ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

A Dynamic Hardware Architecture for Future Networks

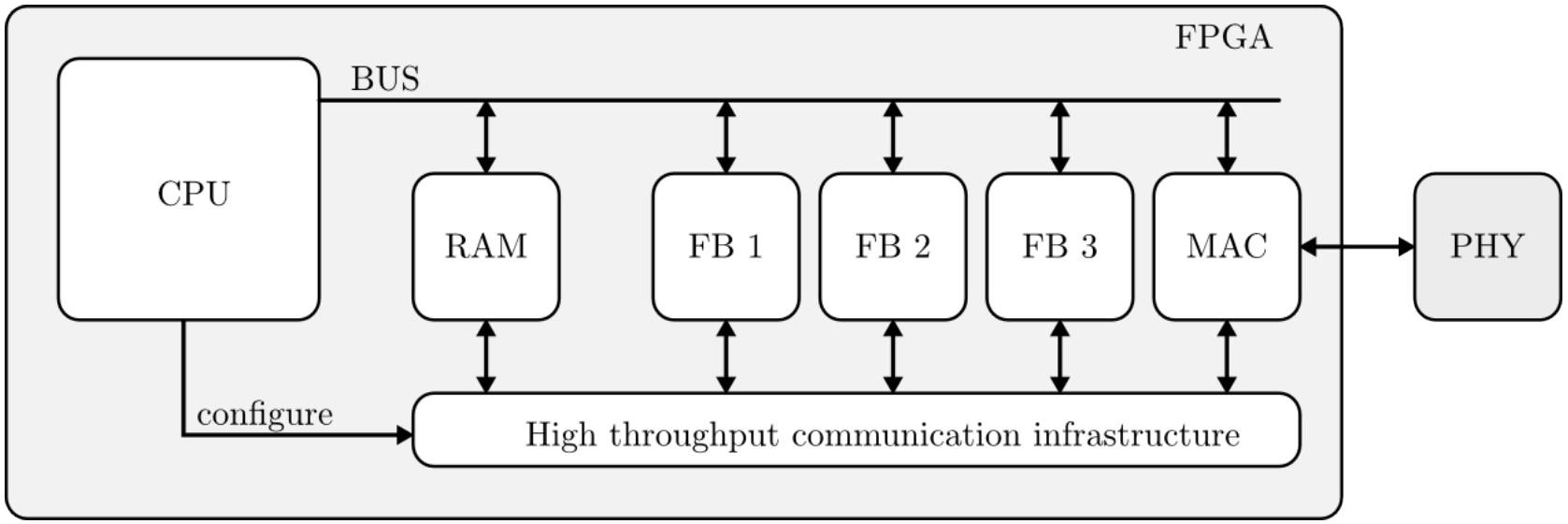
Master Thesis by Richard Huber

2nd Intermediate Presentation

Advisor: Ariane Keller
Co-Advisors: Dr. Stephan Neuhaus, Daniel Borkmann
Professor: Prof. Dr. Bernhard Plattner

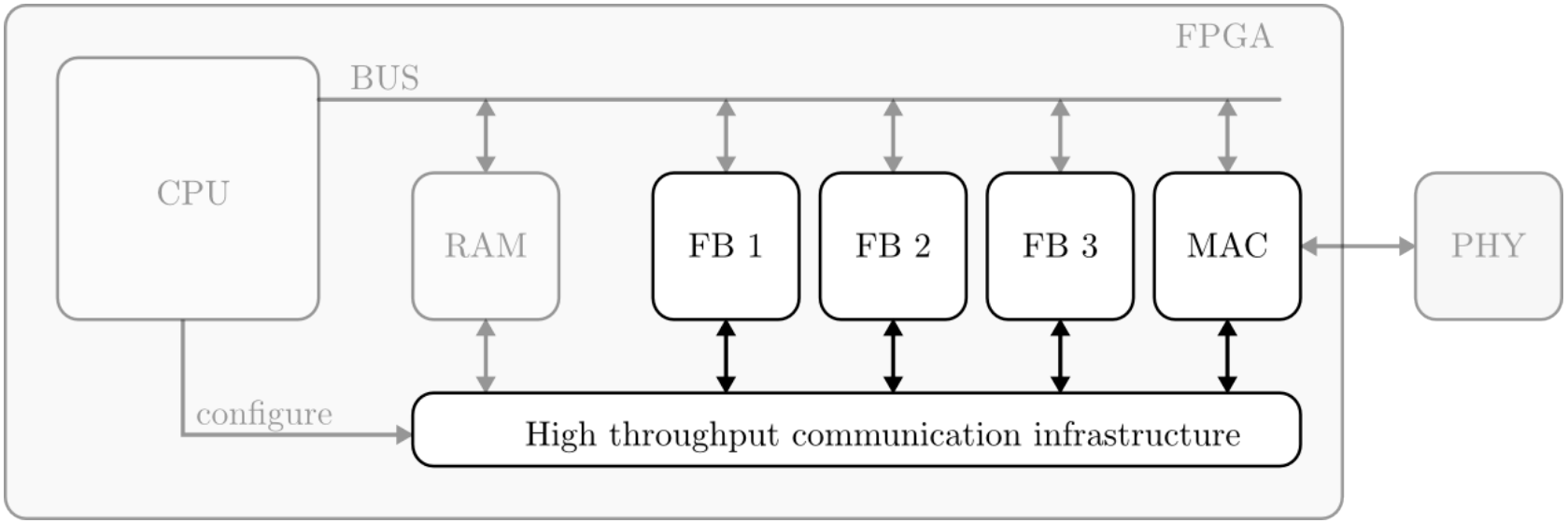
Summary of 1st presentation

- ReconOS: Hardware acceleration with OS resources
- ANA: Autonomous network architecture
- Problem: Shared bus is bottleneck



