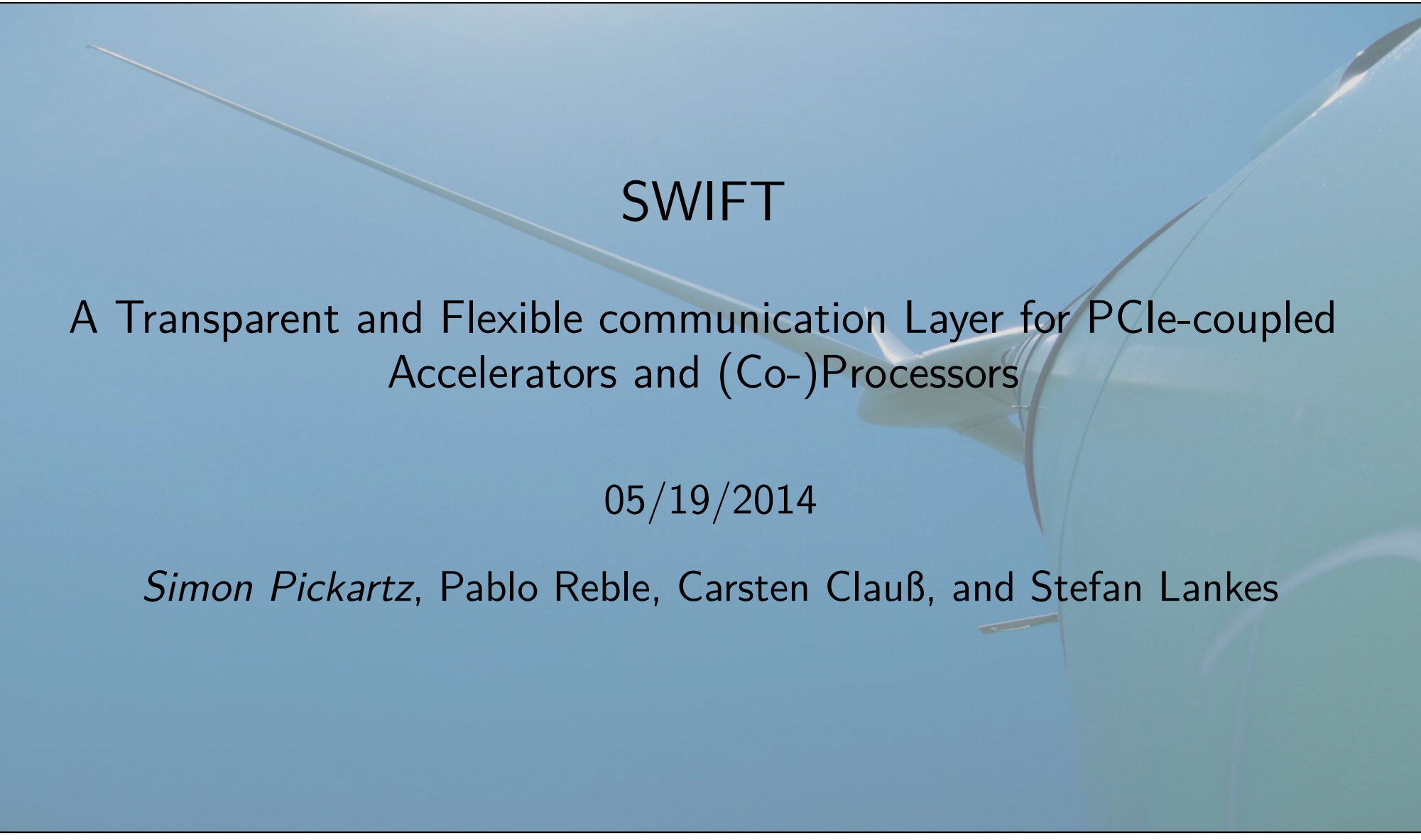


E.ON Energy Research Center

SWIFT

A Transparent and Flexible communication Layer for PCIe-coupled Accelerators and (Co-)Processors

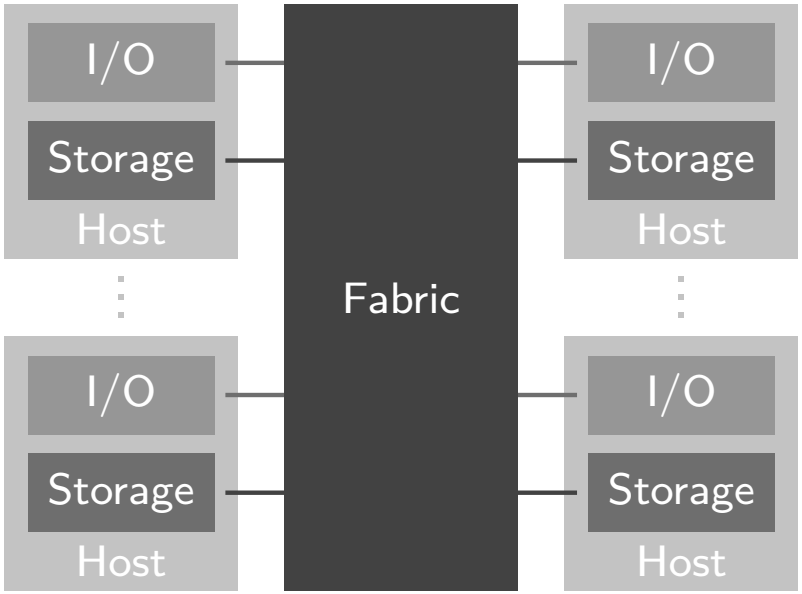
05/19/2014

Simon Pickartz, Pablo Reble, Carsten Clauß, and Stefan Lankes

RWTH AACHEN UNIVERSITY

Today's HPC Systems

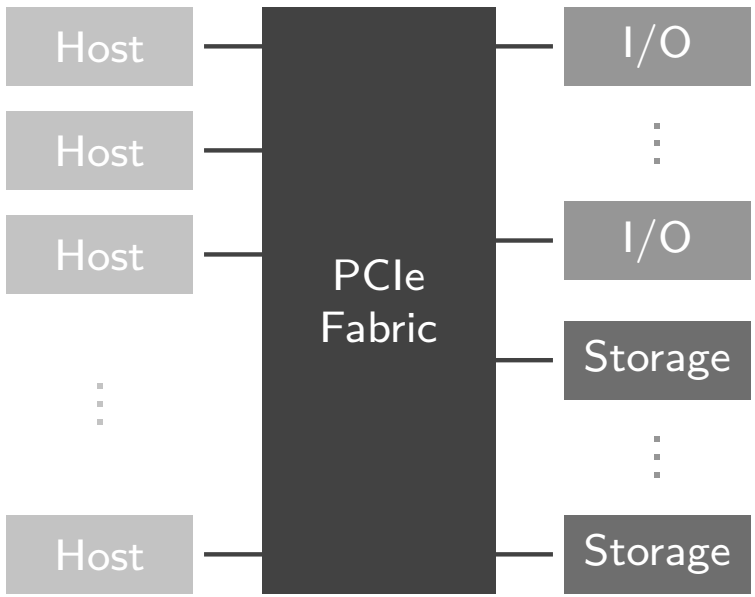
- Homogeneous hardware landscape
- Separate computer systems connected to increase the aggregated compute power
- Different interconnects for LAN and SAN
- RDMA capabilities



→ For portability concerns one standardised interface to layers on top is sufficient (e. g. uDAPL)

Tomorrow's HPC Systems

- Intra-rack network connecting
  - Hosts
  - I/O devices
  - Storage
- Heterogeneous compute nodes
  - CPUs
  - GPUs
  - Accelerators
  - Etc.
- Peer-to-peer communication
- RDMA *and* RMA capabilities
- Still different interconnects



→ Computer systems connected to *share* resources and to increase the aggregated compute power

Agenda

- Socket Wheeled Intelligent Fabric Transport (SWIFT)
- Hardware
- Results
- Outlook

SWIFT – Requirements

- Support for heterogeneous network landscapes
- High portability
- Supply of different programming models
- Consideration of the hardware's RDMA and RMA capabilities
- High performance

SWIFT – Requirements

- Support for heterogeneous network landscapes
  → A transparent solution is targeted
- High portability
- Supply of different programming models
- Consideration of the hardware's RDMA and RMA capabilities
- High performance

SWIFT – Requirements

- Support for heterogeneous network landscapes
  - → A transparent solution is targeted
- High portability
  - → Hardware abstraction
- Supply of different programming models
- Consideration of the hardware's RDMA and RMA capabilities
- High performance

SWIFT – Requirements

- Support for heterogeneous network landscapes
  → A transparent solution is targeted
- High portability
  → Hardware abstraction
- Supply of different programming models
  → Service-oriented, SPMD, etc.
- Consideration of the hardware's RDMA and RMA capabilities

