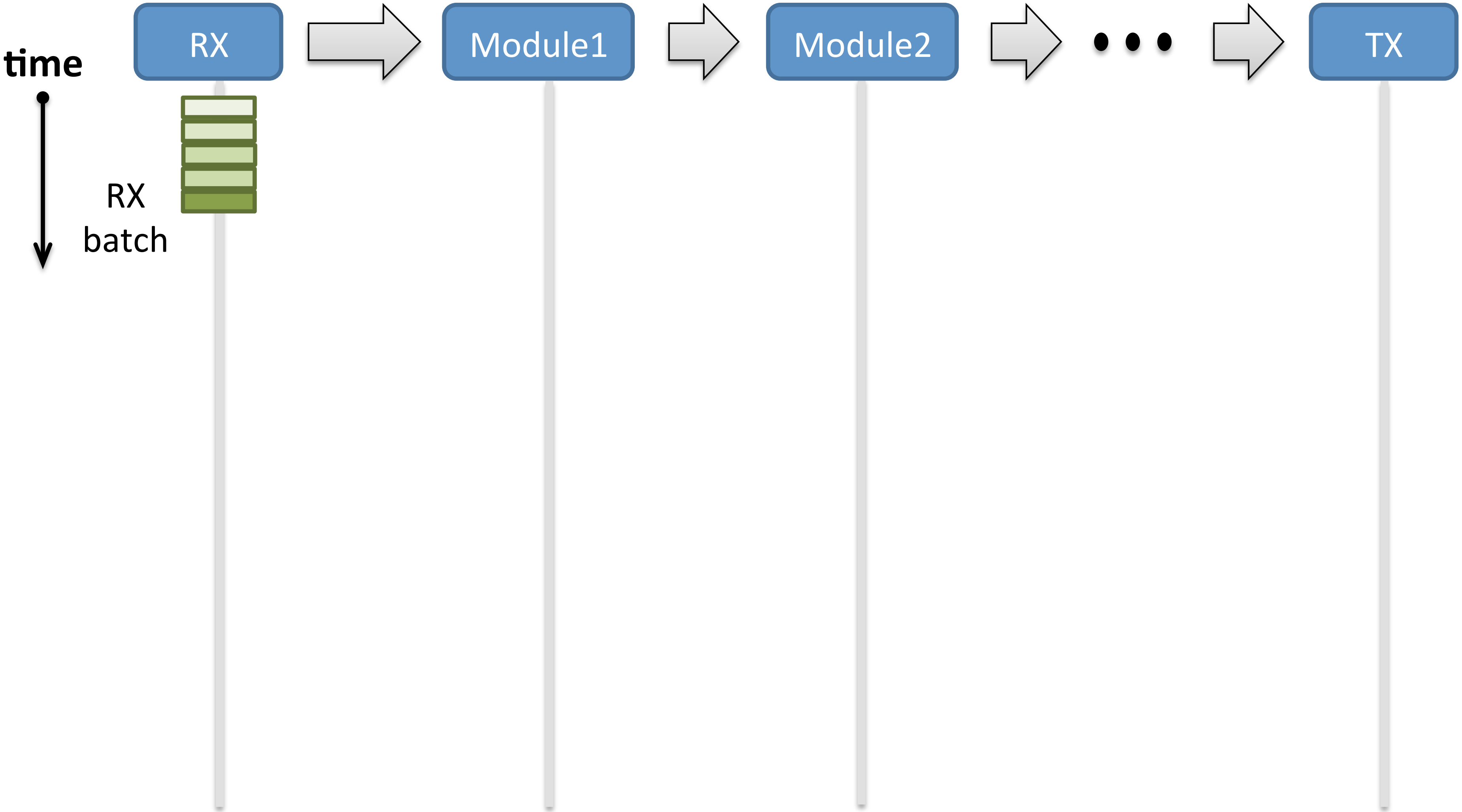
- Modules are executed sequentially or in parallel