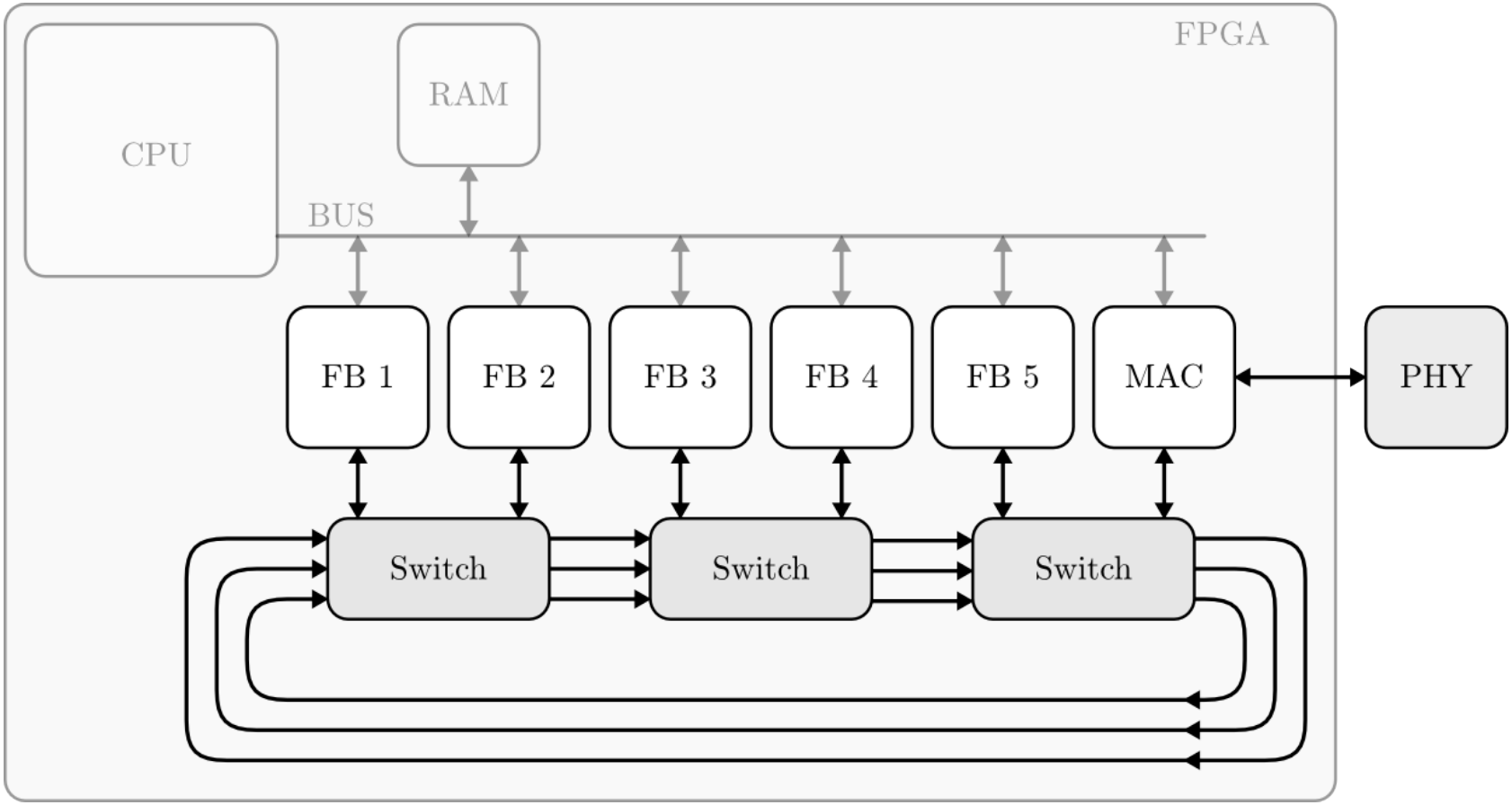
Summary of 1st presentation

- ReconOS: Hardware acceleration with OS resources
- ANA: Autonomous network architecture
- Problem: Shared bus is bottleneck