- High performance

SWIFT – Requirements

- Support for heterogeneous network landscapes
  - → A transparent solution is targeted
- High portability
  - → Hardware abstraction
- Supply of different programming models
  - → Service-oriented, SPMD, etc.
- Consideration of the hardware's RDMA and RMA capabilities
  - → Offer one-sided communication primitives
- High performance

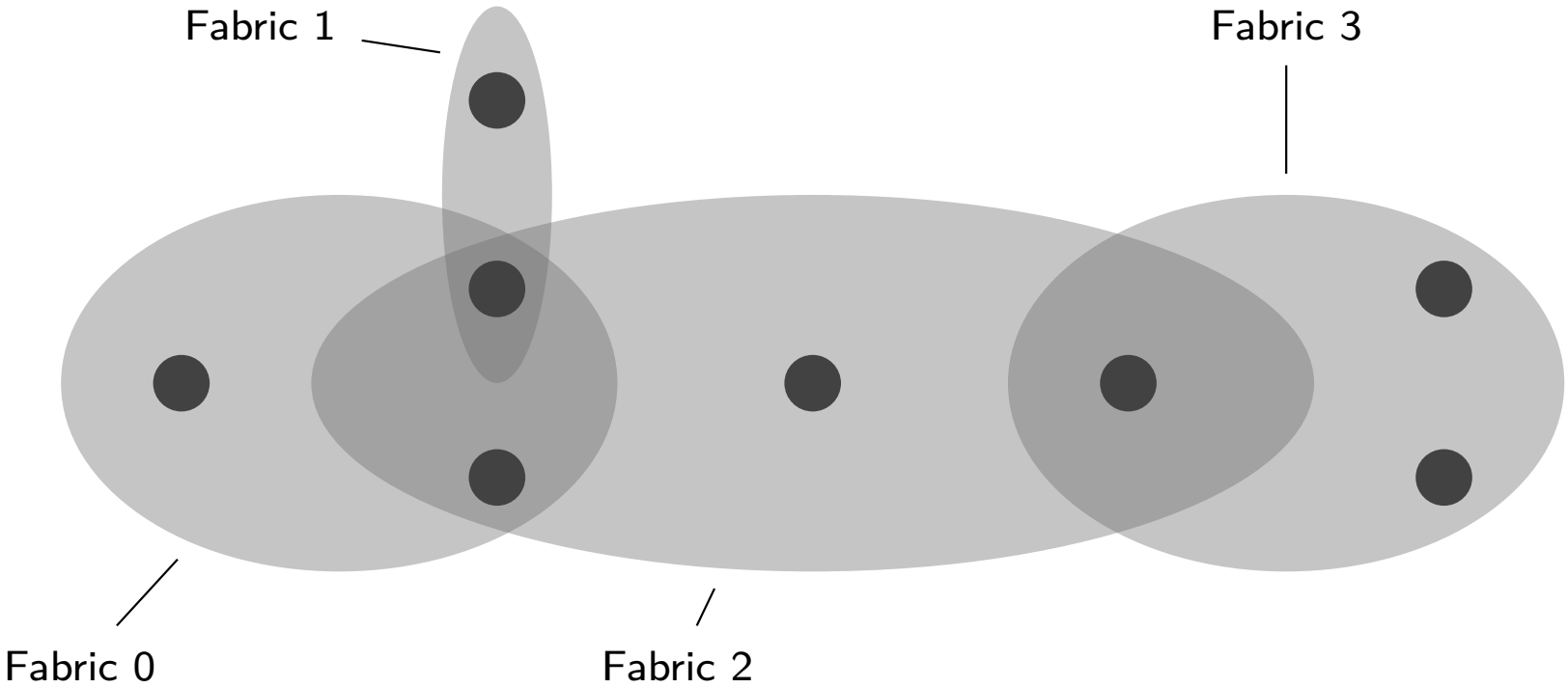
SWIFT – Requirements

- Support for heterogeneous network landscapes
  - → A transparent solution is targeted
- High portability
  - → Hardware abstraction
- Supply of different programming models
  - → Service-oriented, SPMD, etc.
- Consideration of the hardware's RDMA and RMA capabilities
  - → Offer one-sided communication primitives
- High performance
  - → Low latencies and high data rates

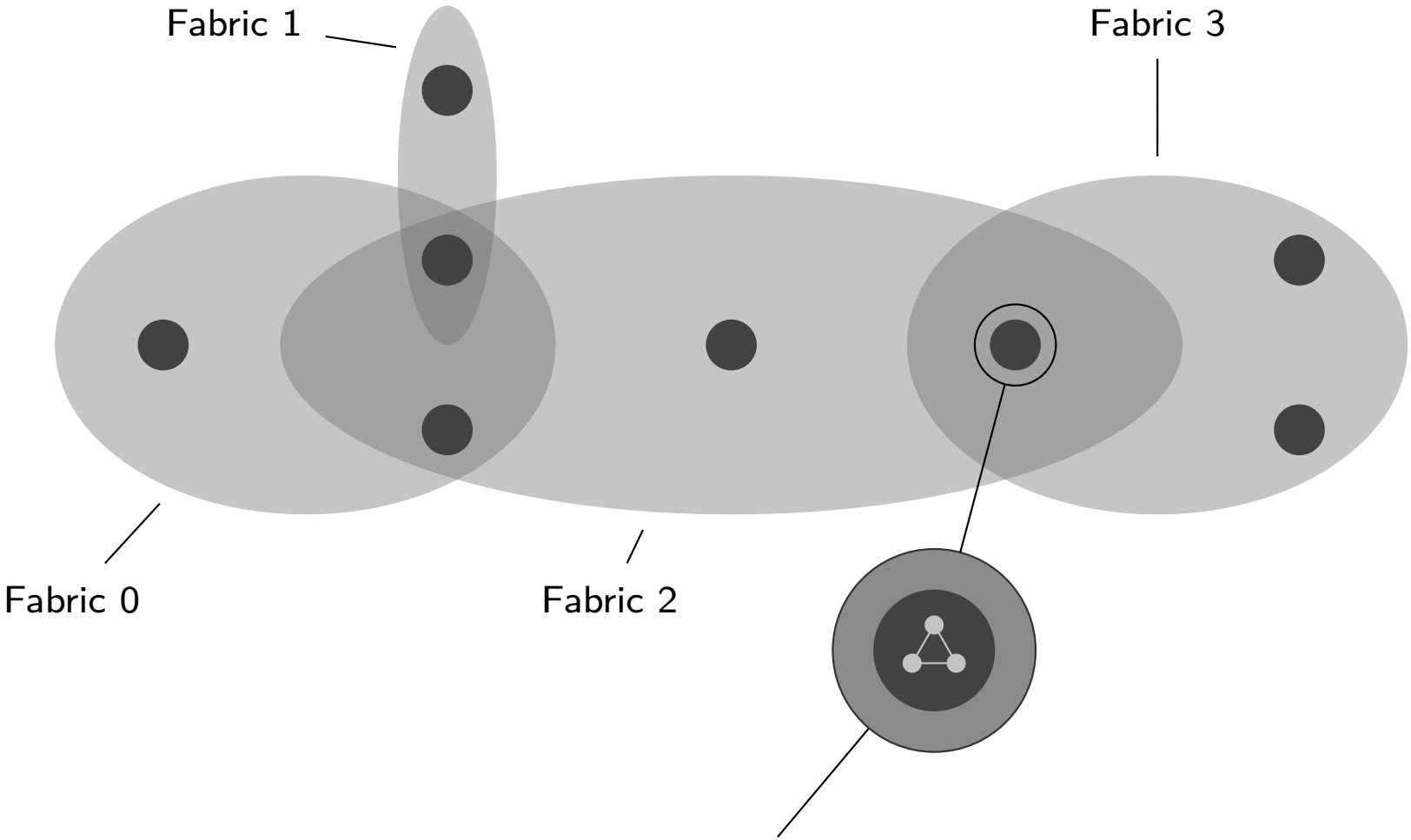
SWIFT – Basic Concepts

- Topology
  - Hosts
  - Nodes
  - Endpoints
- Communication modes
  - Asynchronous signaling via mails
  - Non-blocking two-sided communication
  - One-sided communication (including atomics)
- Gateway-based fabric connection