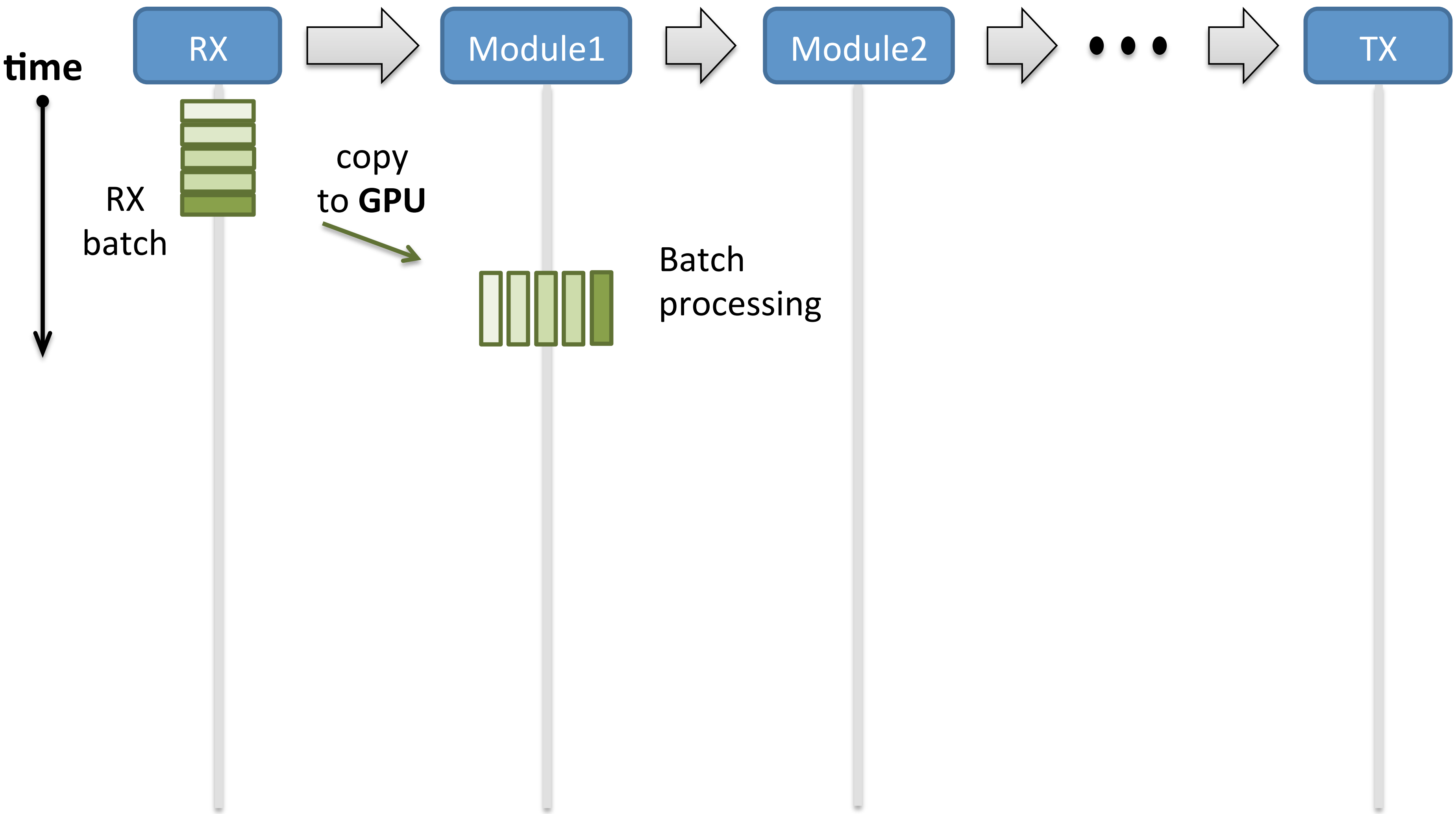
Batch Processing Pipeline



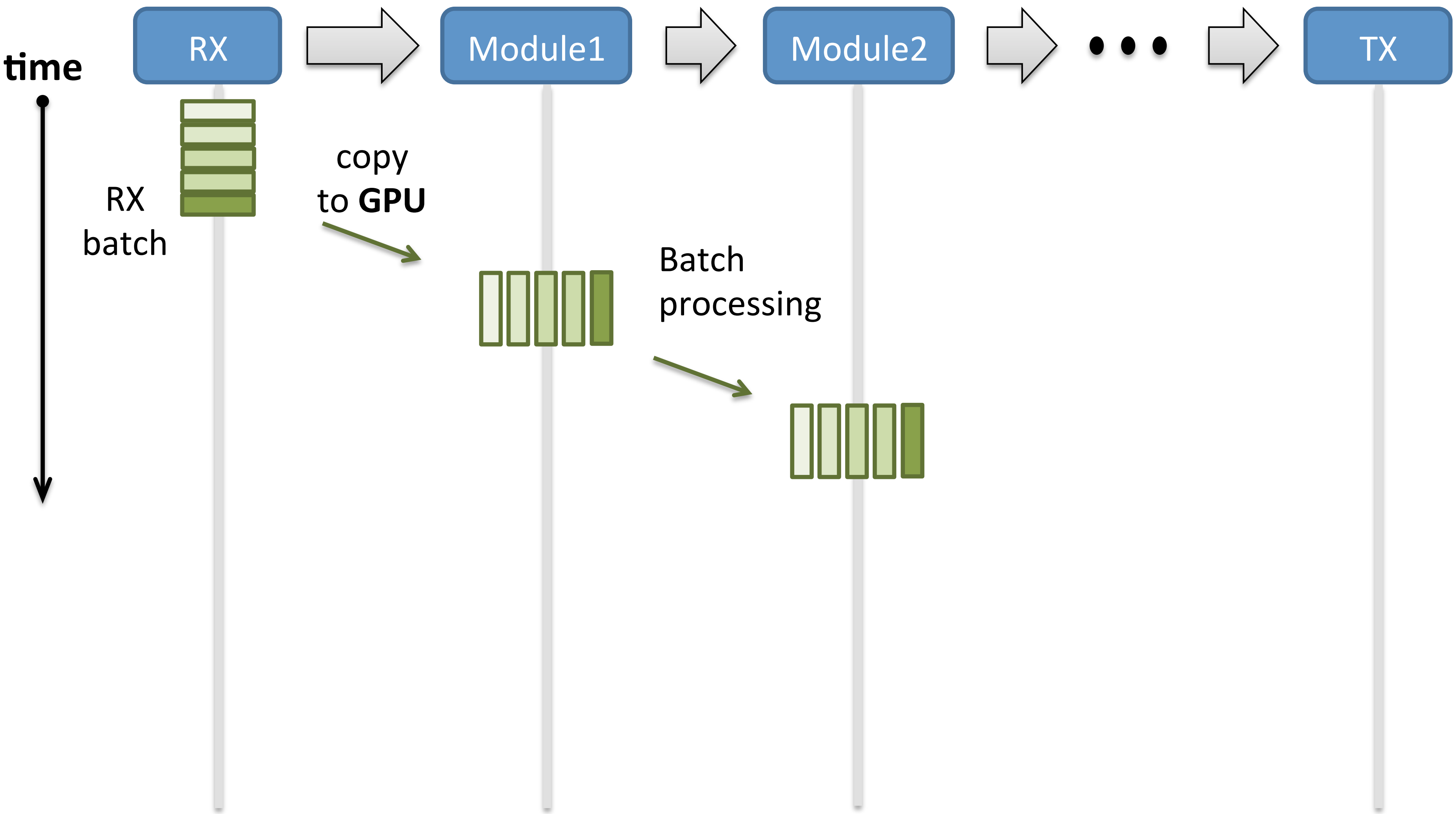
Batch Processing Pipeline



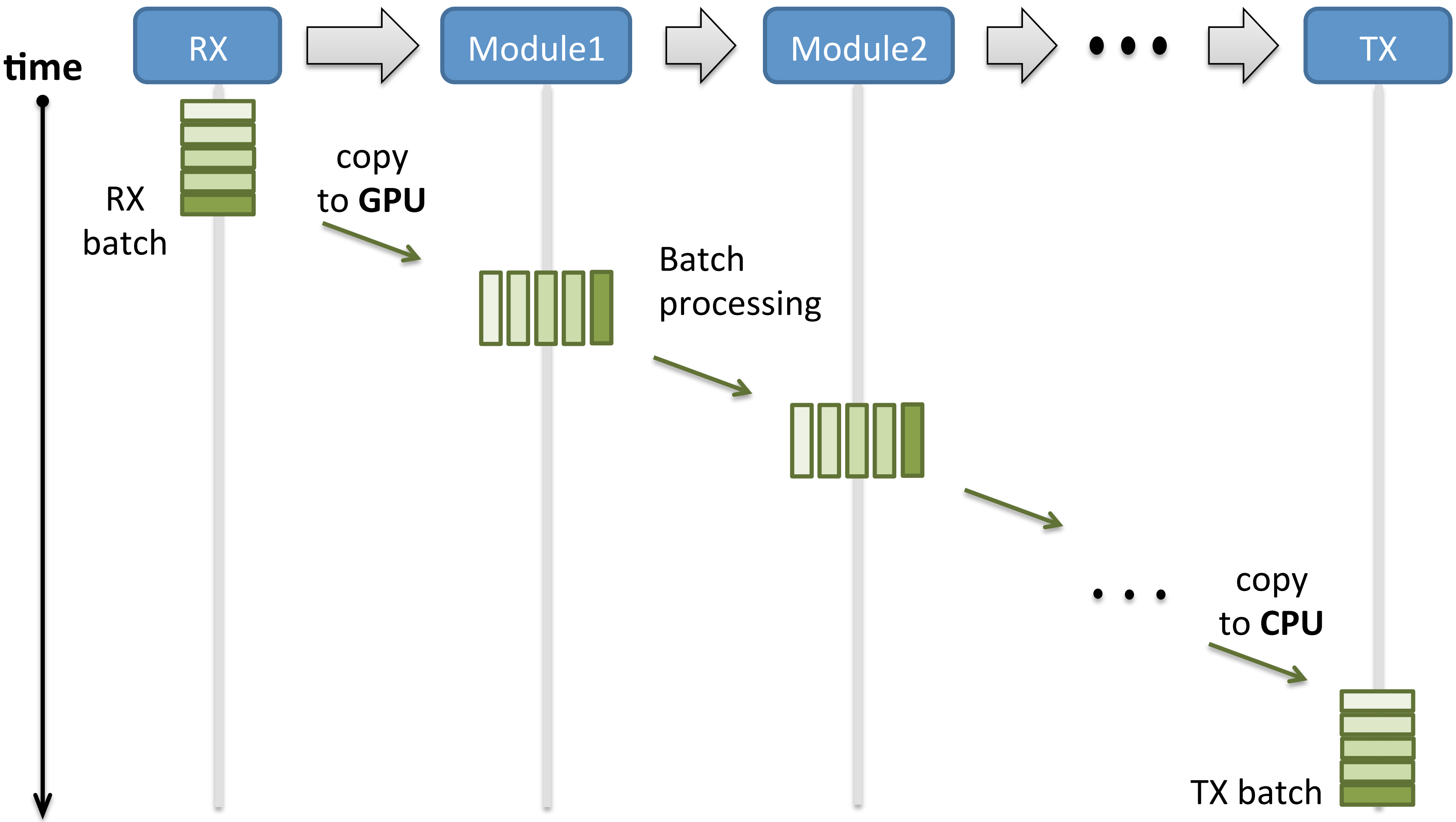
Batch Processing Pipeline



Batch Processing Pipeline



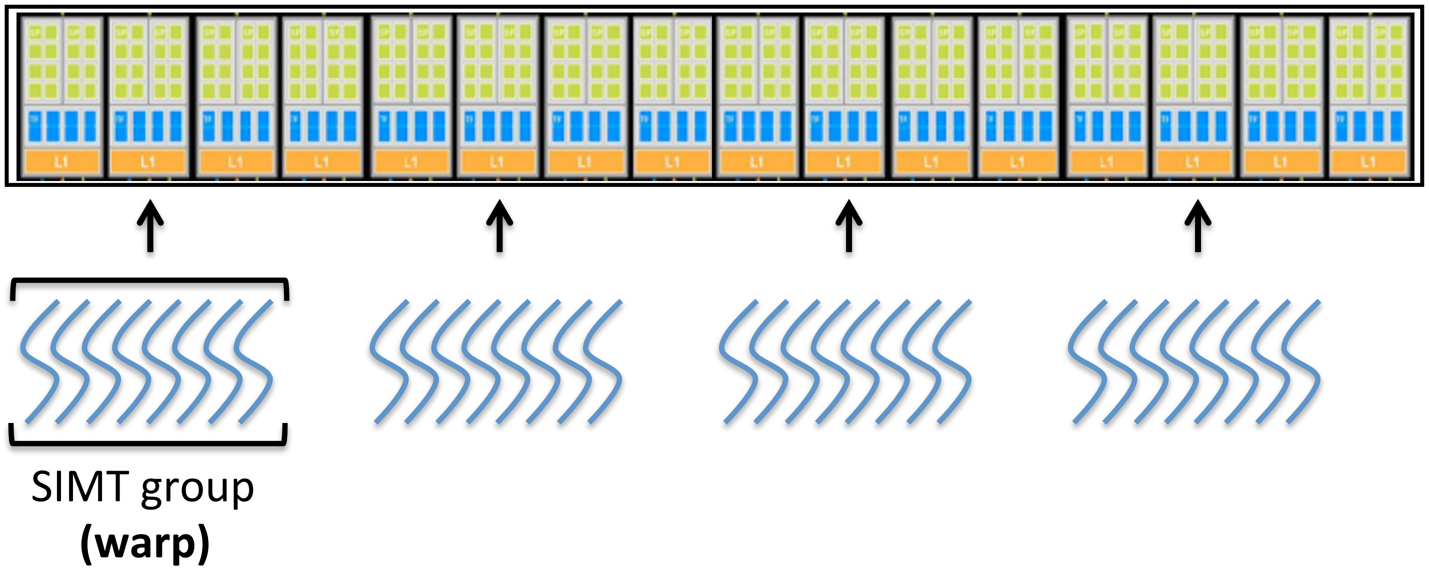
Batch Processing Pipeline



GASPP Framework

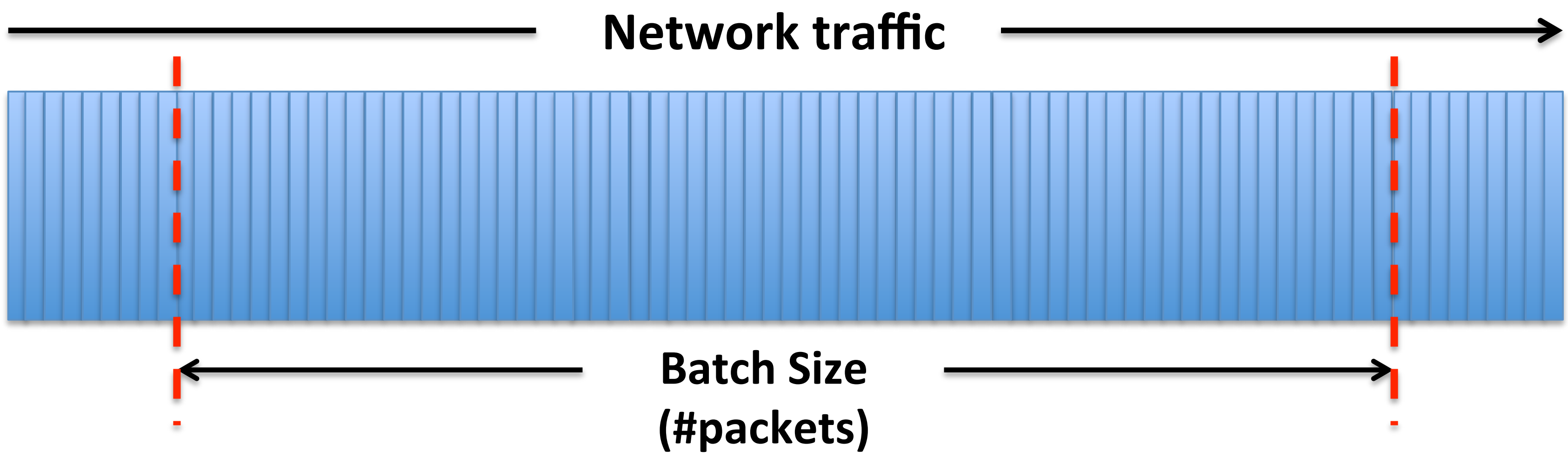
- Fast user-space packet capturing
- Modular and flexible
- Efficient packet scheduling mechanisms
- TCP processing and flow management support

Single **I**nstruction, **M**ultiple **T**hreads



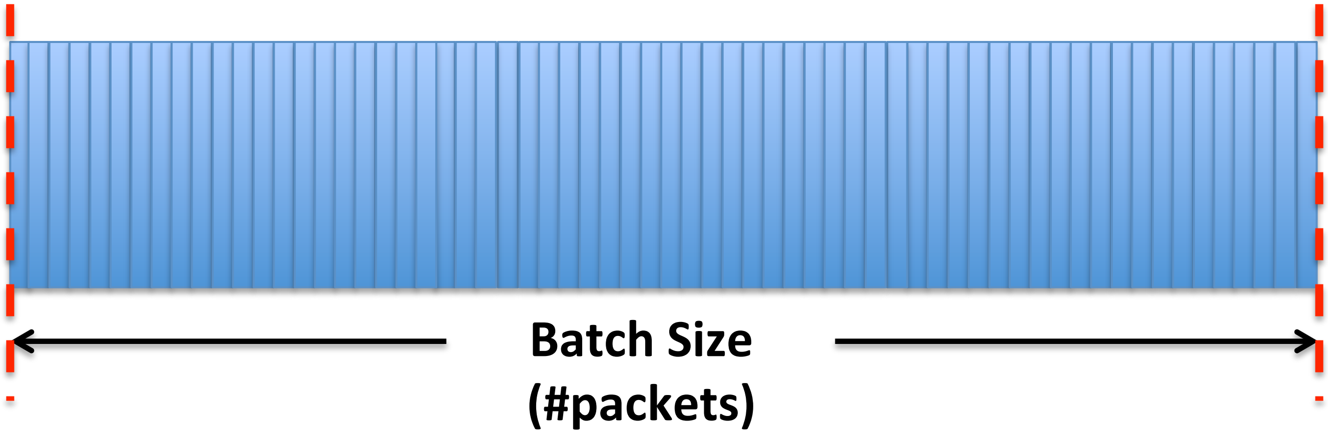
- Threads within the same ***warp*** have to execute the same instructions
- *Great for regular computations!*