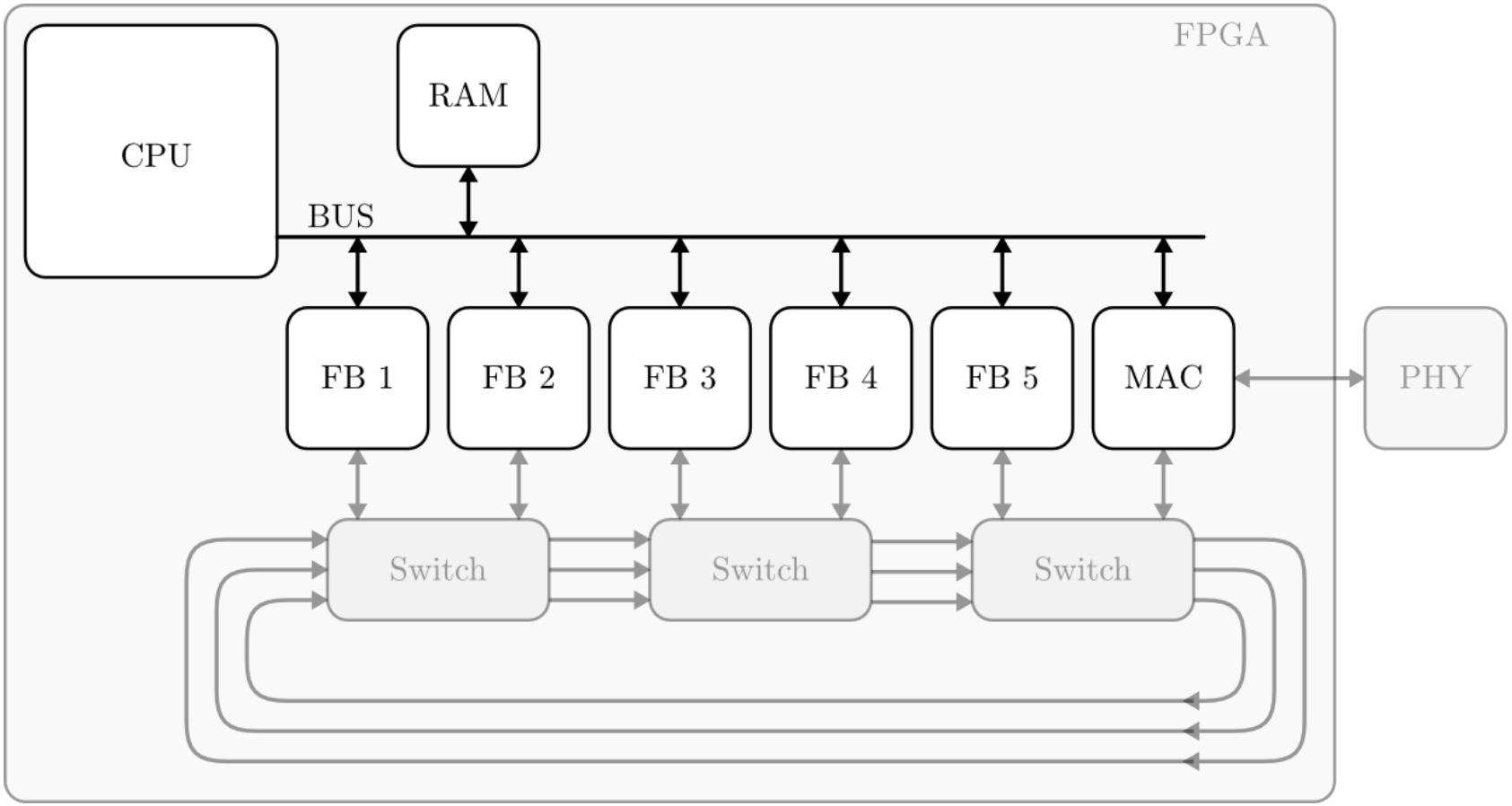
Network on Chip (NoC) Architecture

- NoC implemented and tested



Hardware – Software Data Transfer

- Transfer data between HW- and SW-Threads

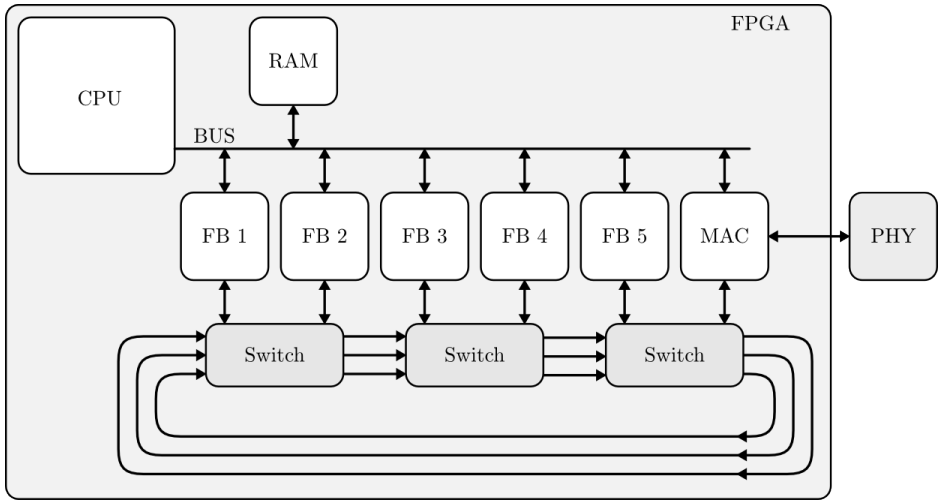


Hardware – Software Data Transfer

- Problems:
  - Synchronization
  - Serialization
  - High interrupt overhead
- ReconOS provides:
  - Shared memory for HW- and SW-Threads
  - Thread-save message passing interface