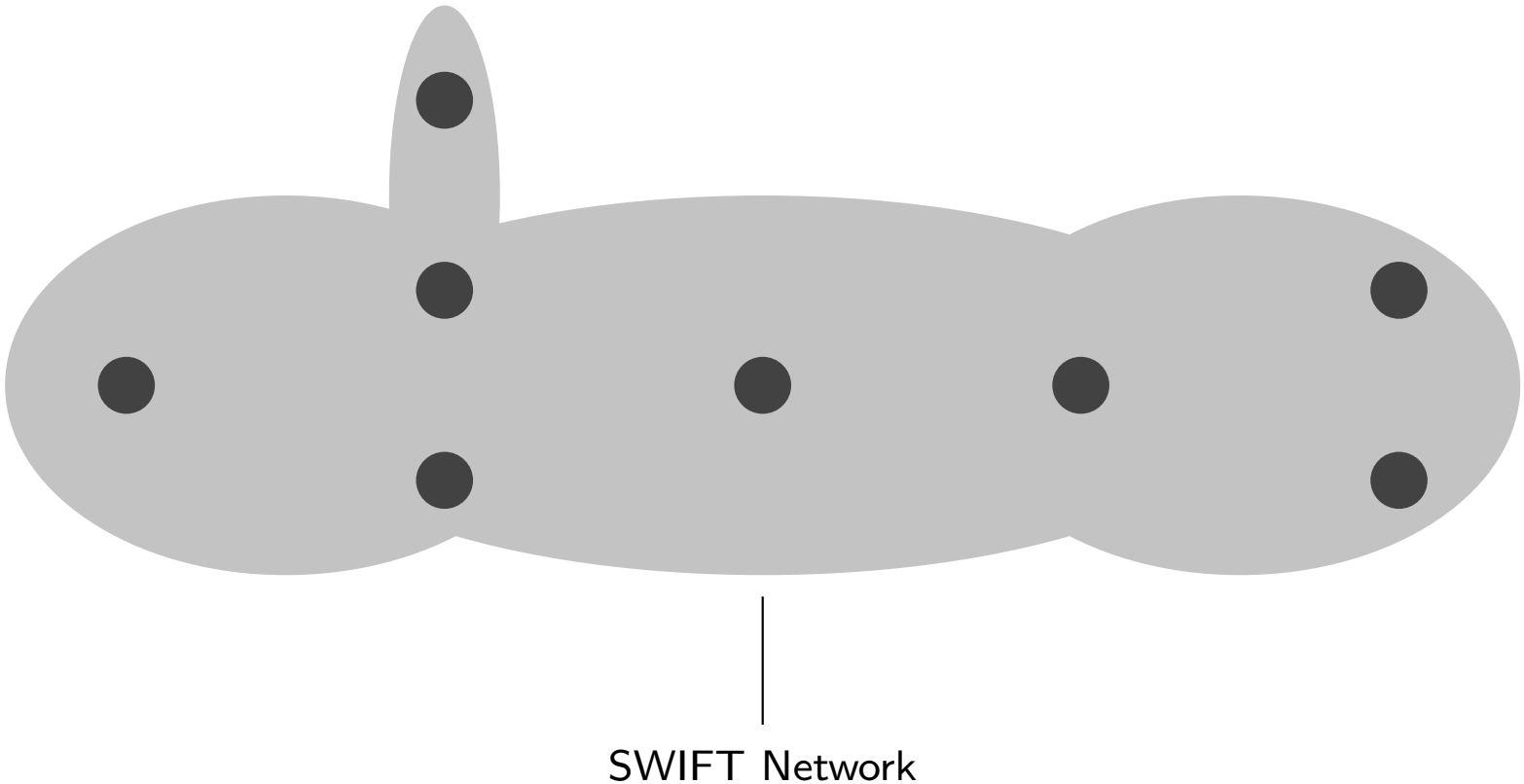
Topology



Topology

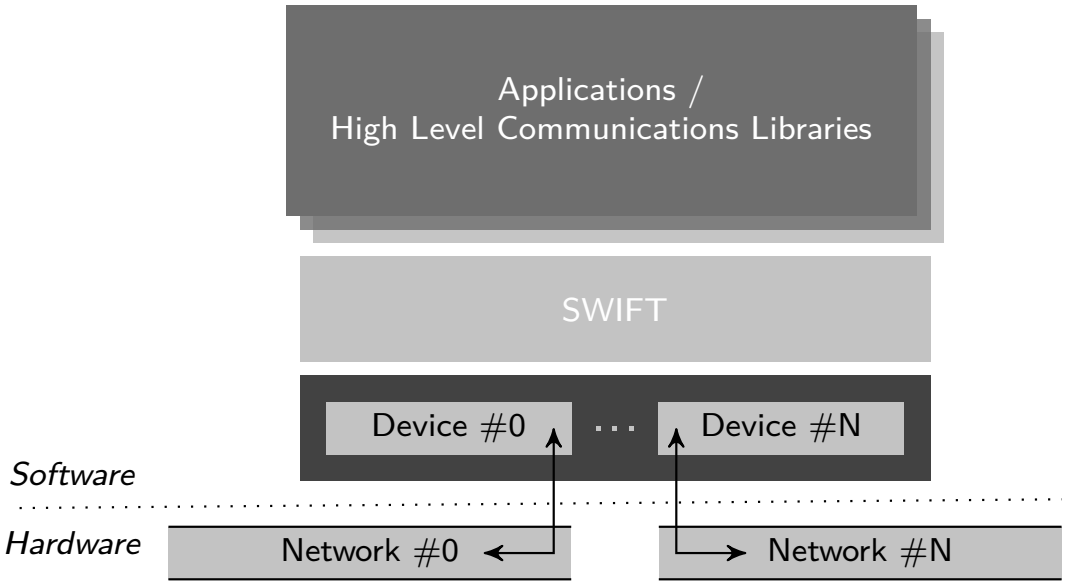


Topology



Layered Architecture

- Application layer
  - Higher level libraries (e. g. MPI)
  - Parallel applications
  - Service-oriented apps
- SWIFT layer
  - Routing
  - Topology
  - Messaging services
- Device layer
  - Hardware abstraction
  - Optimization



→ Well-defined interfaces to layers above and below

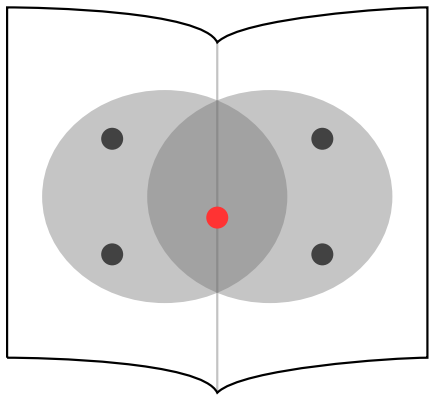
SWIFT Device

- Small interface (around 20 prototypes only)
- Administration module
  - Constructor and destructor
  - Automatic discovery of fabric nodes
- Channel module
  - Bi-directional FIFO channel
  - Fixed channel size
  - Asynchronous connection establishment via `create()` and `connect()`
  - Three transfer modes: PIO, DMA, and AUTO

→ High portability

Connection Setup

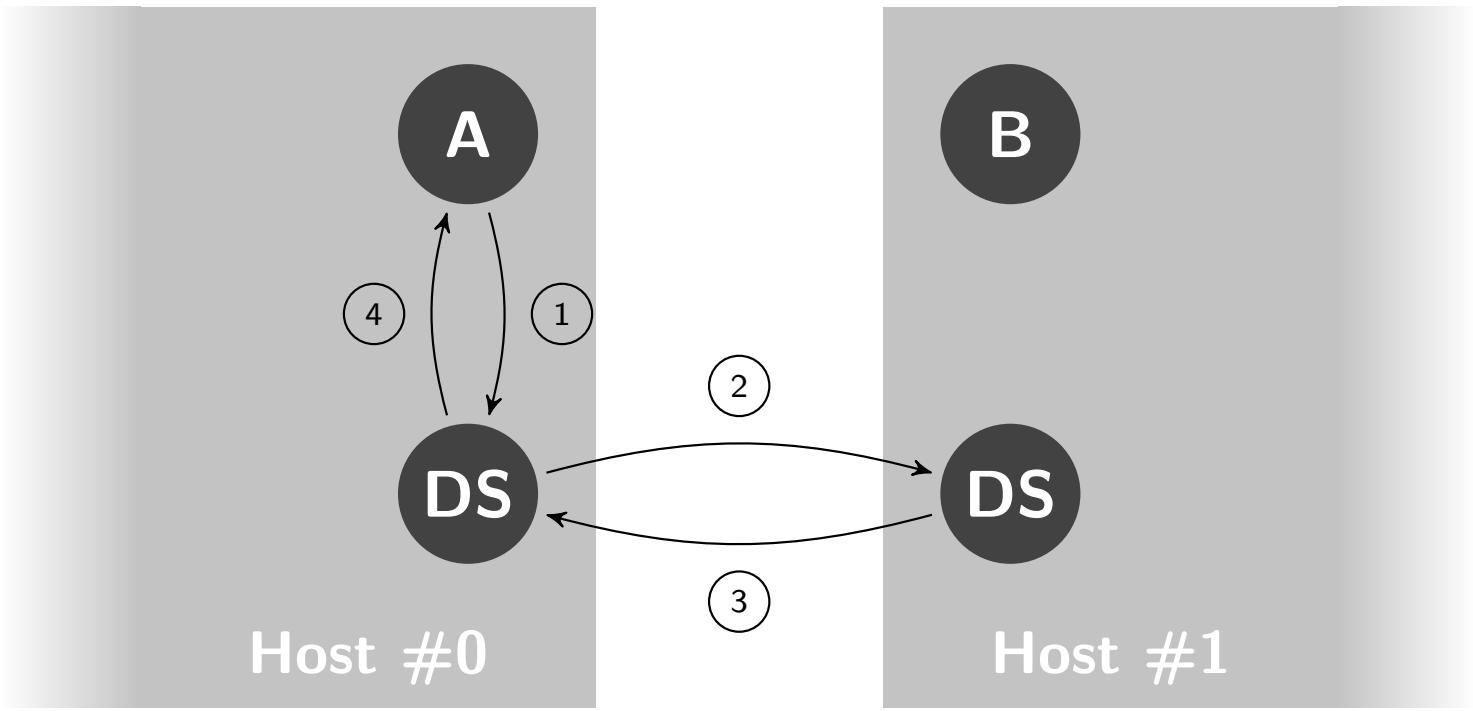
- A distributed *Directory Service* (DS)
  - Dedicated process for holding topology information
  - Automatically maintains connections to other DS on neighbor hosts
  - Manages node IDs for the local nodes
  - No single point of failure