Parallelism in packet processing



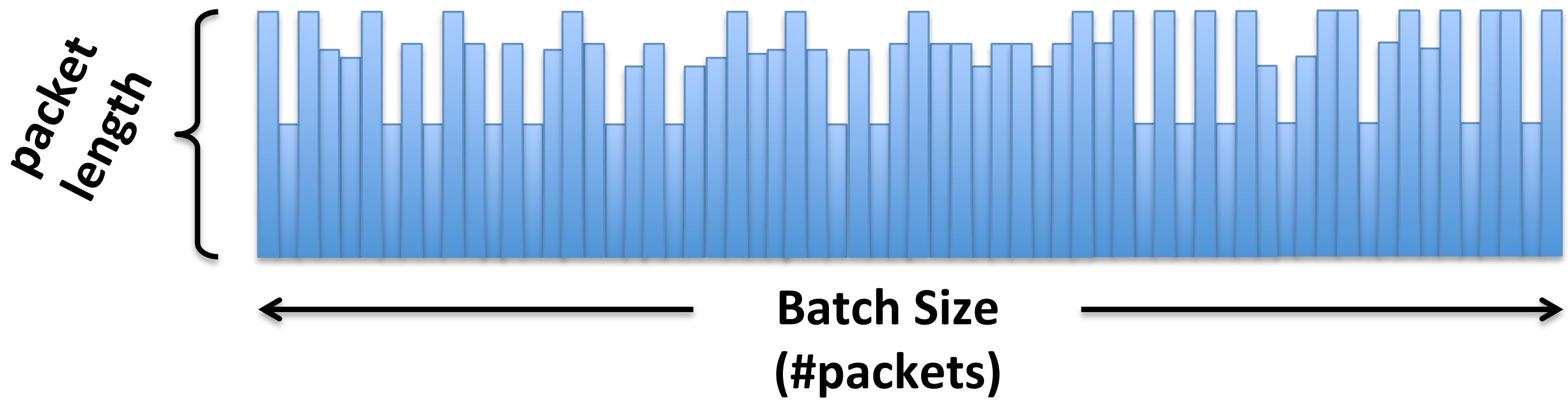
- Network packets are processed in batches
  - More packets => more parallelism

Dynamic Irregularities



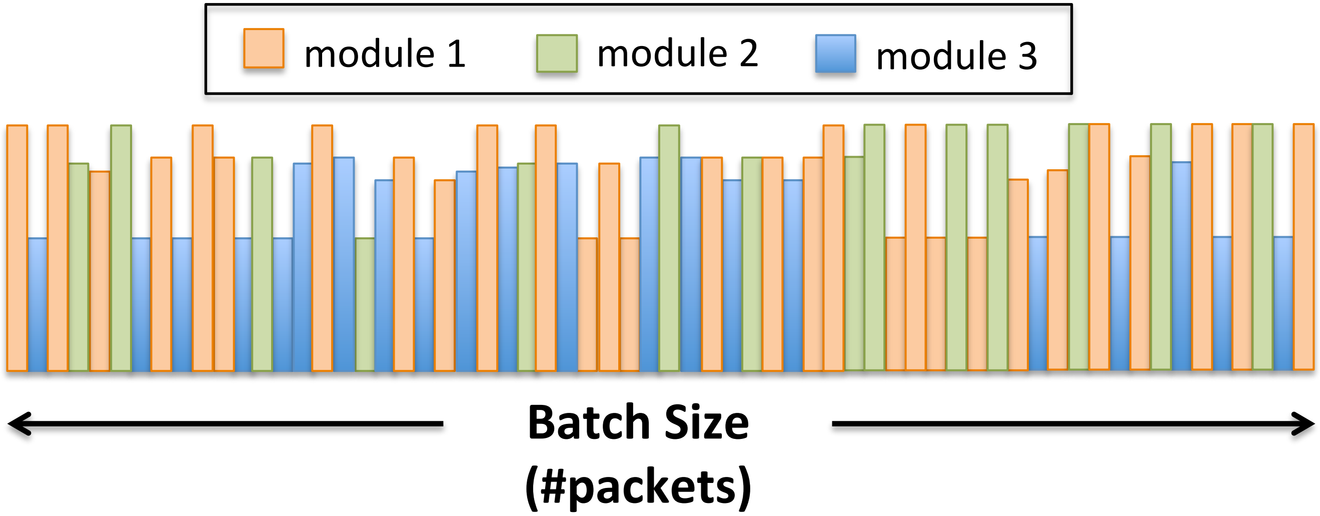
- Received network packets mix is very dynamic

Dynamic Irregularities



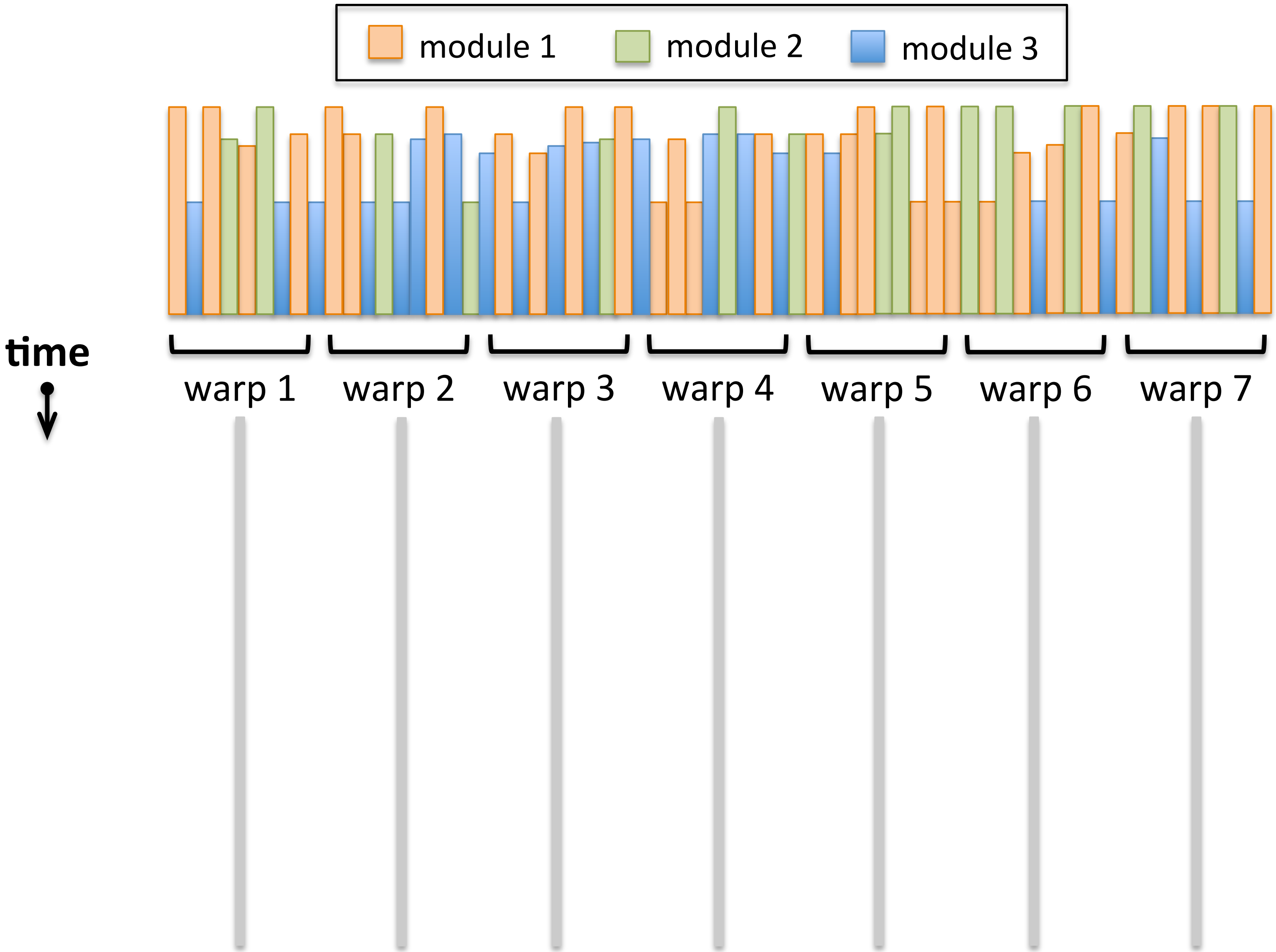
- Received network packets mix is very dynamic
  - *Different packet lengths*

Dynamic Irregularities

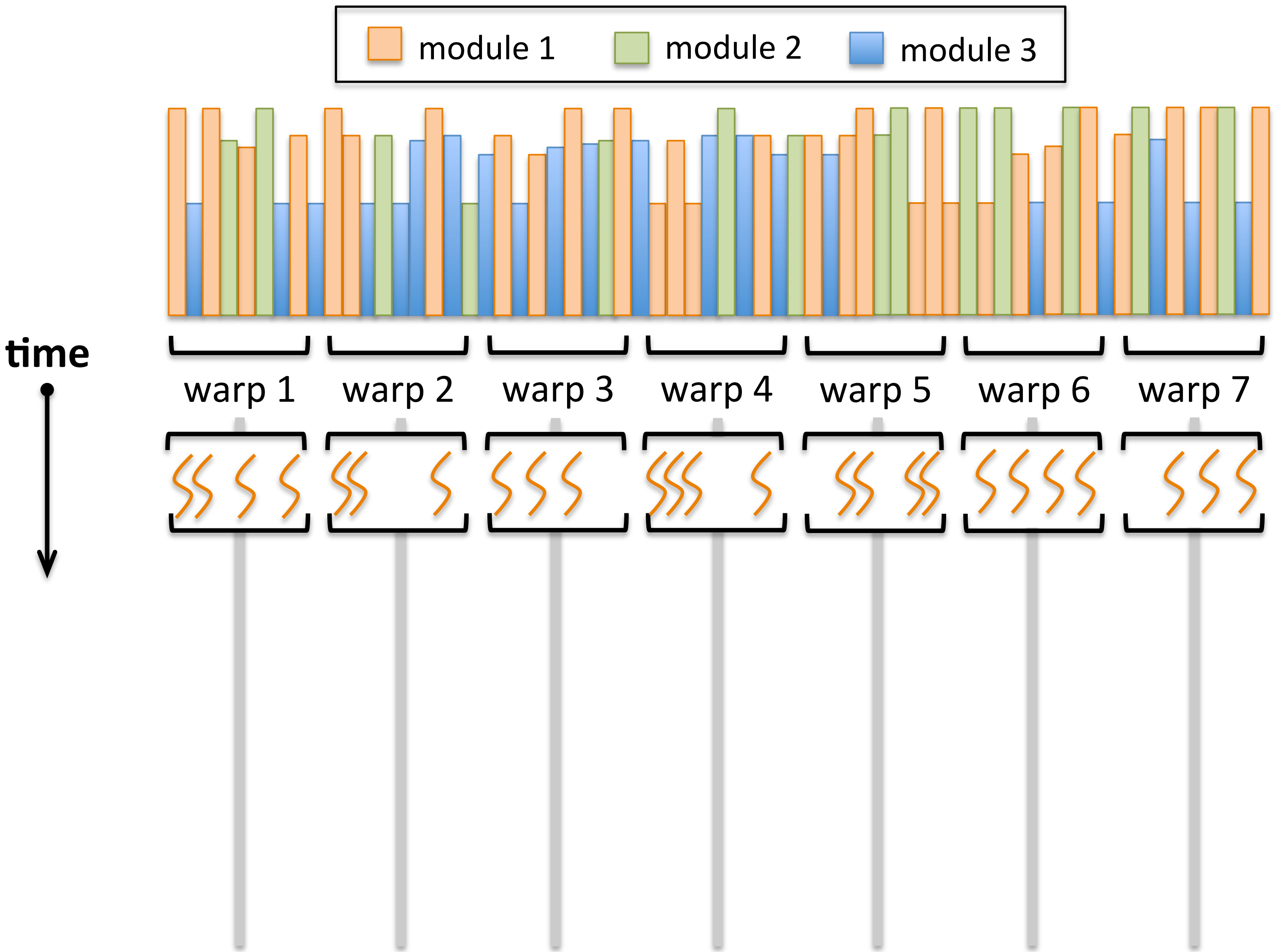


- Received network packets mix is very dynamic
  – *Different packet lengths*
  – *Divergent parallel module processing*