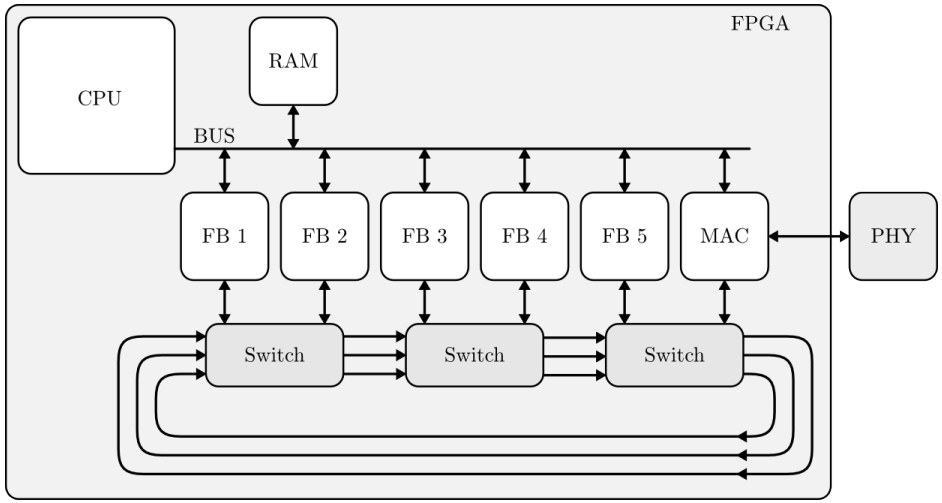
Hardware – Software Data Transfer

- Two possible approaches
  a) All FB read/write data to RAM
  b) A Pair of dedicated HW threads build a HW/SW gateway

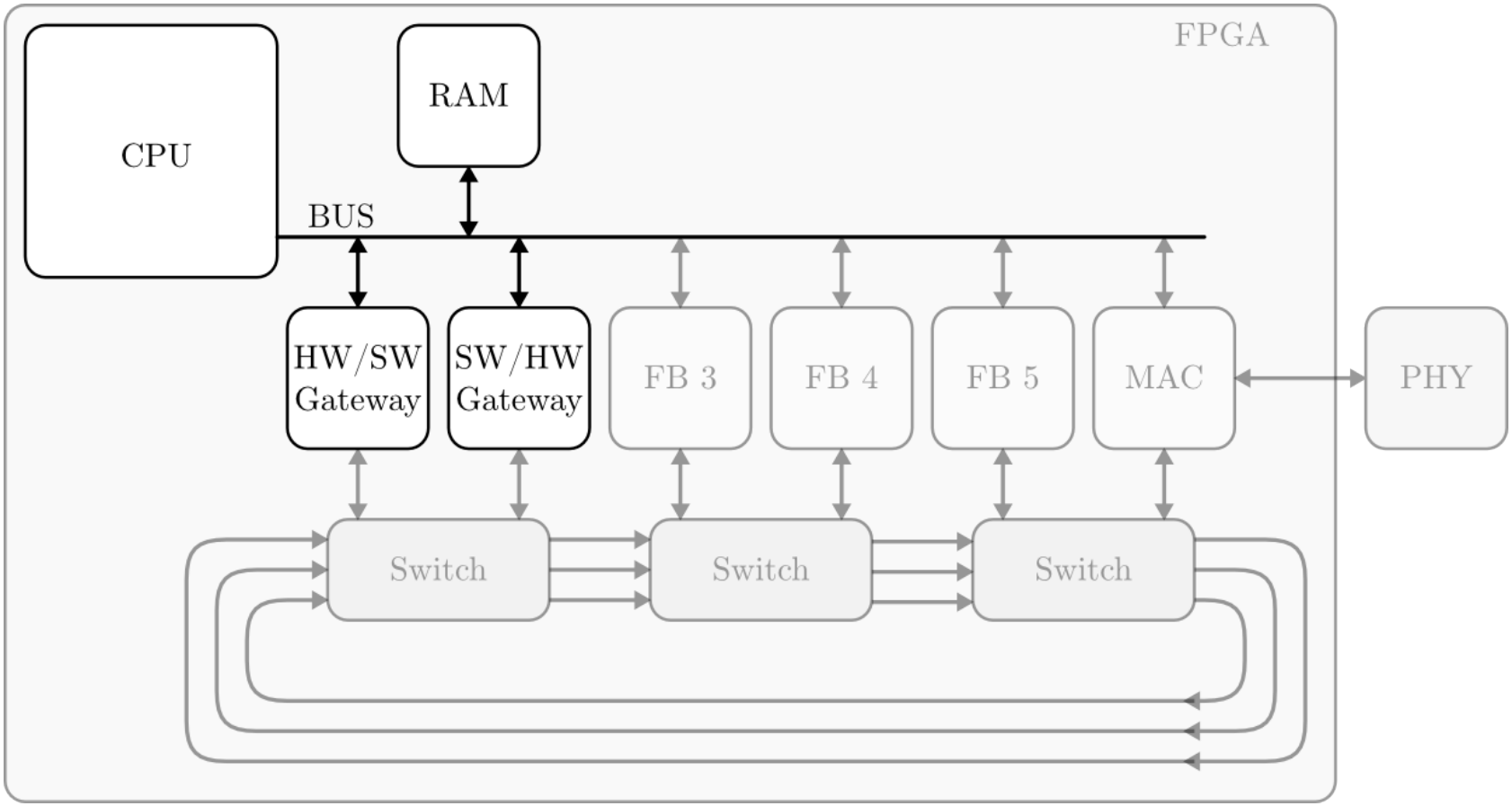


Hardware – Software Data Transfer

- Two possible approaches
  a) All FB read/write data to RAM
     - More interrupts
  b) A Pair of dedicated HW threads build a HW/SW gateway
     - Higher latency