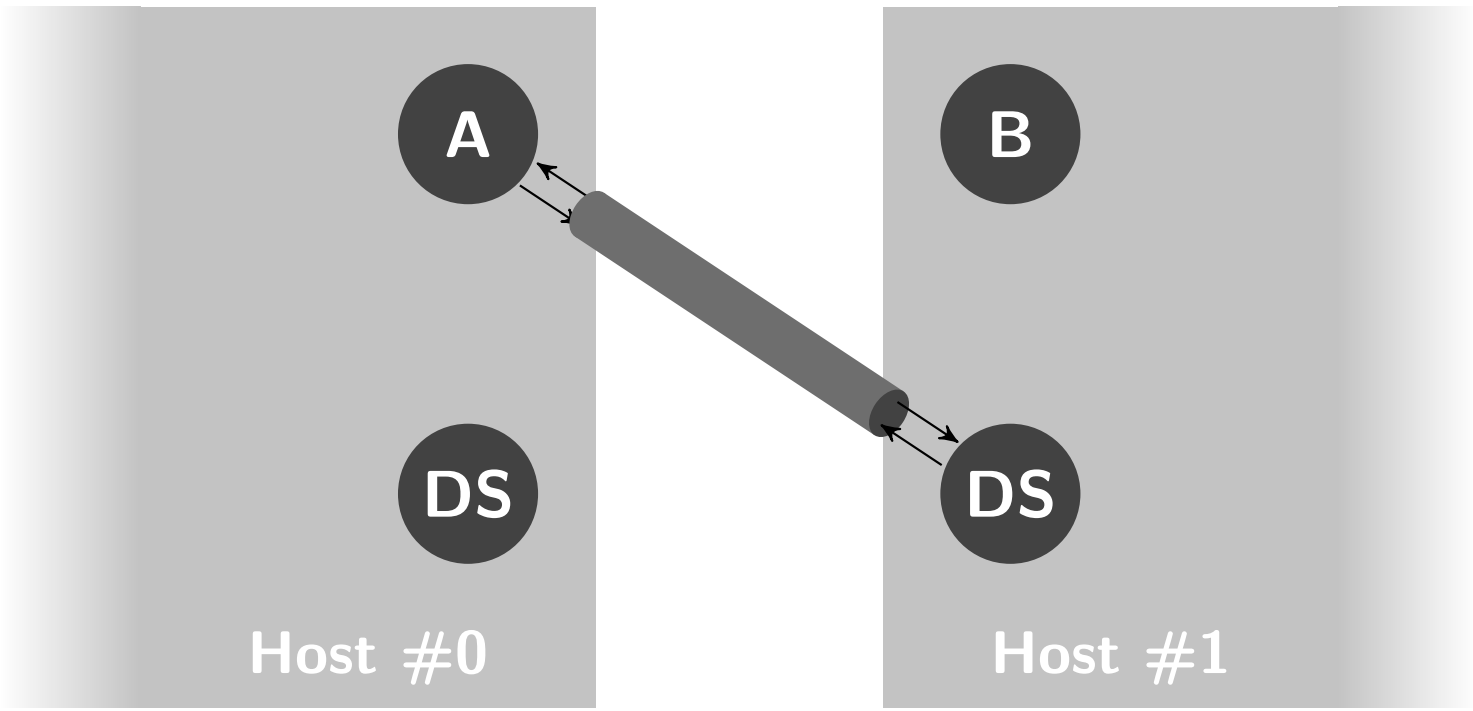
- On-demand connection setup via DS
  - Direct communication between nodes on different hosts

    → Minimization of the hop count
  - Automatically connect to destination DS if necessary
  - A bit of proactivity