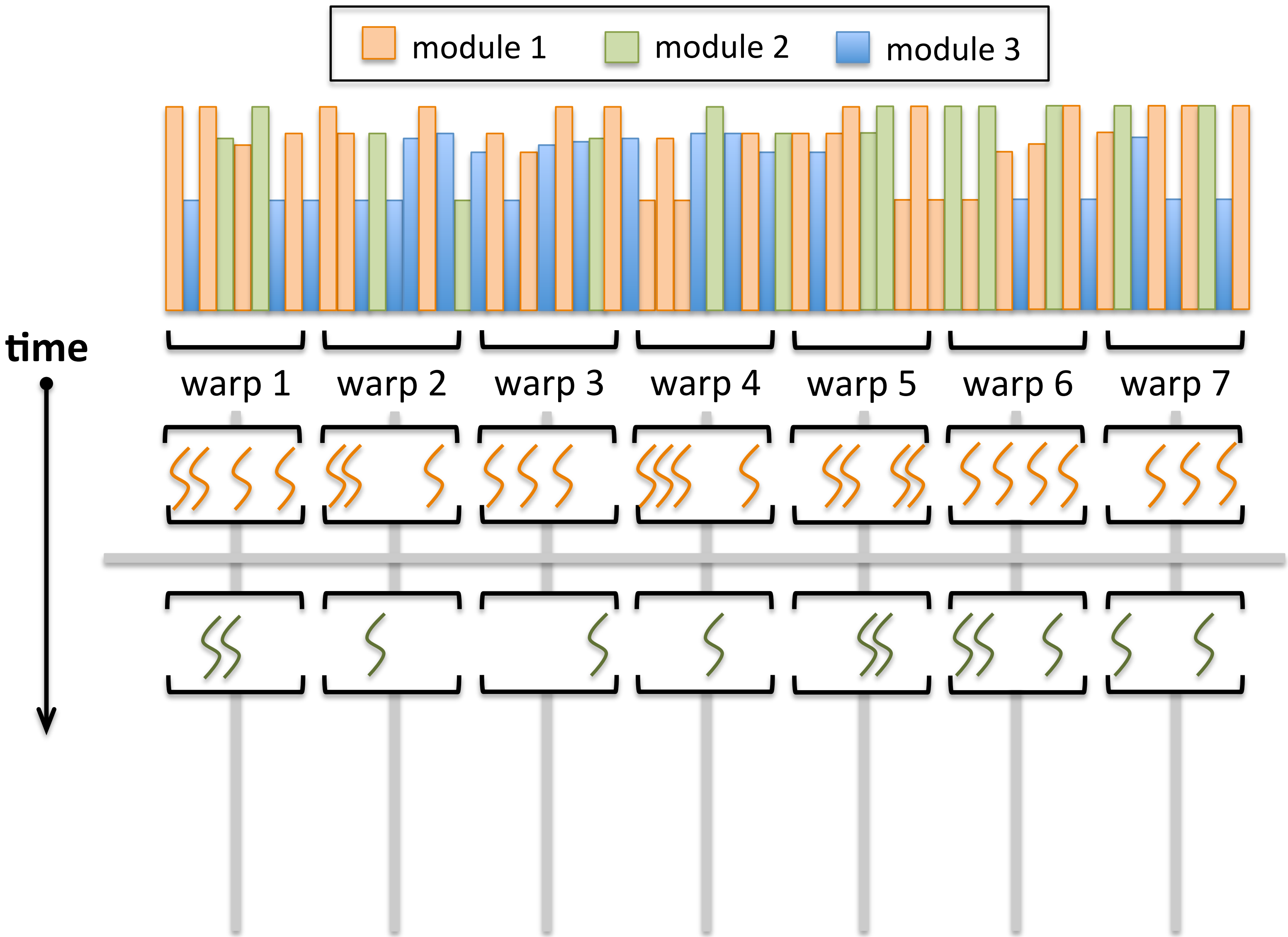
Dynamic Irregularities



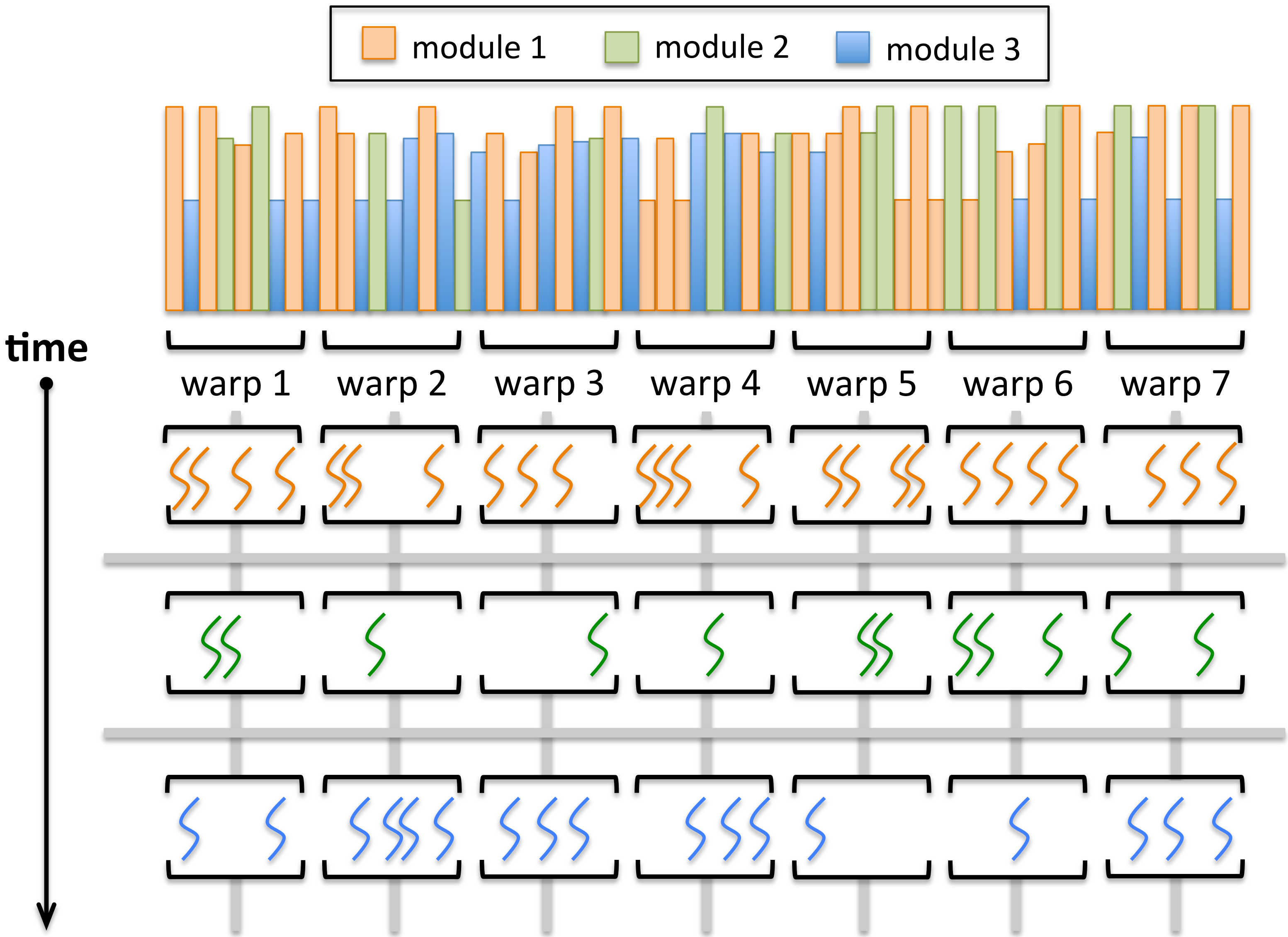
Dynamic Irregularities



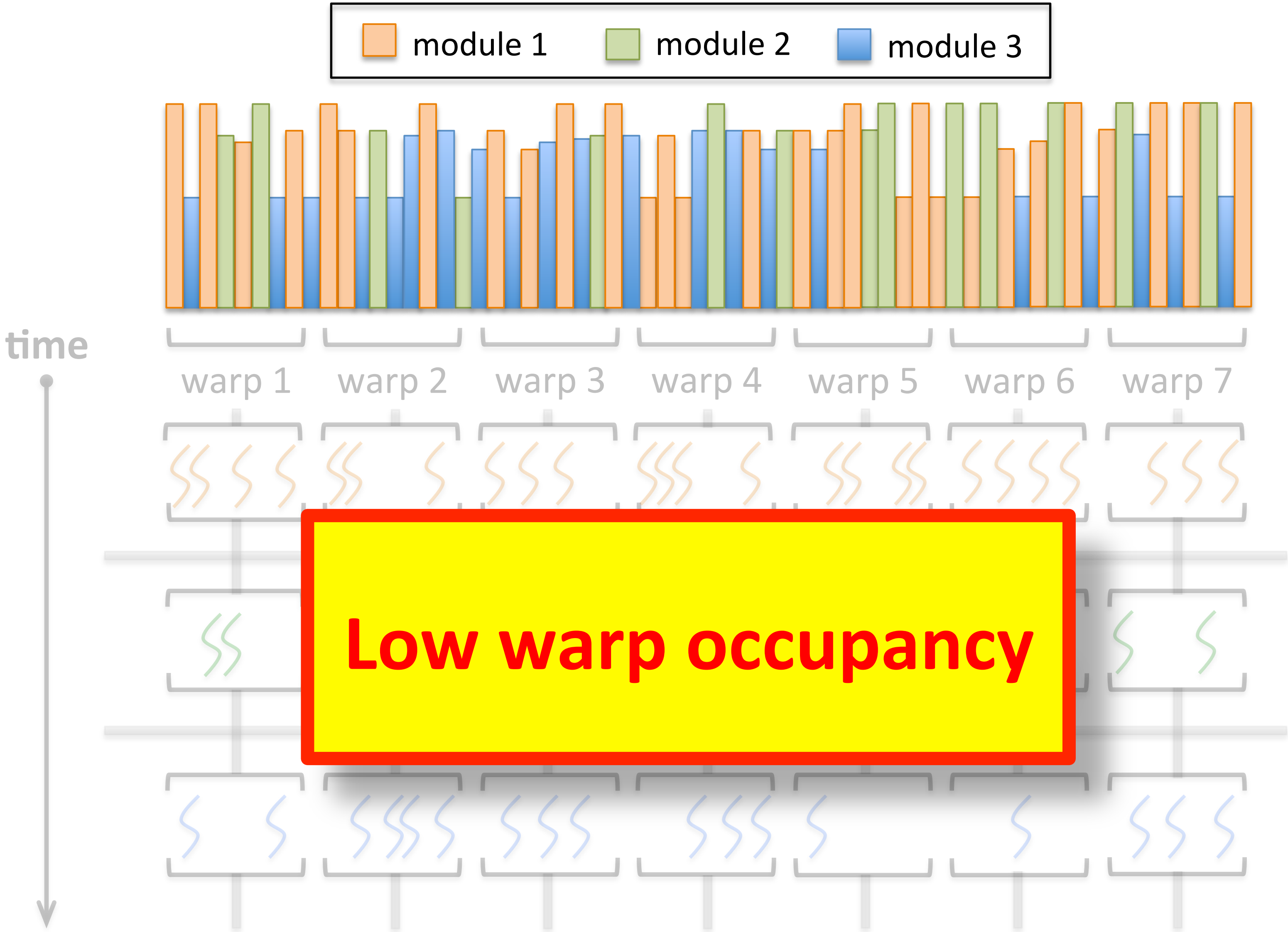
Dynamic Irregularities



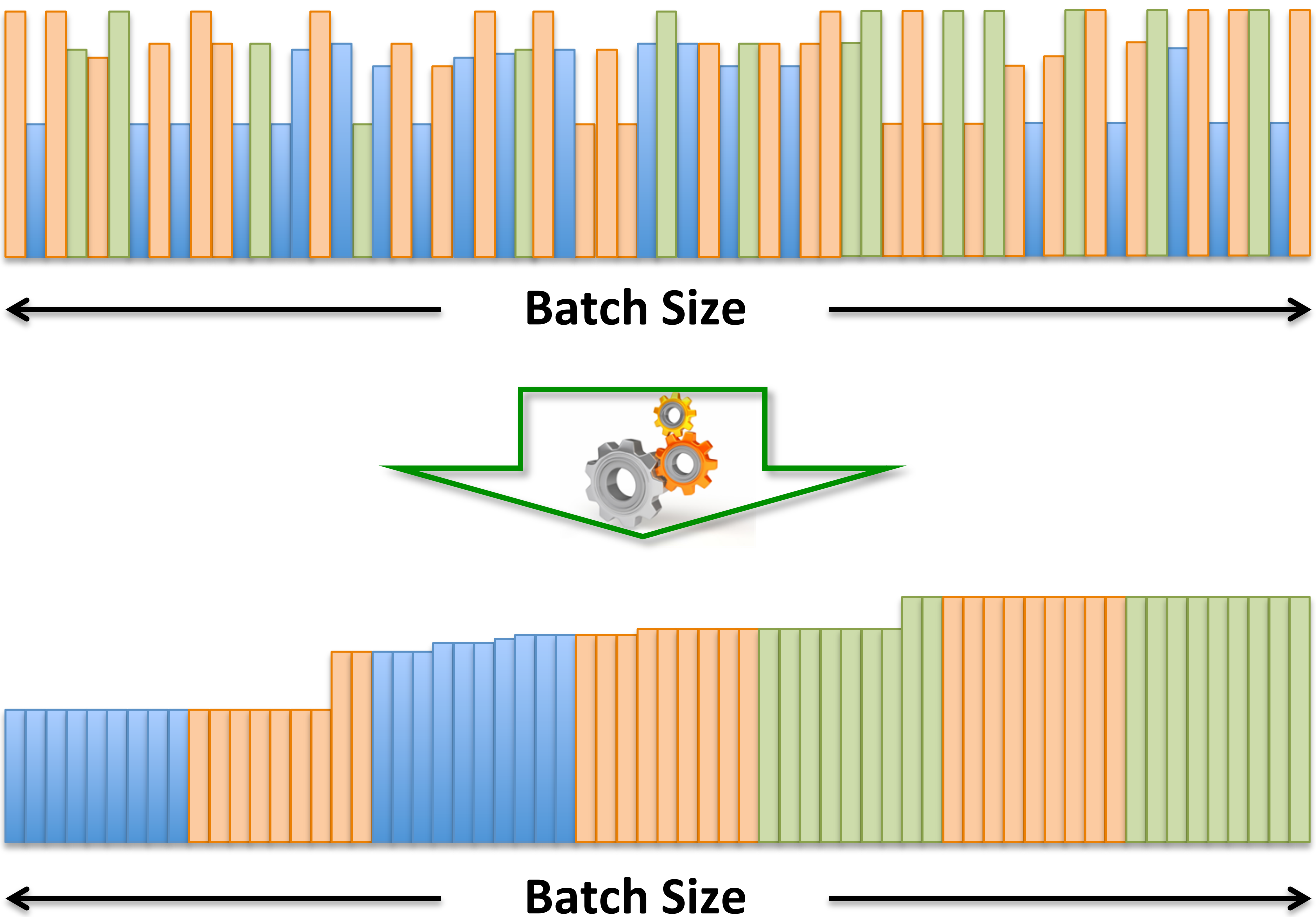
Dynamic Irregularities



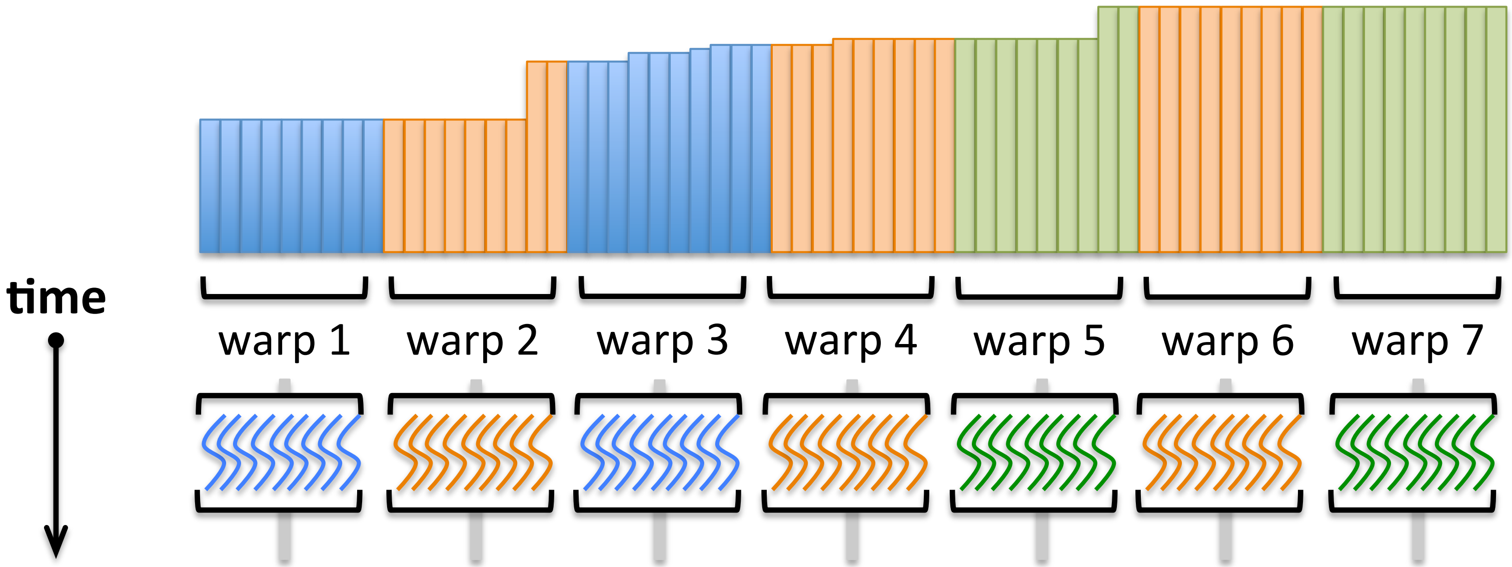
Dynamic Irregularities



Packet grouping



Packet grouping

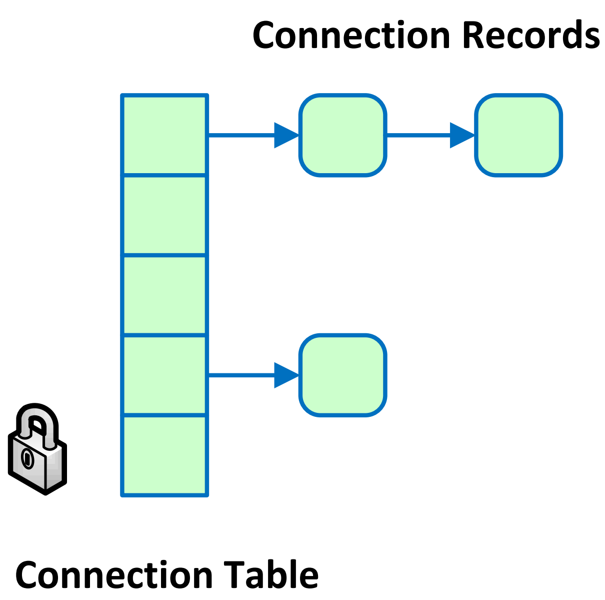


- ➢ Harmonized execution
- ➢ Symmetric processing

GASPP Framework

- Fast user-space packet capturing
- Modular and flexible
- Efficient packet scheduling mechanisms
- TCP processing and flow management support