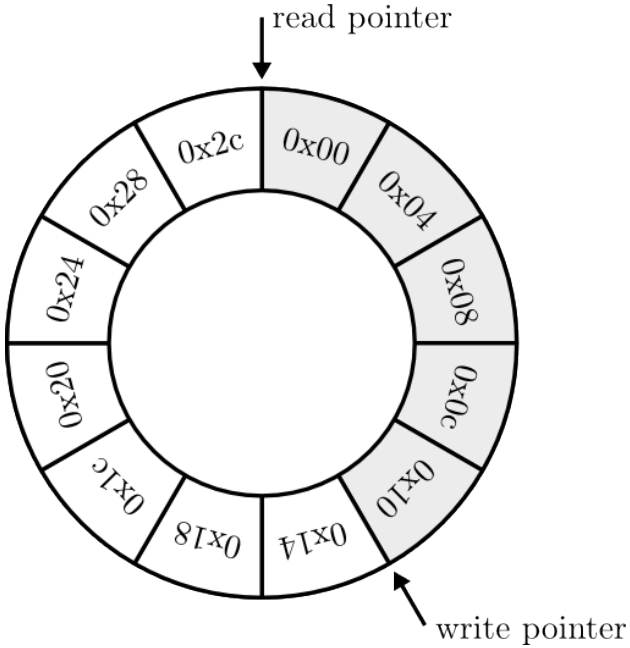


Hardware – Software Data Transfer

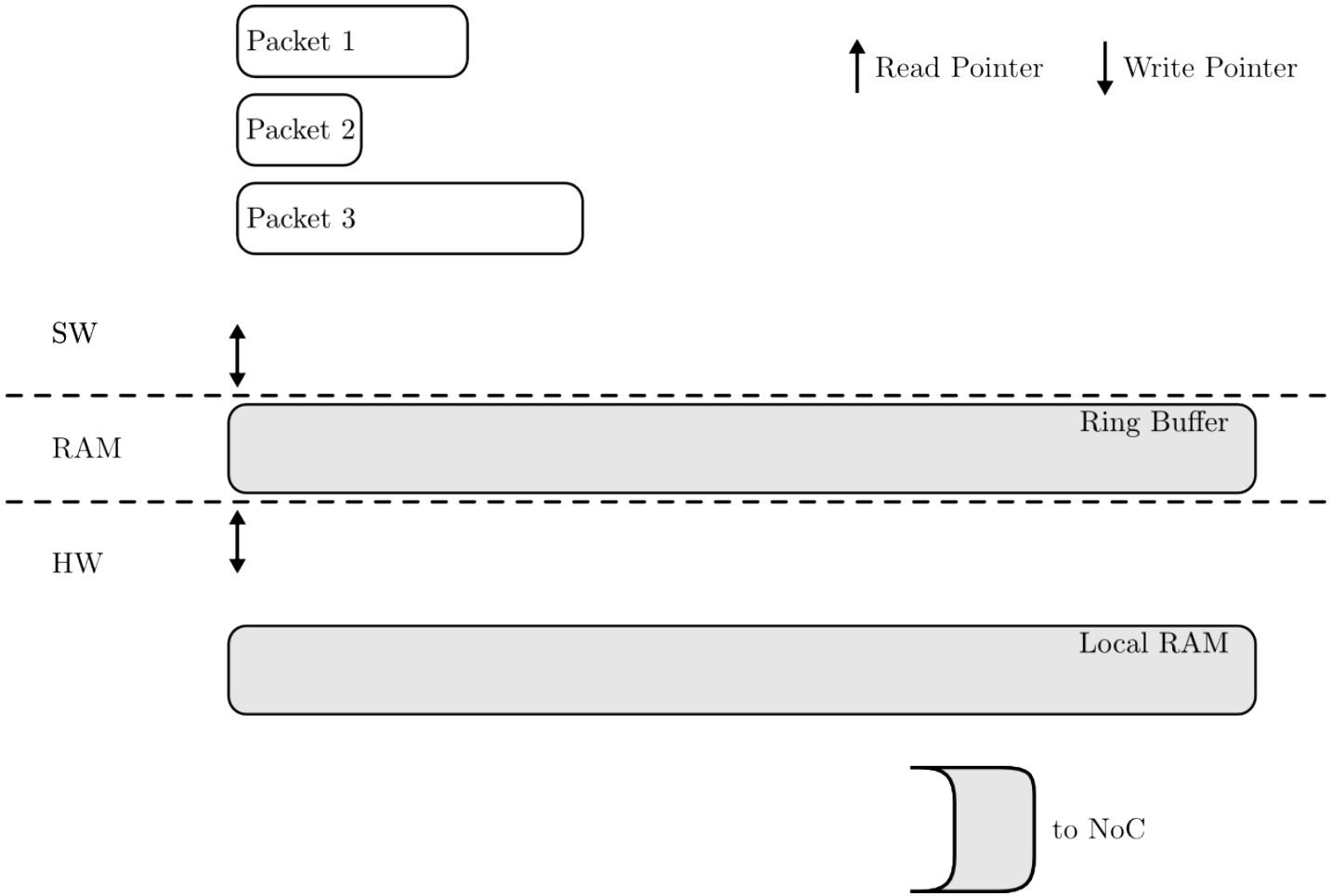


Principle of SW to HW gateway

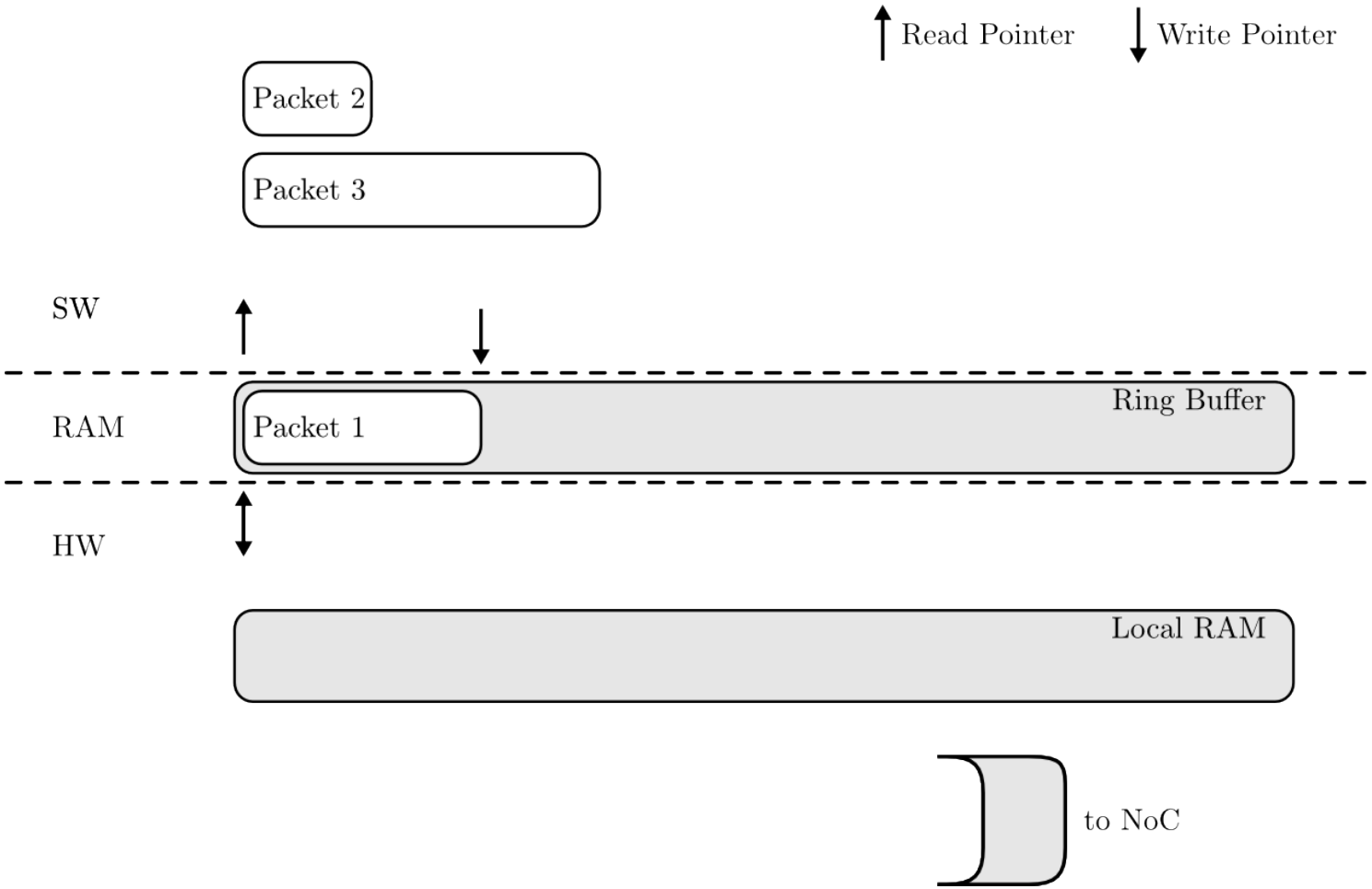
- Use a part of the shared memory as ring buffer
- Exchange read/write pointers using message boxes
- HW to SW gateway similar, just the other way around