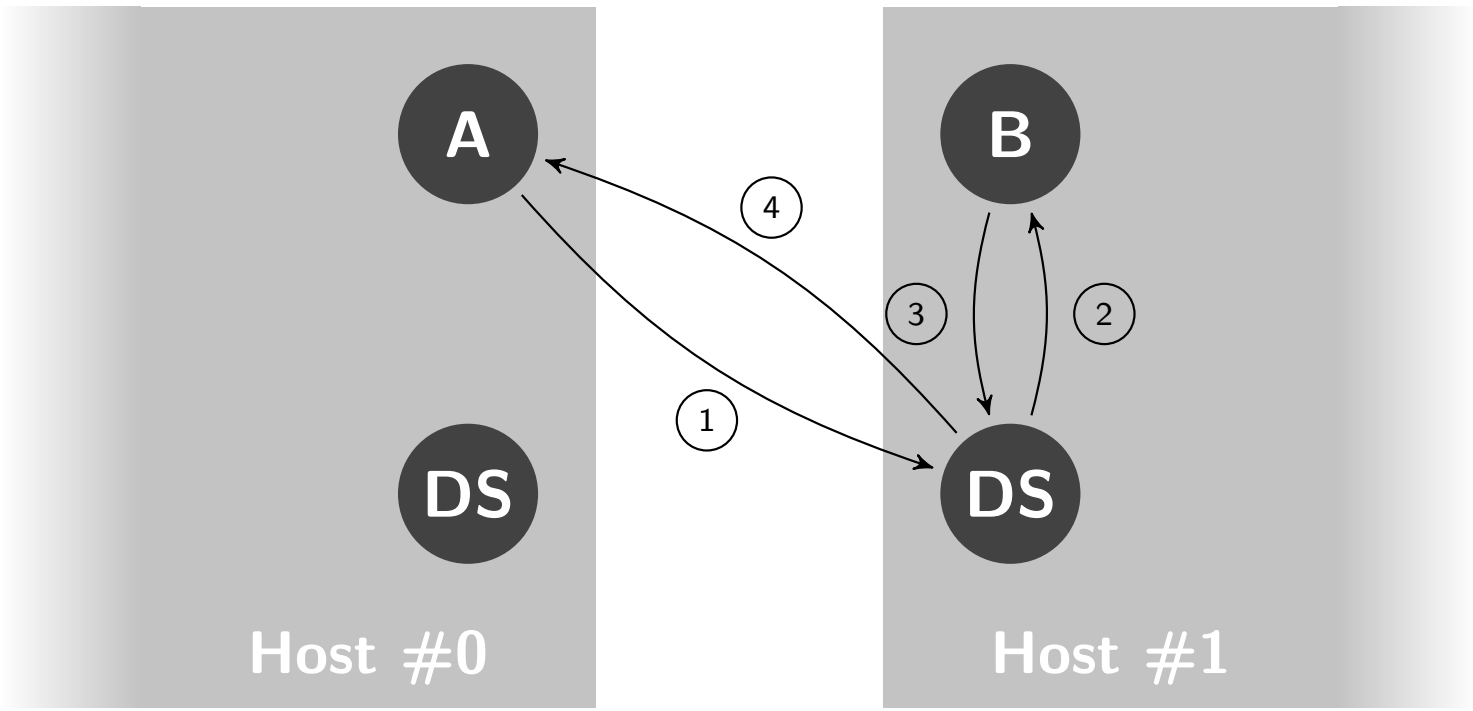
Connection Setup



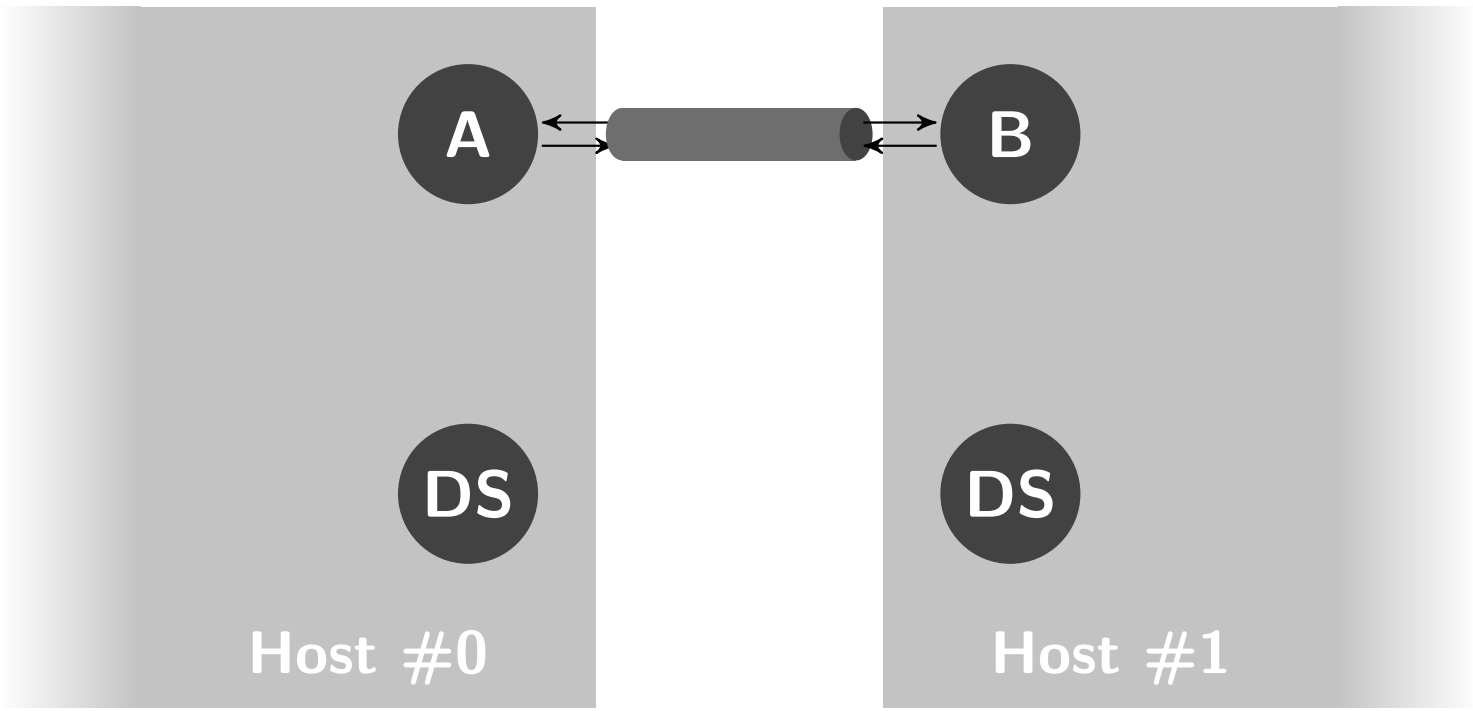
Connection Setup



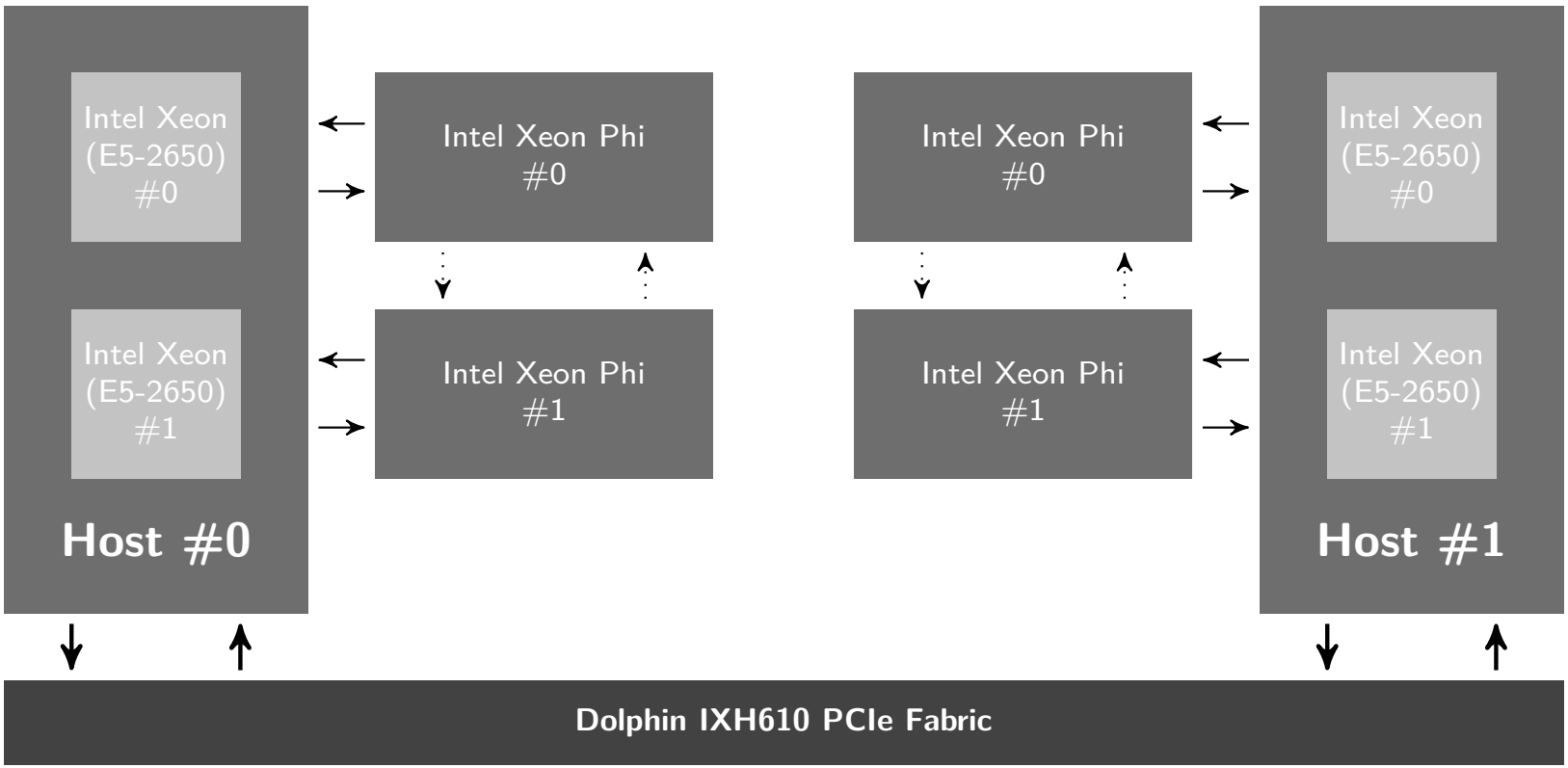
Connection Setup



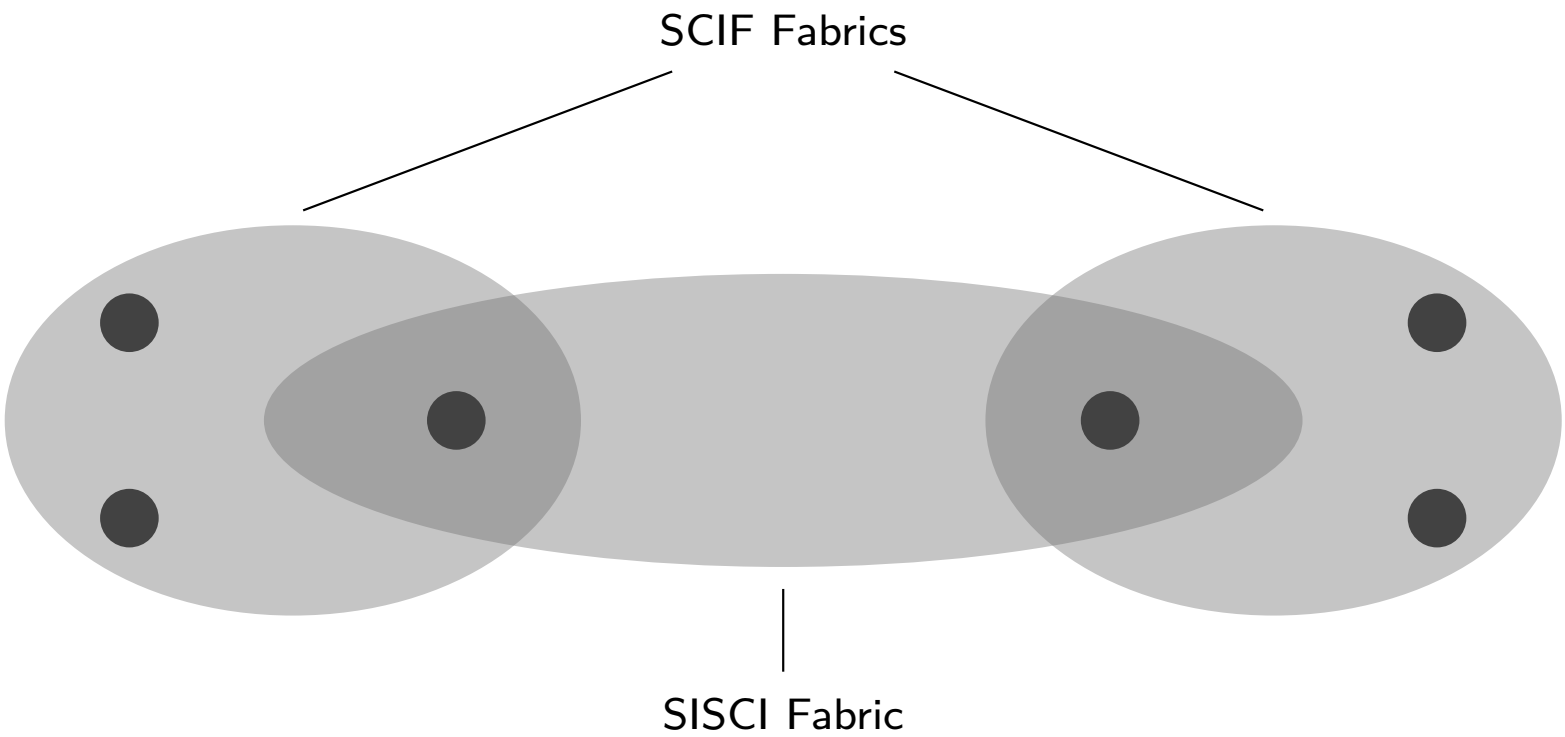