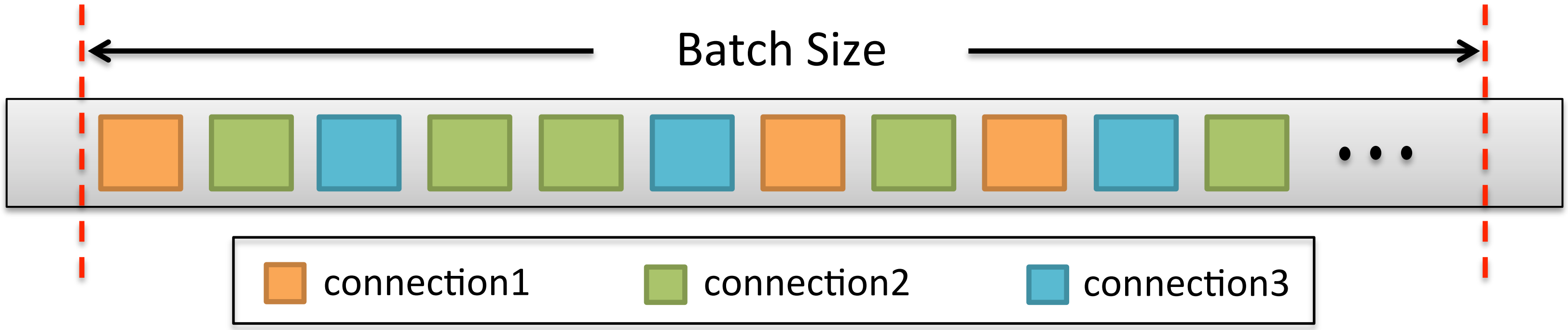
TCP Flow State Management



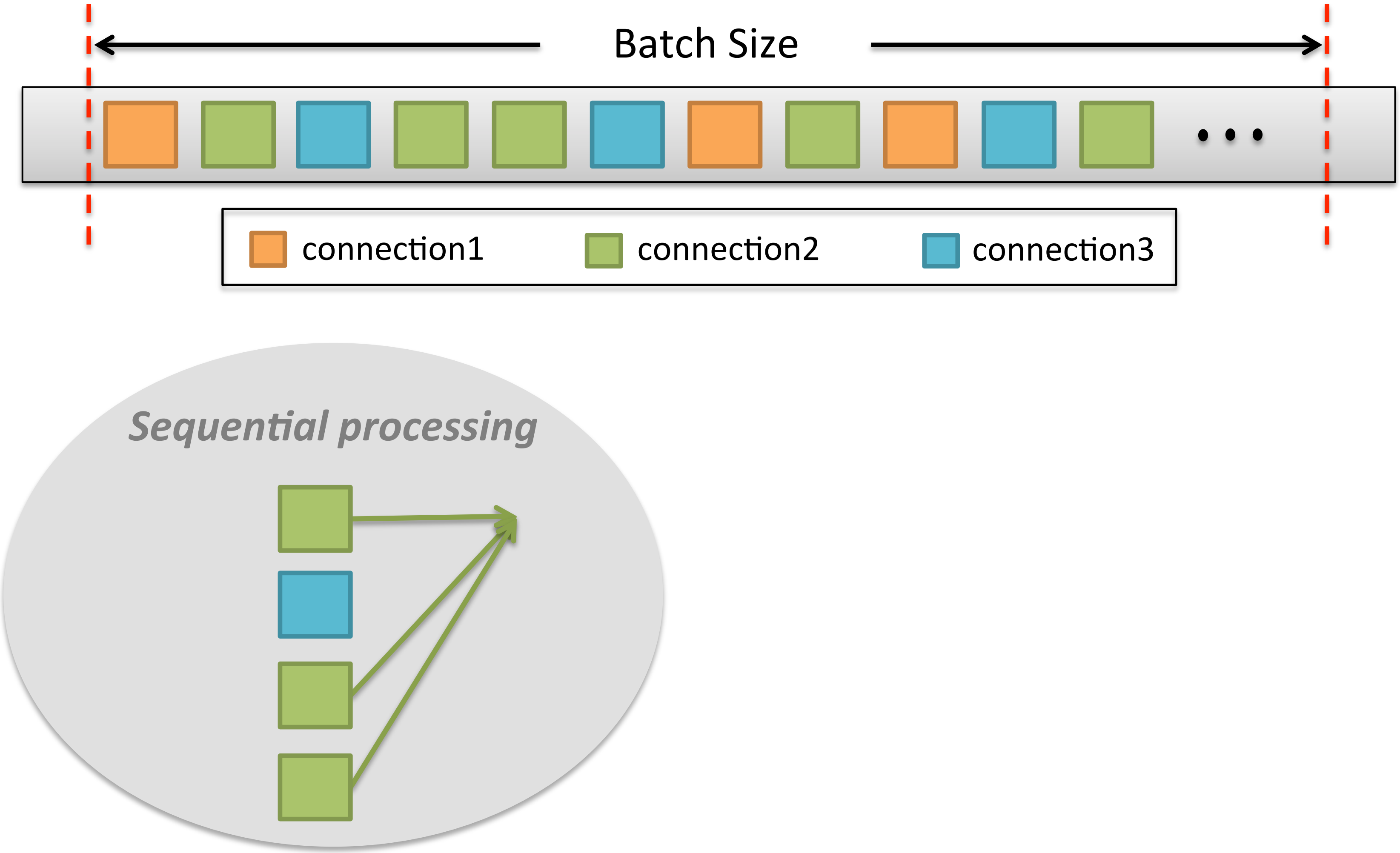
Connection Record	
Hash key	: 4 bytes
State	: 1 byte
Seq $_{CLIENT}$	: 4 bytes
Seq $_{SERVER}$	: 4 bytes
Next	: 4 bytes