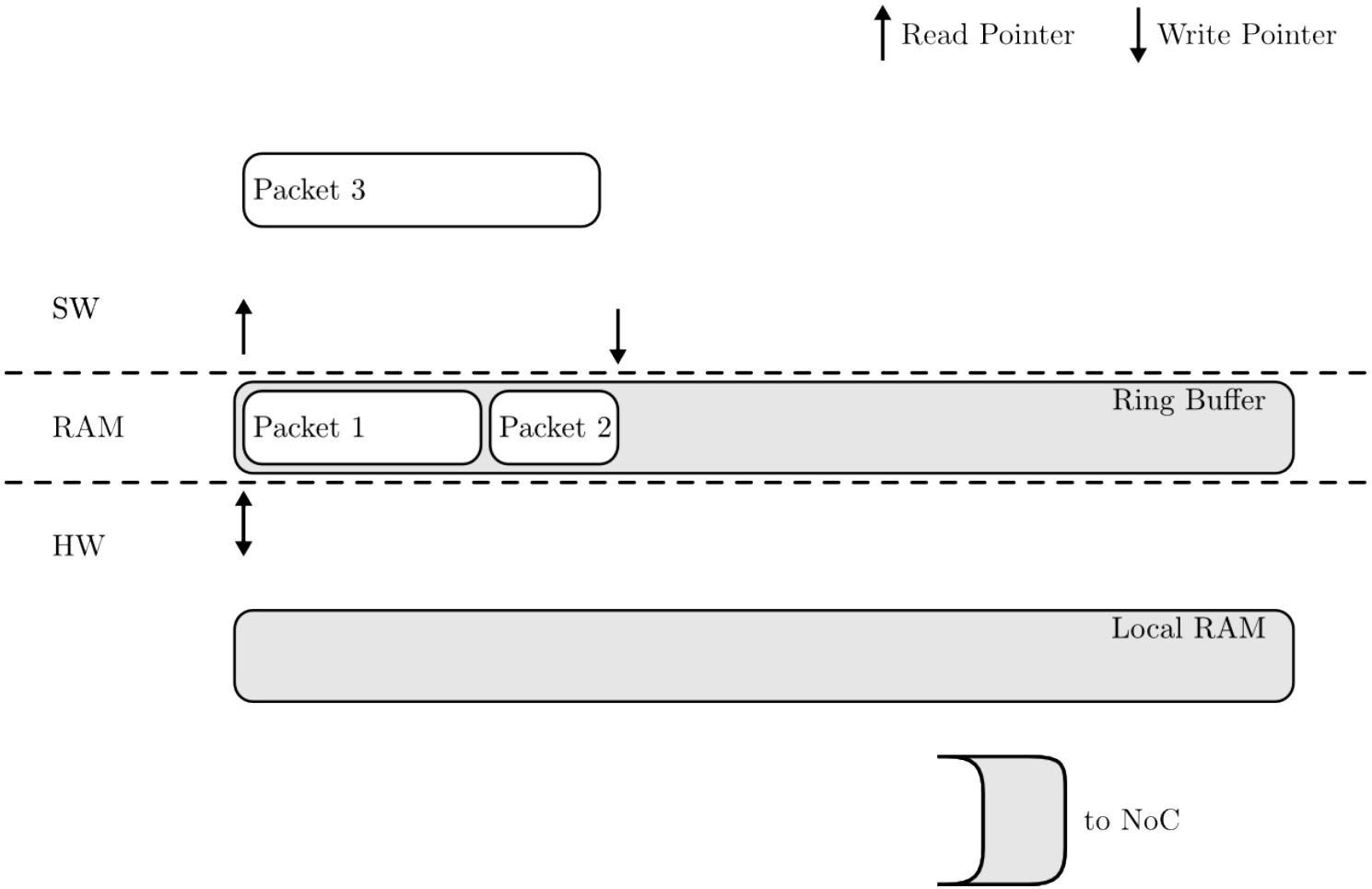
Software to Hardware Gateway



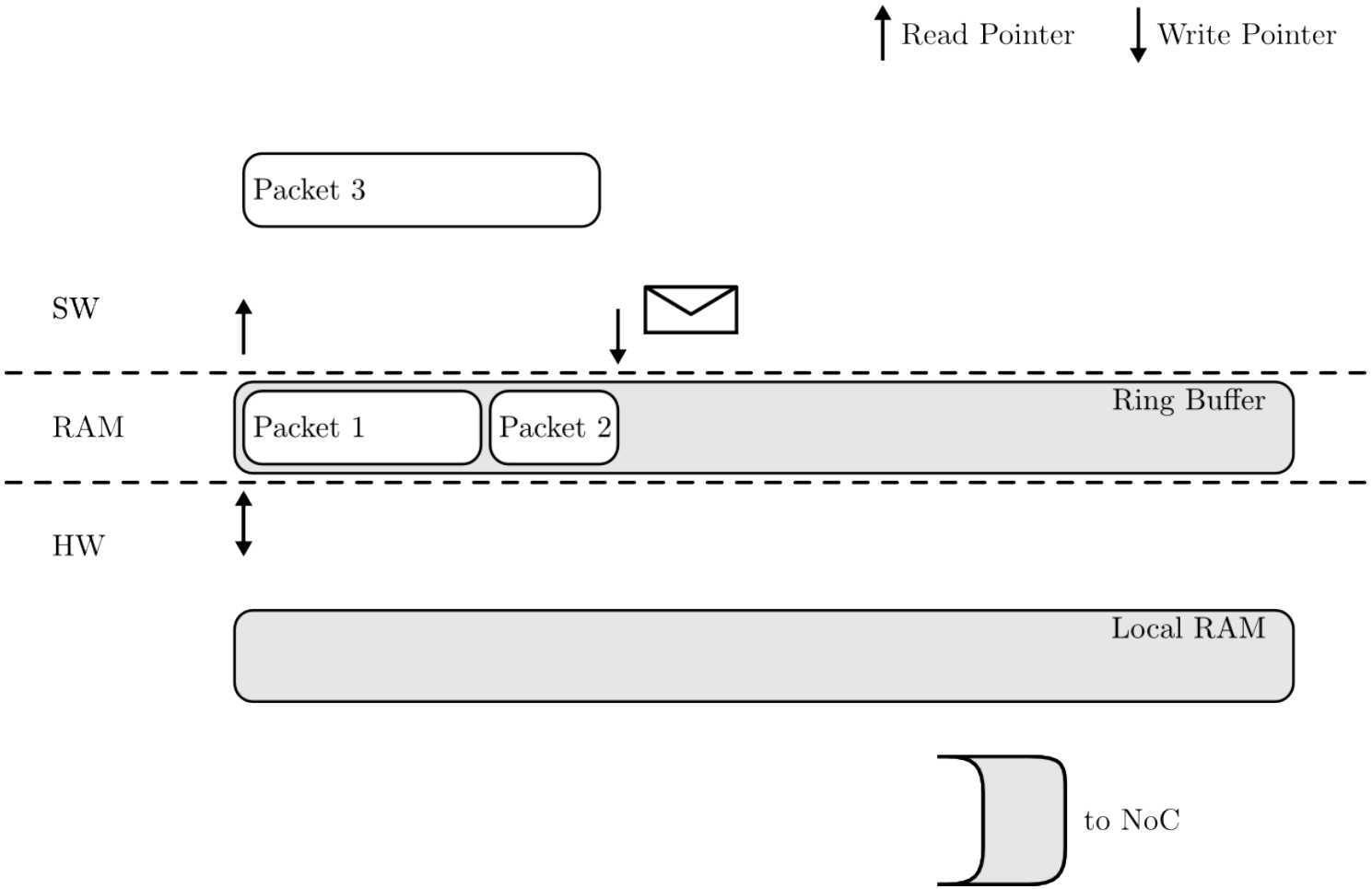
Software to Hardware Gateway



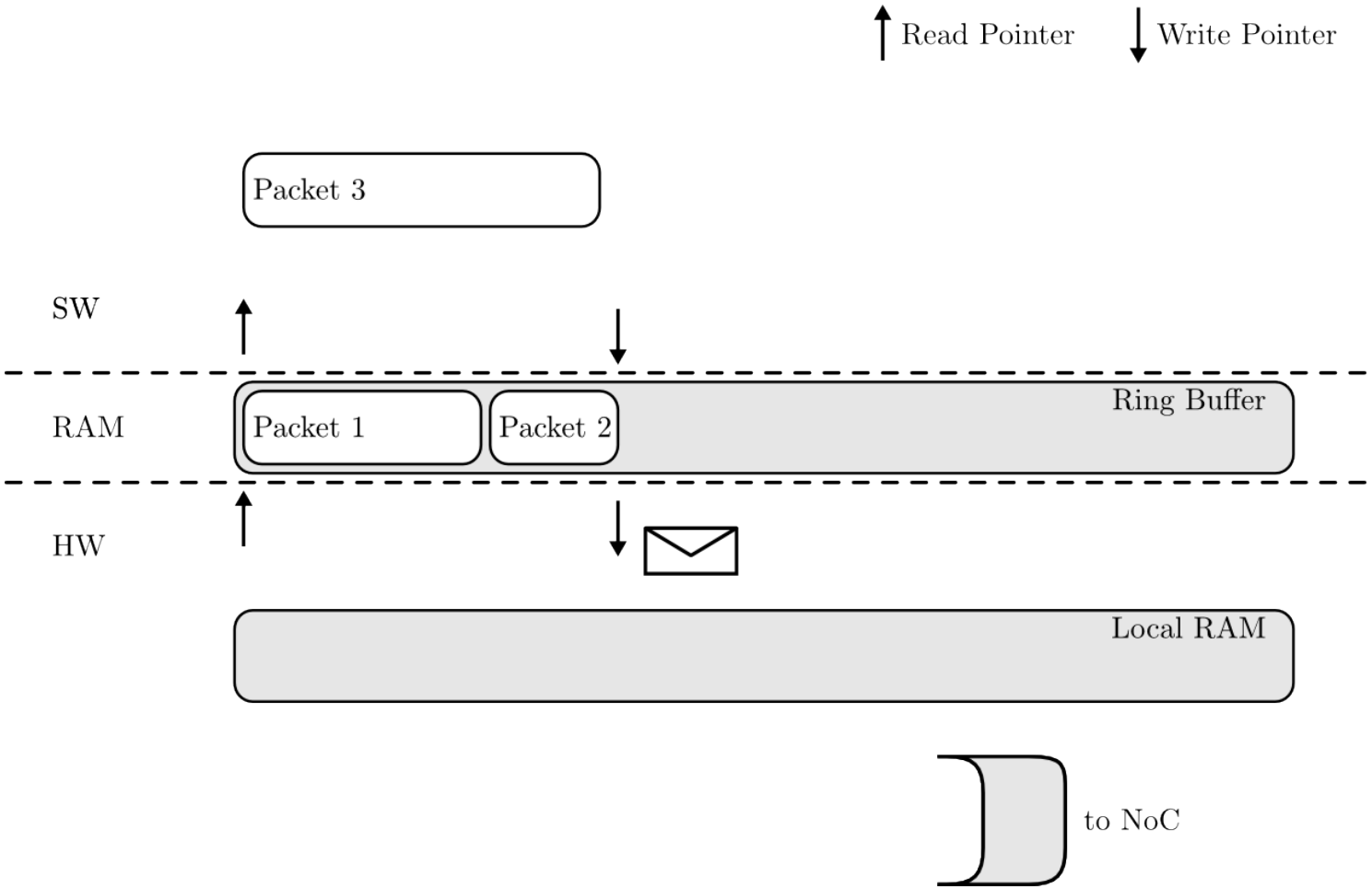
Software to Hardware Gateway



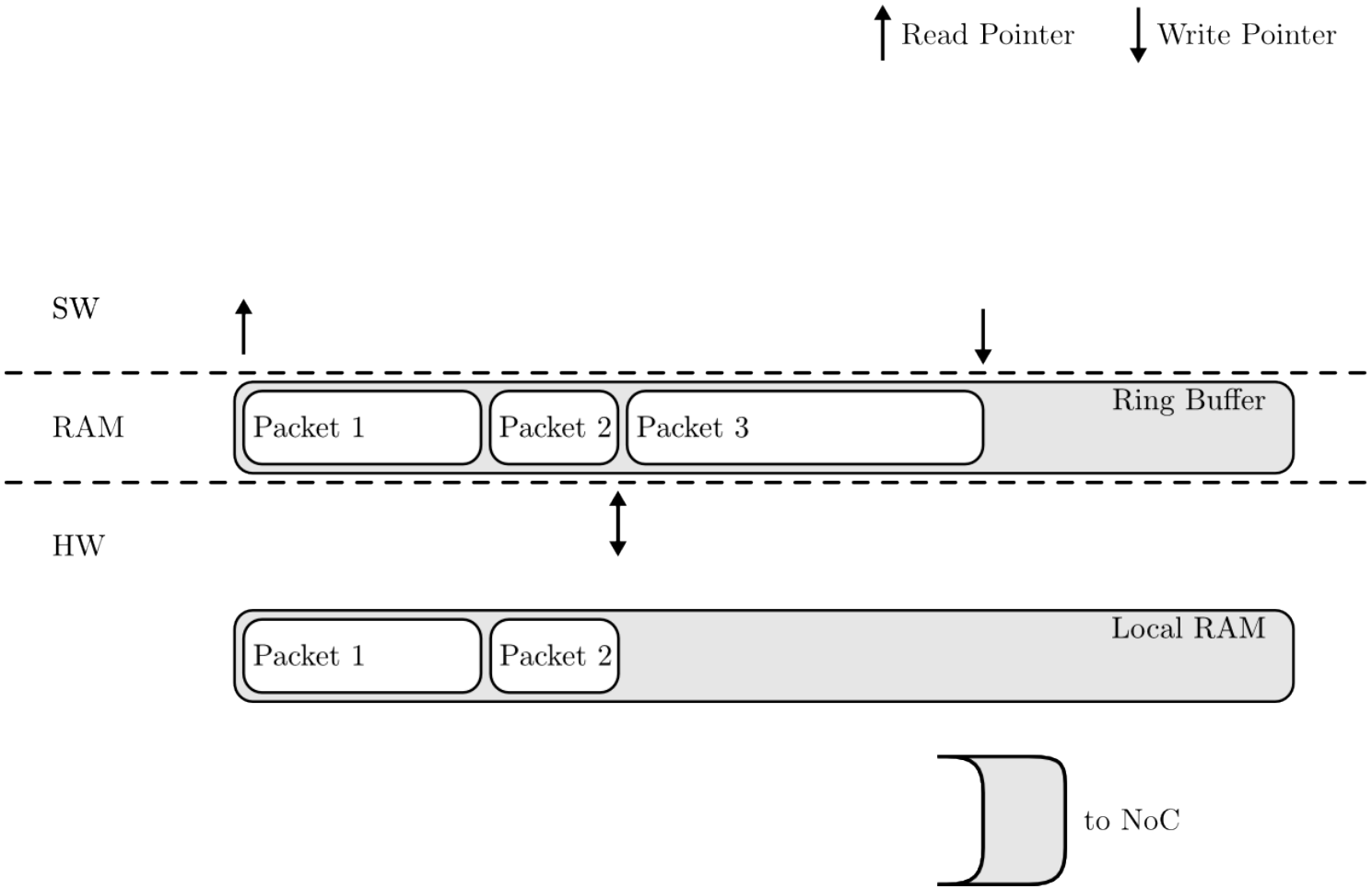
Software to Hardware Gateway



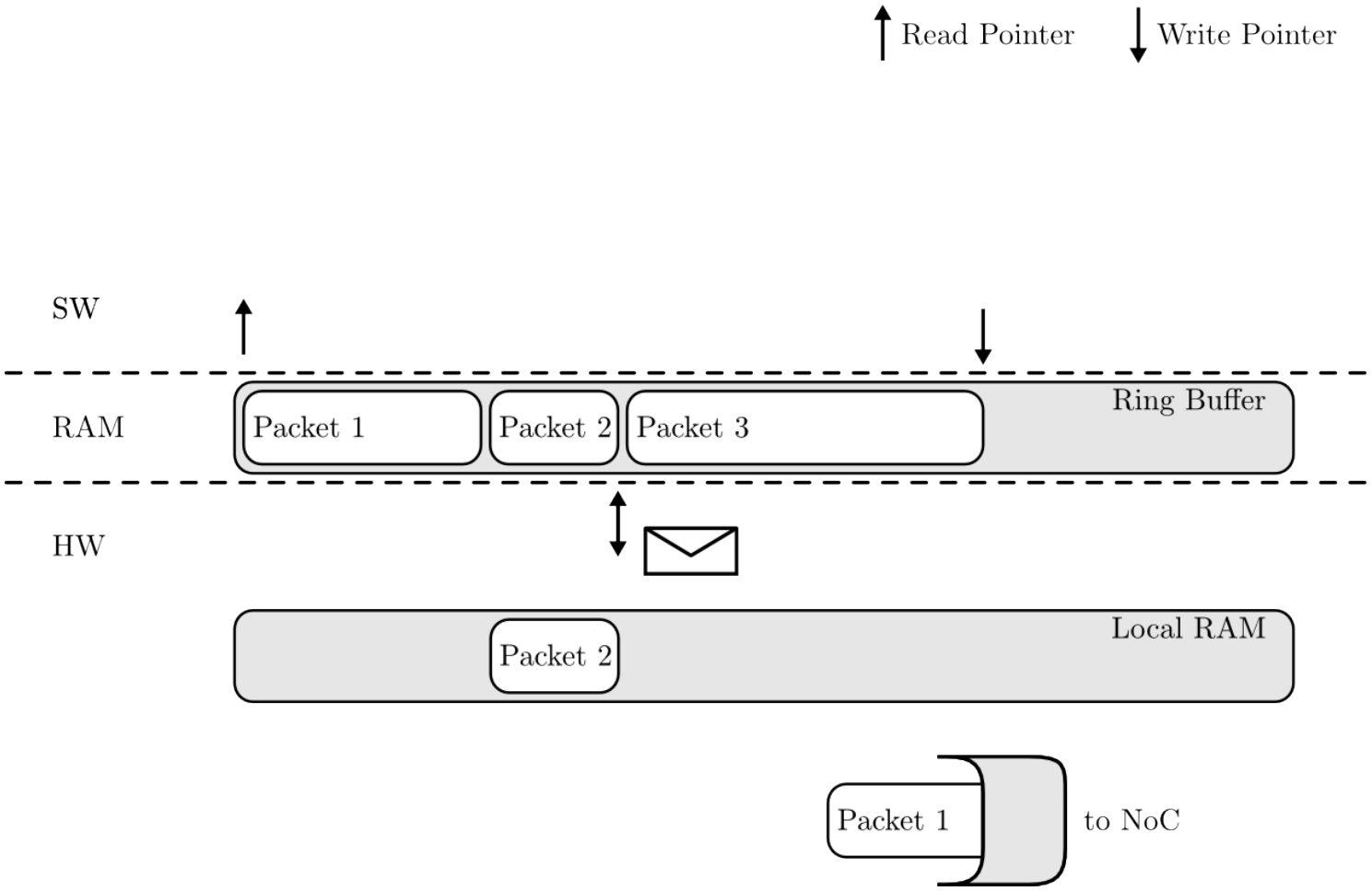
Software to Hardware Gateway



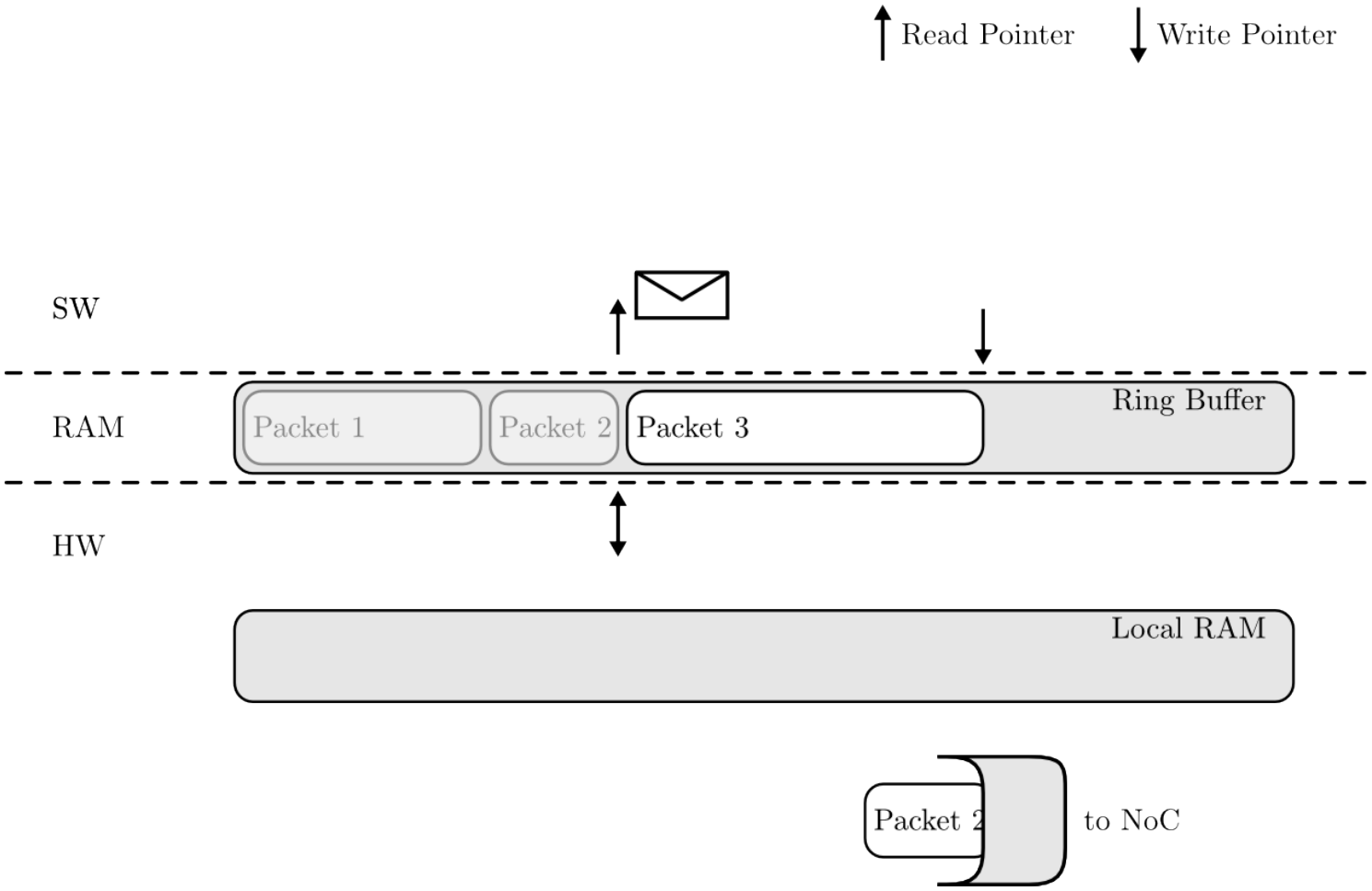
Software to Hardware Gateway



Software to Hardware Gateway



Software to Hardware Gateway



When to send write pointer to HW

- Sending pointer after every packet generates a high interrupt load
- Timely delivery of packets
- Non-blocking write access
- Delay-sensitive packets

When to send write pointer to HW

- Sending pointer after every packet generates a high interrupt load
  - Aggregate packets
- Timely delivery of packets
  - First packet starts a timer
  - Send write pointer when timer expires
  - Reset timer when sending write pointer for other reasons
- Non-blocking write access
  - Send write pointer when amount of free space below treshold
- Delay-sensitive packets
  - Packets have 'delay-sensitive' flag
  - When flag set, send write pointer immediately

Task List

- ✔ Familiarization
- ✔ Evaluation of possible architectures
- ✔ Communication between HW blocks
- ➔ Configuration interface
- ✔ Communication between SW and HW blocks
- ➔ Communication between HW blocks and SW
- ➔ Performance evaluation
- ➔ Final report

Challenges

- Simulation
  - Very complex BUS interface of HW-threads
  - Simulating overall system is a Hercules task
- Data representation in SW
  - MSB-LSB
  - Endianness
- Multithreaded SW
  - Take care of race conditions

Summary

- Network on Chip implemented for data transfer between HW threads
- Using ring buffers to transfer data from SW to HW and vice versa
- Dedicated HW and SW threads implemented for this purpose