Connection Setup



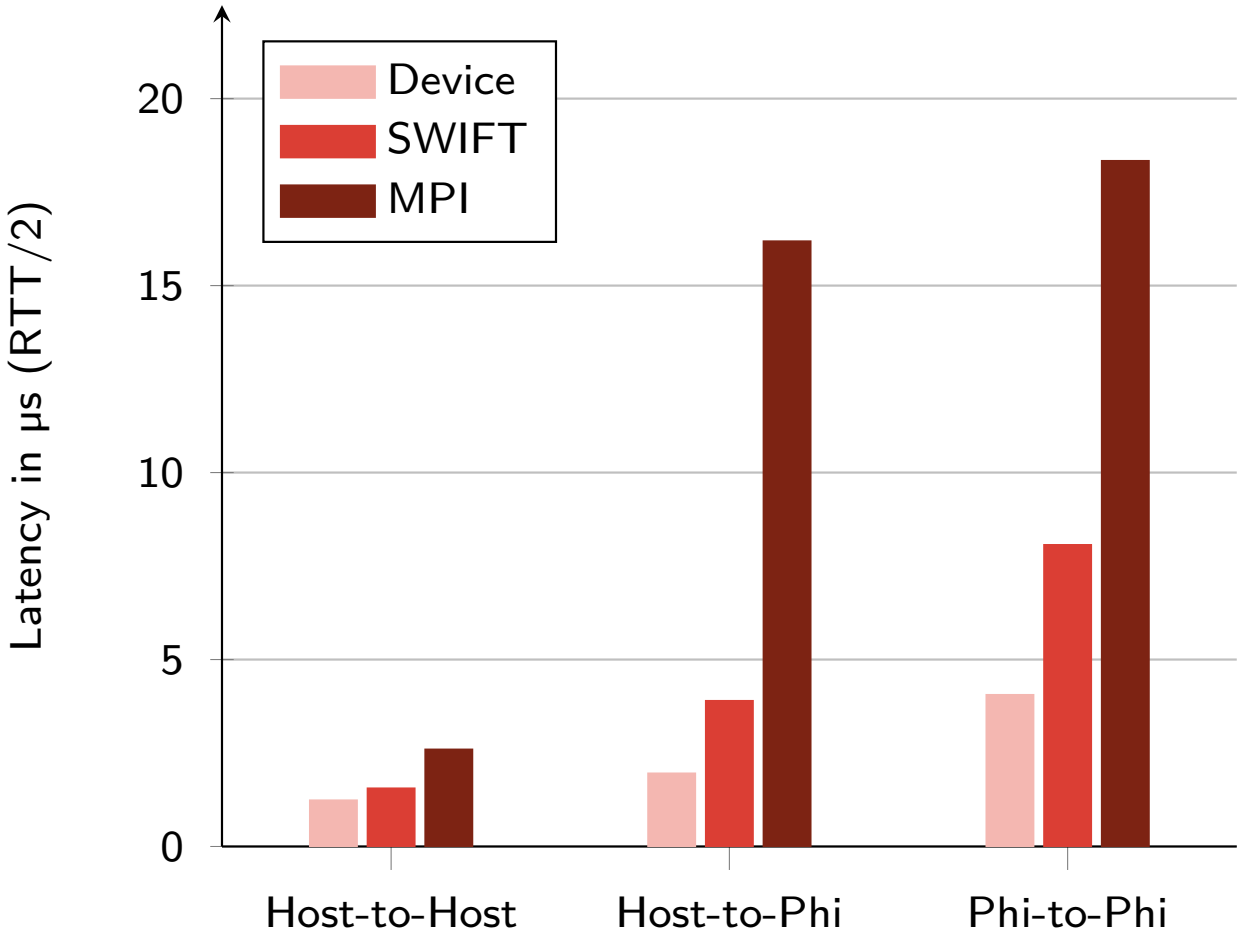
The ACS Cluster



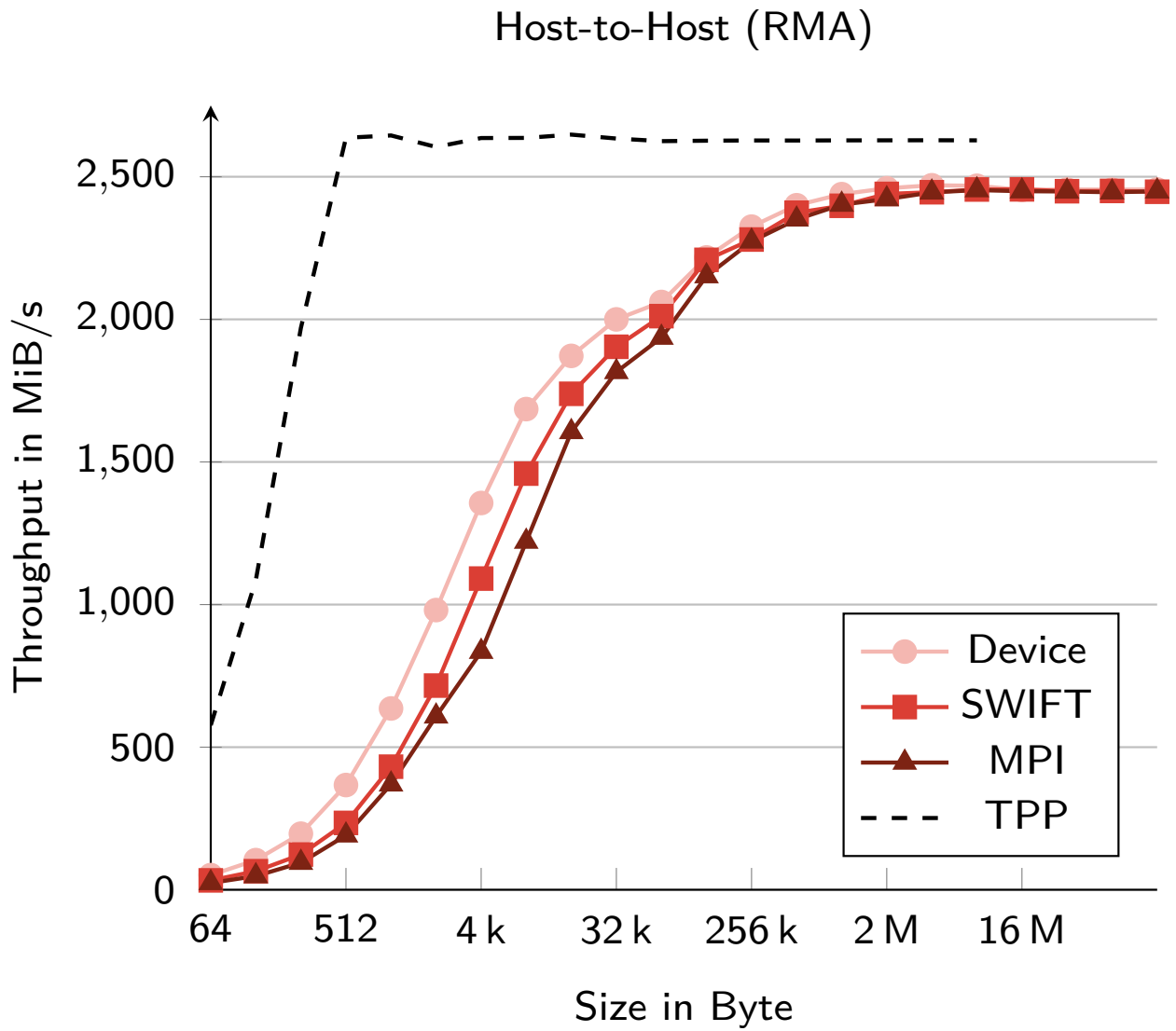
Mapping SWIFT onto the Cluster



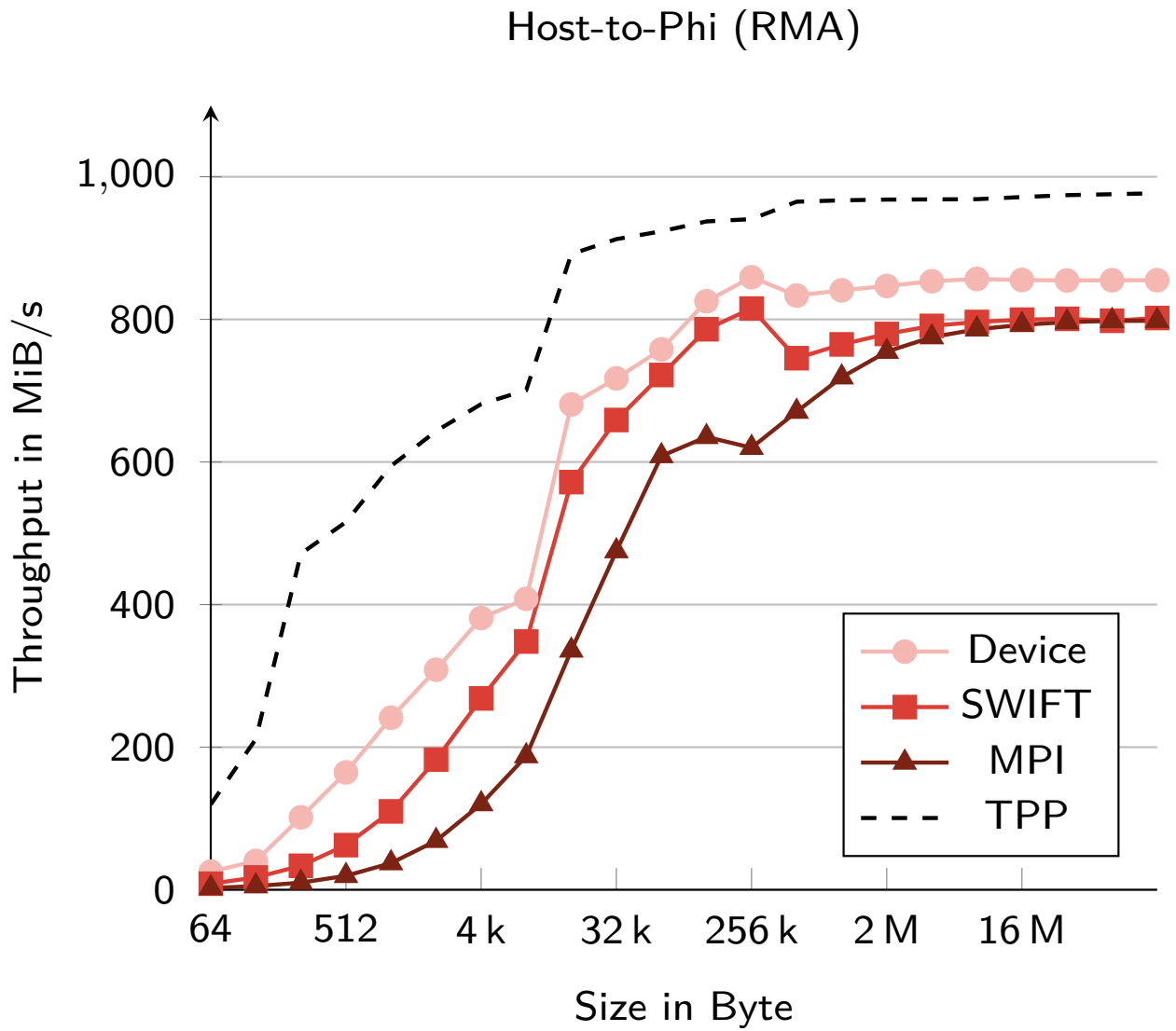
SWIFT Overhead – Latencies