- Maintain the state of TCP connections

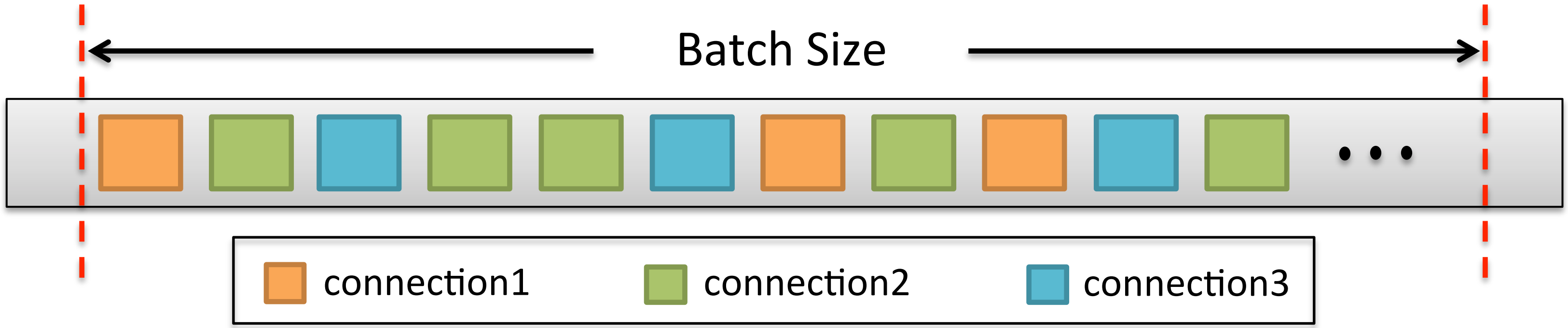
TCP Stream Reassembly

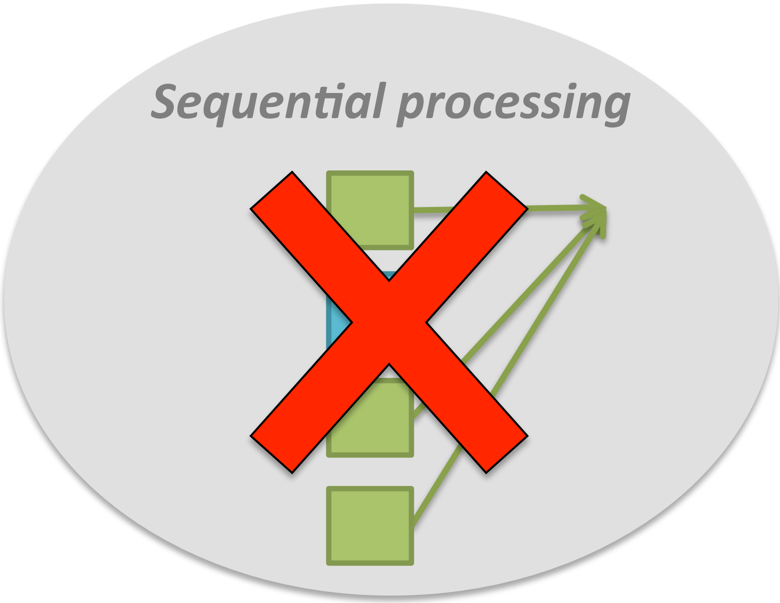


TCP Stream Reassembly

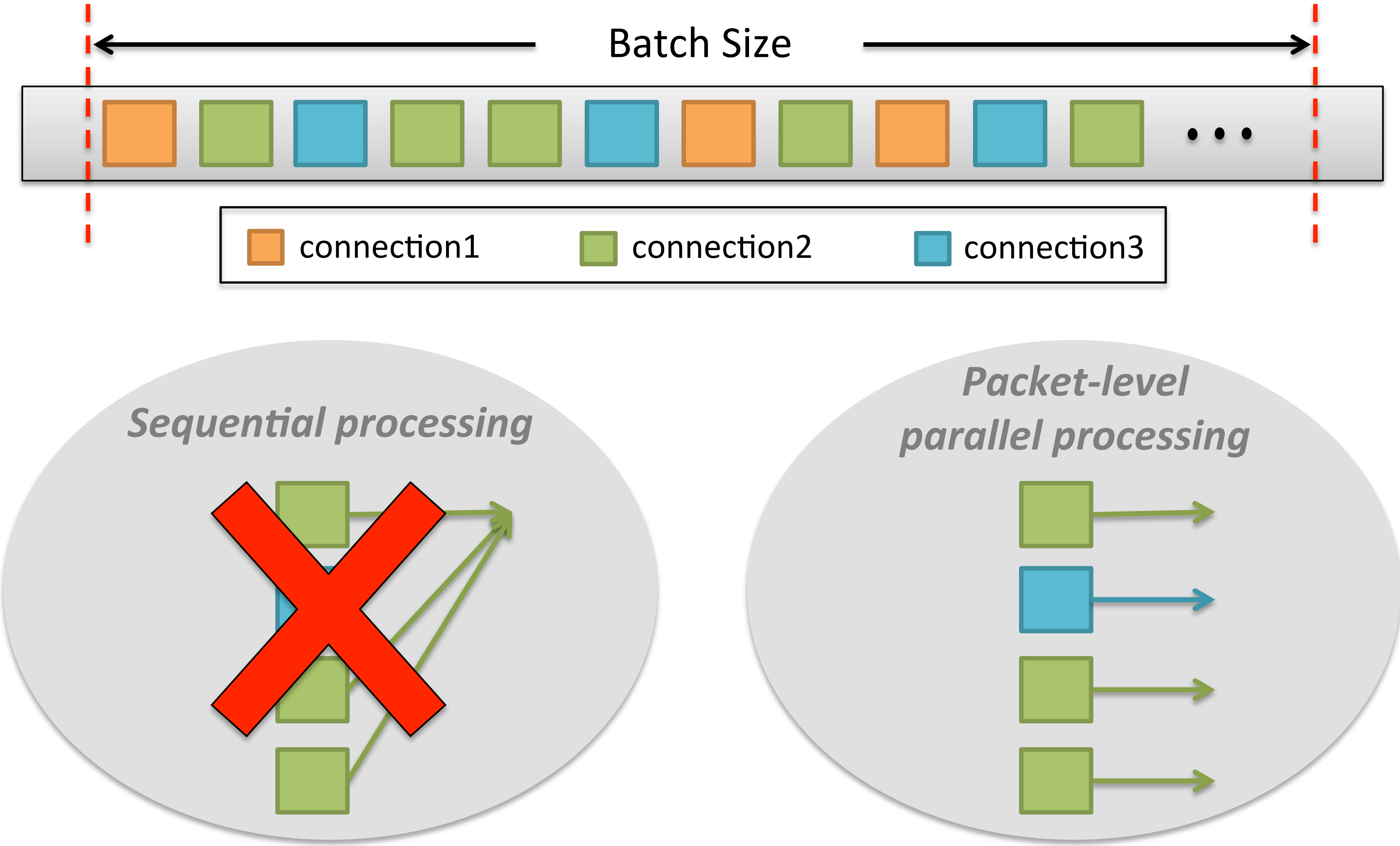


TCP Stream Reassembly

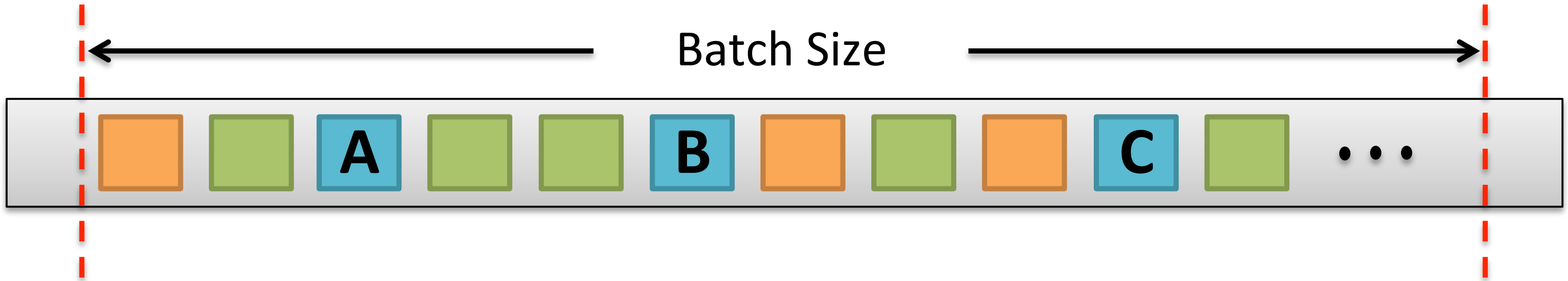




TCP Stream Reassembly

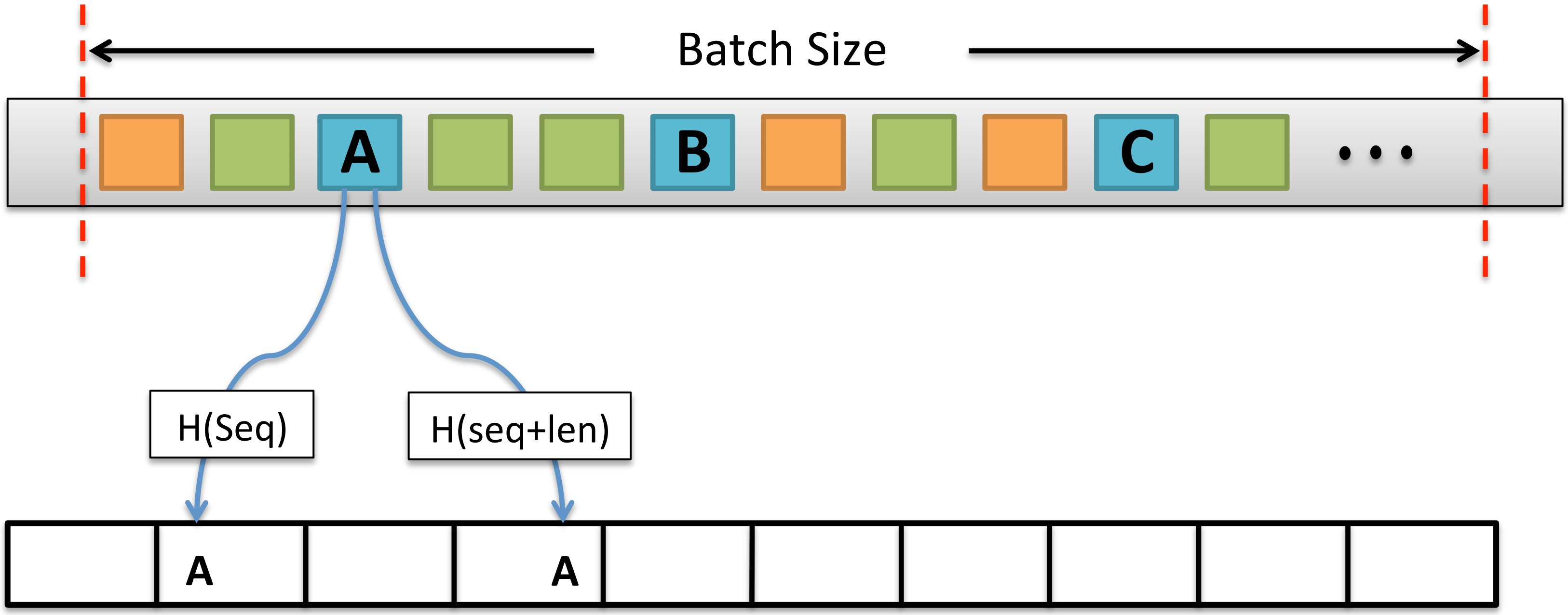


TCP Stream Reassembly

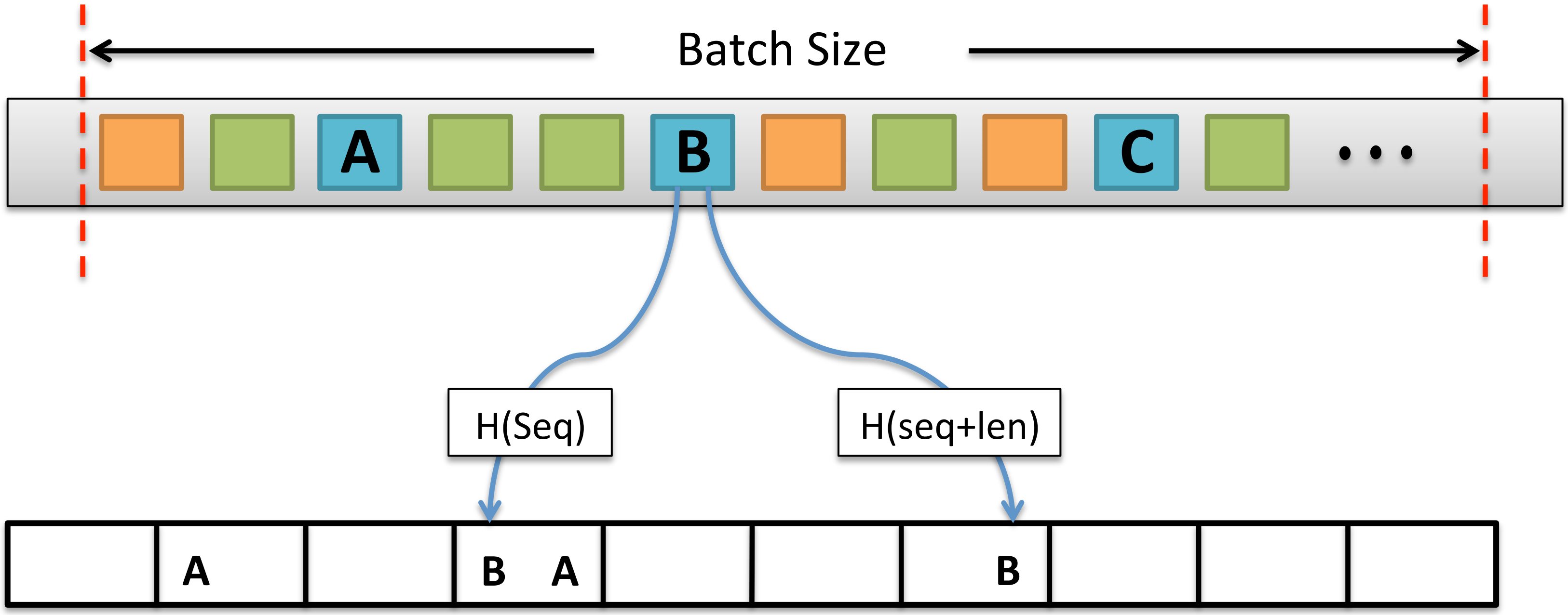


- Key insight
  - Packets $<A, B>$ are consecutive if $Seq_B = (Seq_A + len_A)$

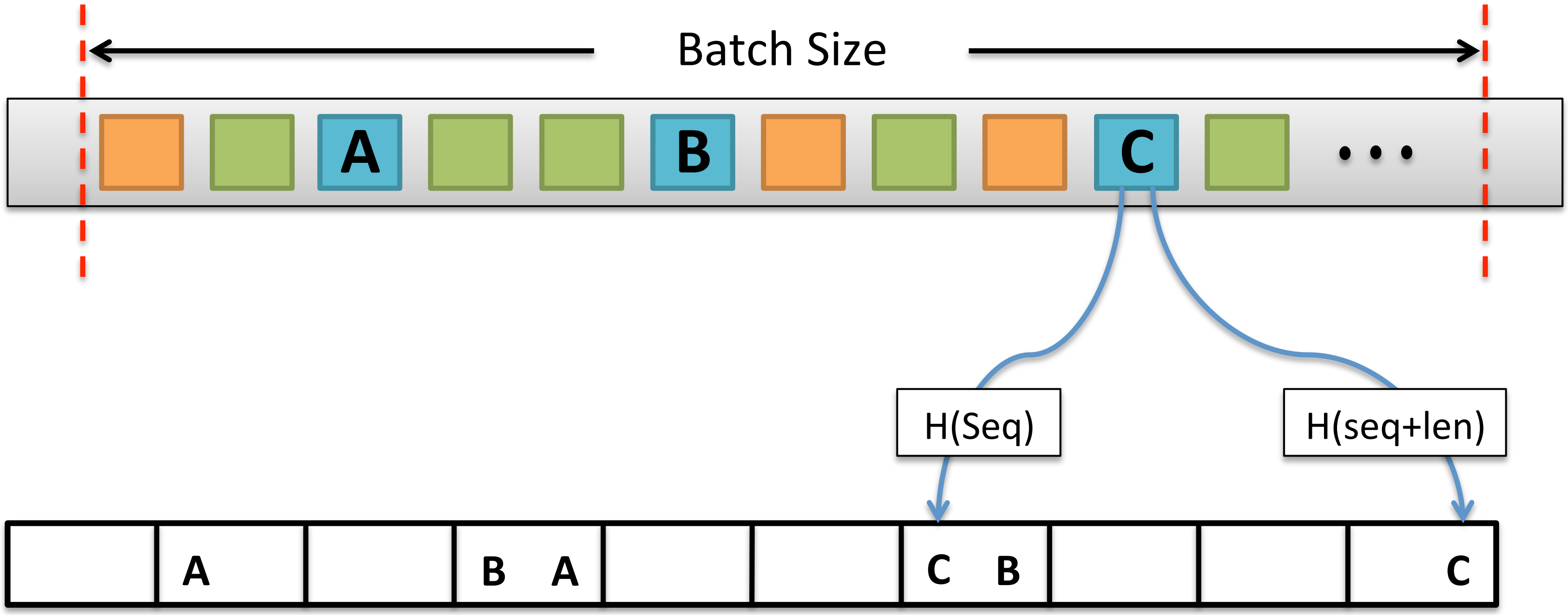
TCP Stream Reassembly



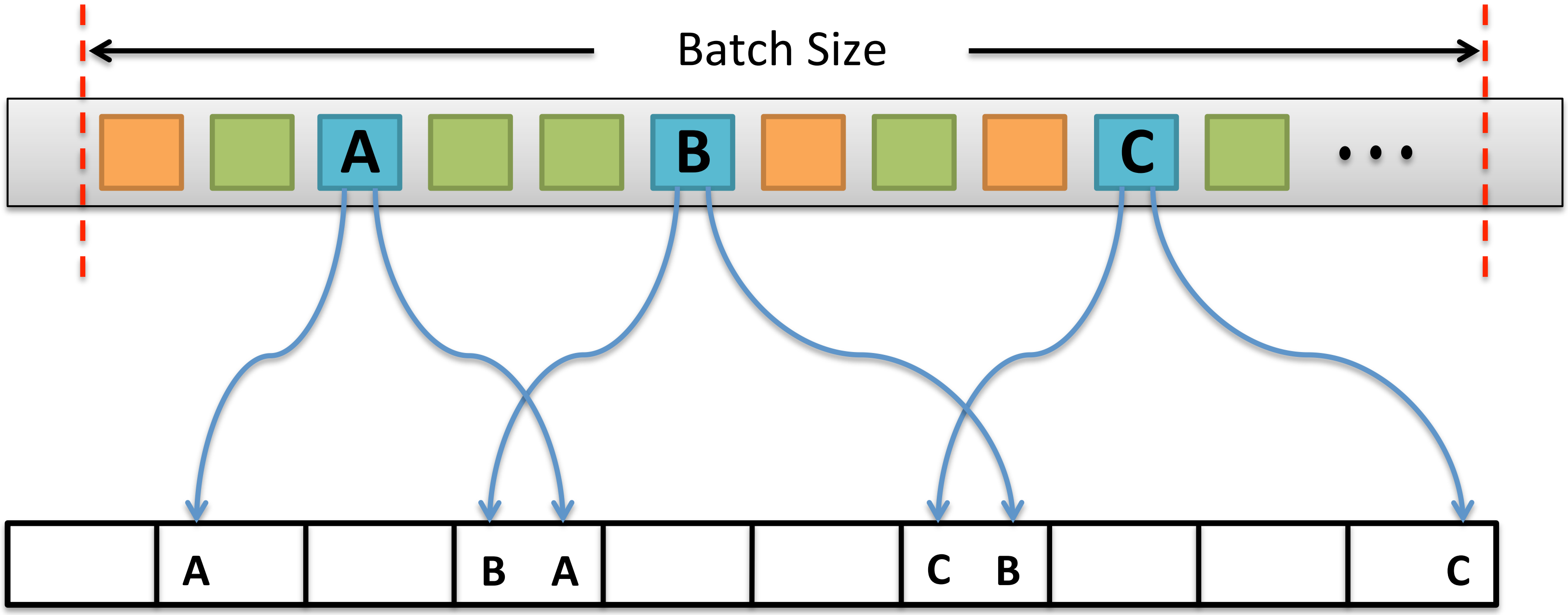
TCP Stream Reassembly



TCP Stream Reassembly

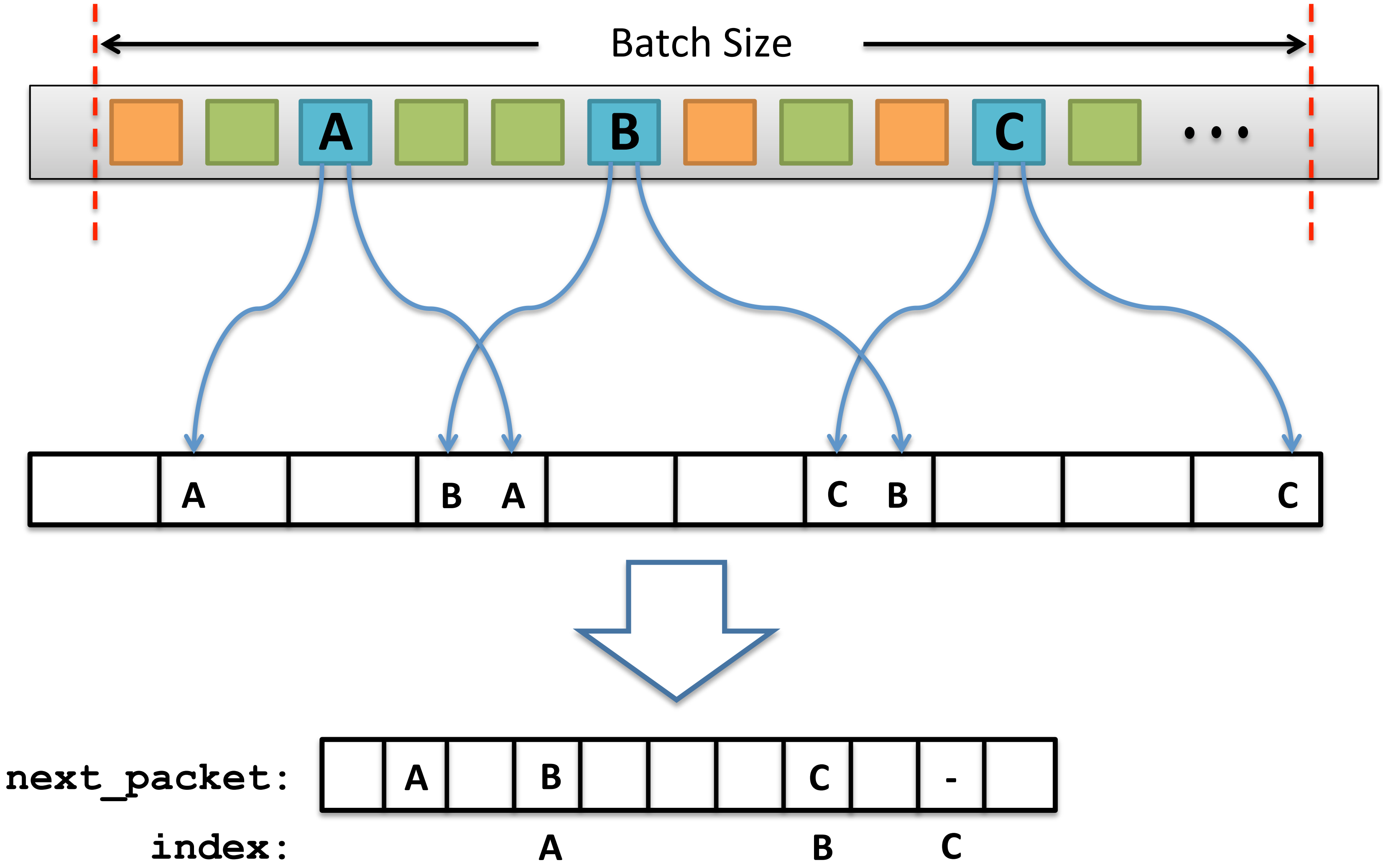


TCP Stream Reassembly



Parallel Processing

TCP Stream Reassembly



Other TCP corner cases

- TCP sequence holes
- Out-of-order packets