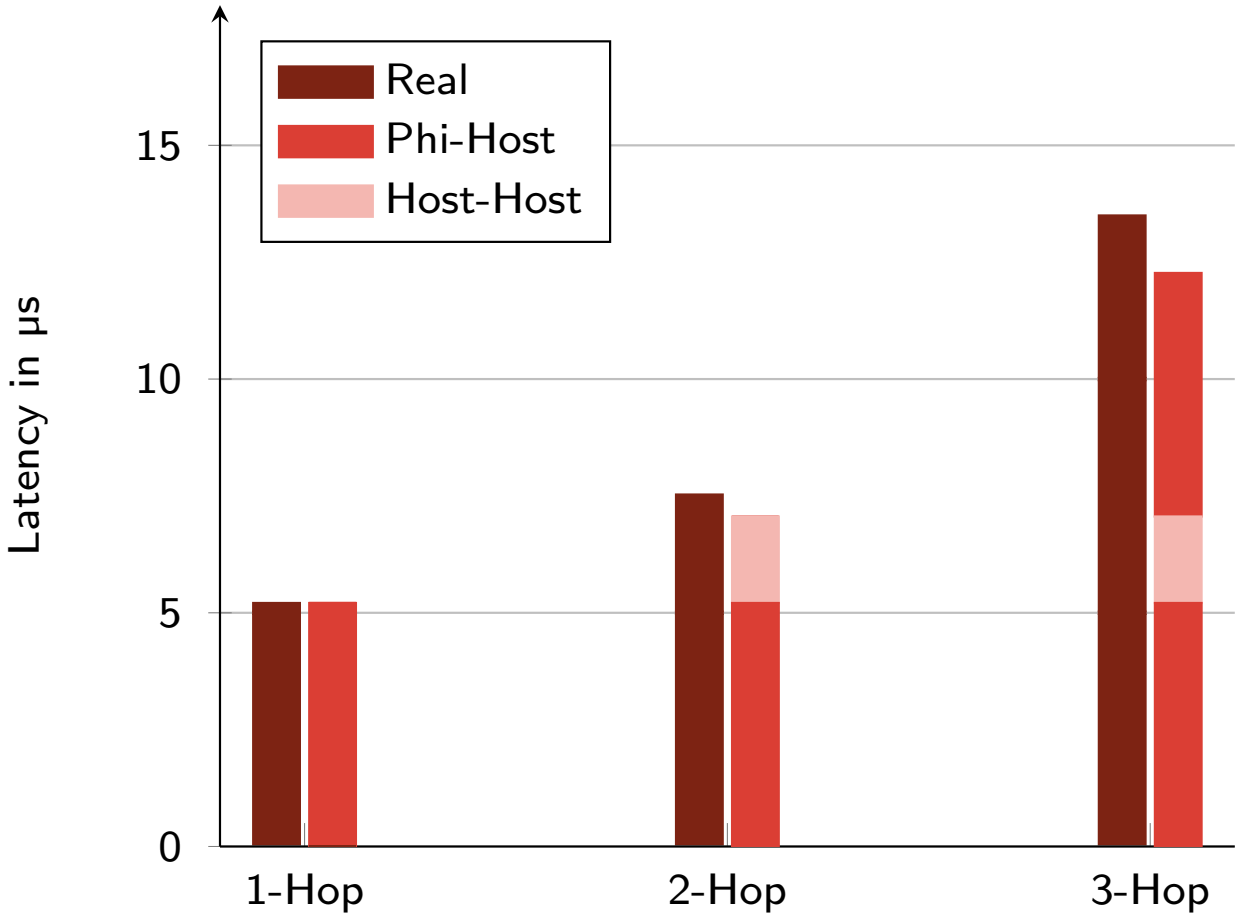
SWIFT Overhead – Throughput



SWIFT Overhead – Throughput

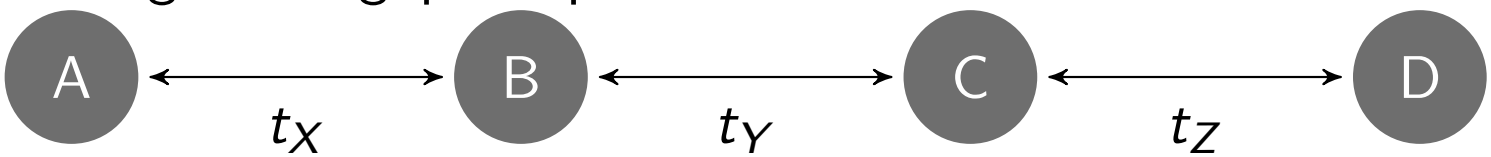


Multi-Hop PingPong – Latency

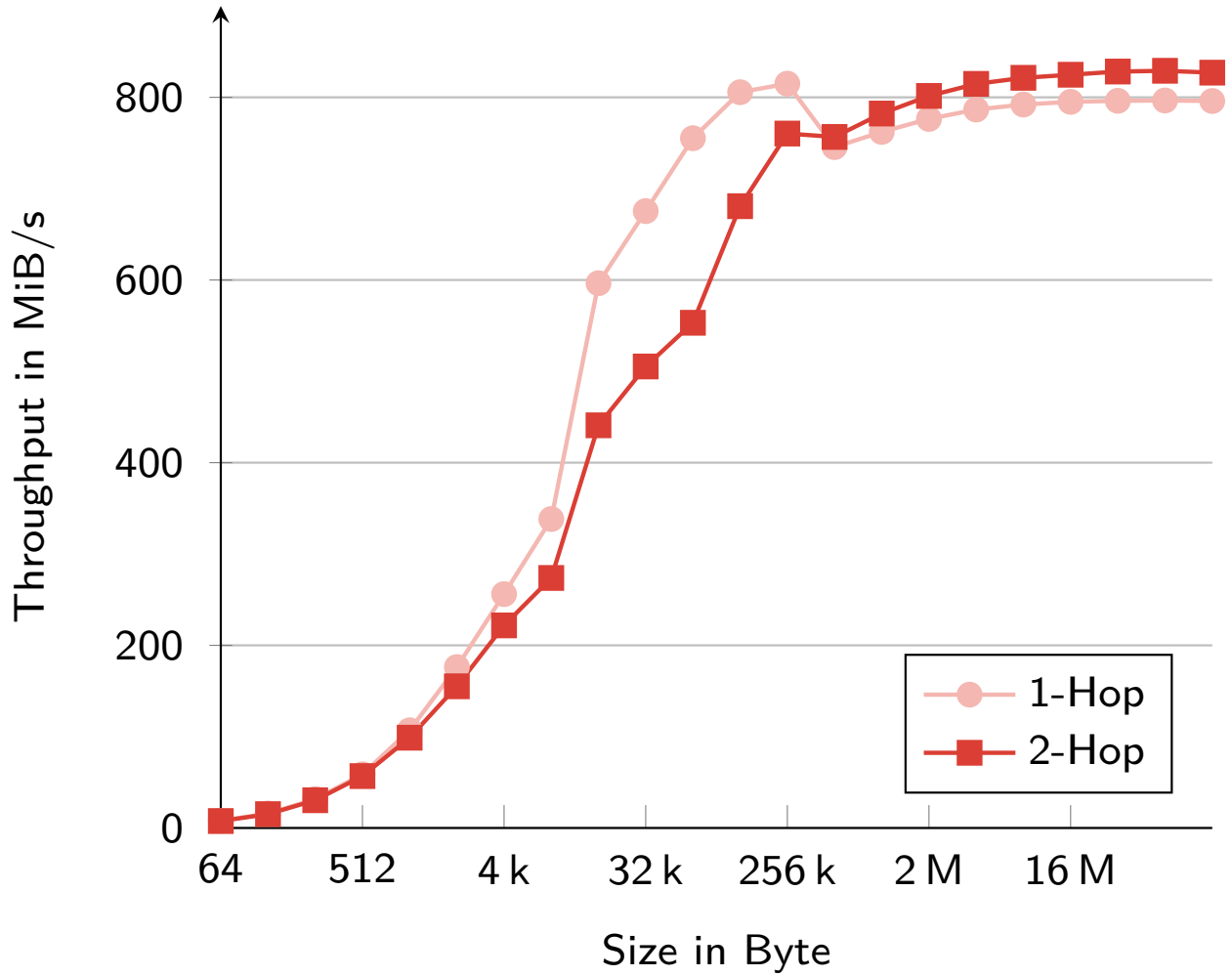


Multi-Hop PingPong – Throughput