Other TCP corner cases

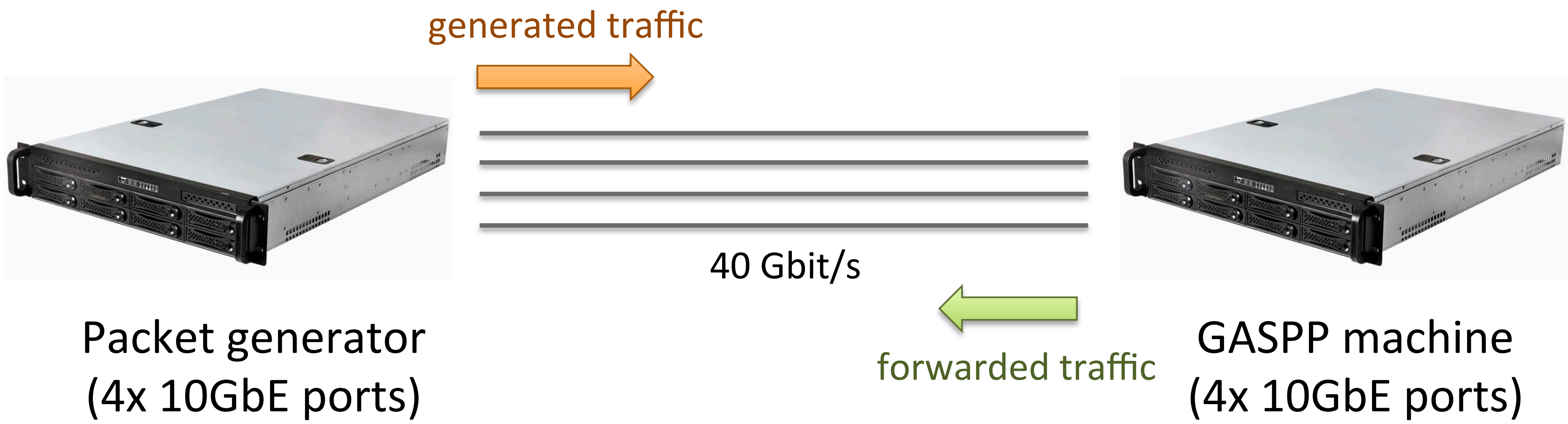
- TCP sequence holes
- Out-of-order packets

Read the paper for details

Evaluation

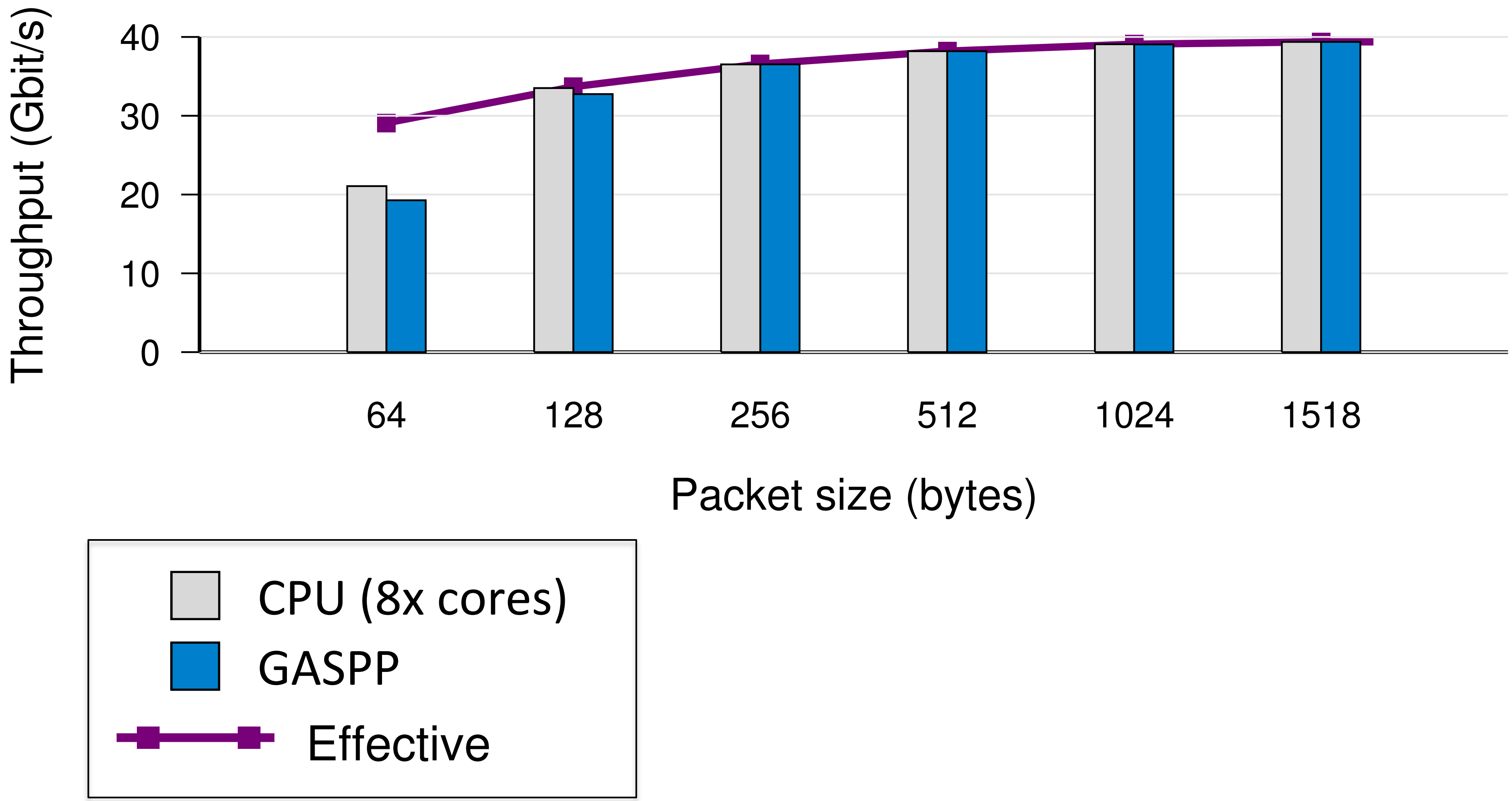
- Forwarding
- Latency
- Individual Applications
- Consolidated applications

Evaluation Setup



- GASPP machine has:
  - 2x NUMA nodes (Intel Xeon E5520 2.27GHz quad-core CPUs)
  - 2x banks of 6GB of DDR3 1066MHz RAM
  - 2x Intel 82599EB network adapters (with dual 10GbE ports)
  - 2x NVIDIA GTX480 graphics cards

Basic Forwarding



Latency

- 8192 batch:
  - CPU: 0.48 us
  - GASPP: 3.87 ms
- 1024 batch: 0.49 ms
  - Same performance for basic forwarding
  - …but 2x-4x throughput slowdown for heavyweight processing applications

Individual Applications

- Each application is written as a GPU kernel
  - No CPU-side development

- Speedup over a single CPU-core

Applications	GASPP (8192 batch)	GASPP (1024 batch)
Firewall	3.6x	3.6x
StringMatch	28.4x	9.3x
RegExMatch	173.1x	36.9x
AES	14.6x	6.5x

Consolidating Applications

	Firewall	Firewall StringMatch	Firewall StringMatch RegExMatch	Firewall StringMatch RegExMatch AES
GASPP:	-	1.19x	2.12x	1.93x

GASPP reduces irregular execution by 1.19-2.12X

Conclusions

- What we offer:
  - Fast inter-device data transfers
  - GPU-based flow state management and stream reconstruction
  - Efficient packet scheduling mechanisms
- Limitations
  - High packet processing latency

GASPP: A GPU-Accelerated Stateful Packet Processing Framework

Giorgos Vasiliadis, ***FORTH-ICS, Greece***

Lazaros Koromilas, *FORTH-ICS, Greece*

Michalis Polychronakis, *Columbia University, USA*

Sotiris Ioannidis, *FORTH-ICS, Greece*