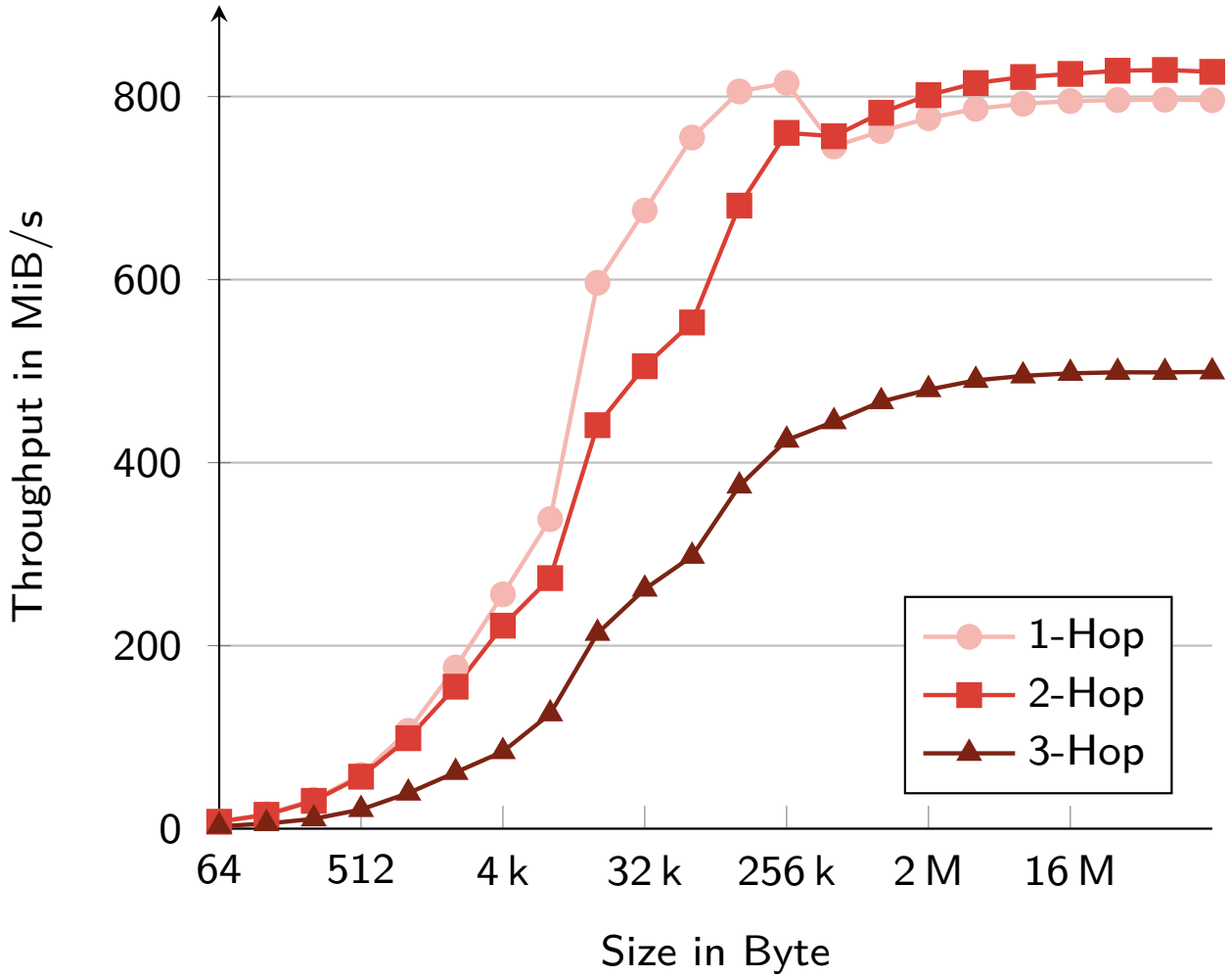
- Latencies accumulate
- Average throughput equals that of the bottleneck link



Multi-Hop PingPong – Throughput

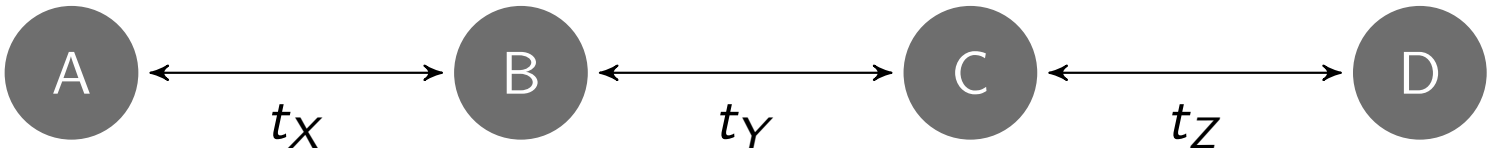


Multi-Hop PingPong – Throughput



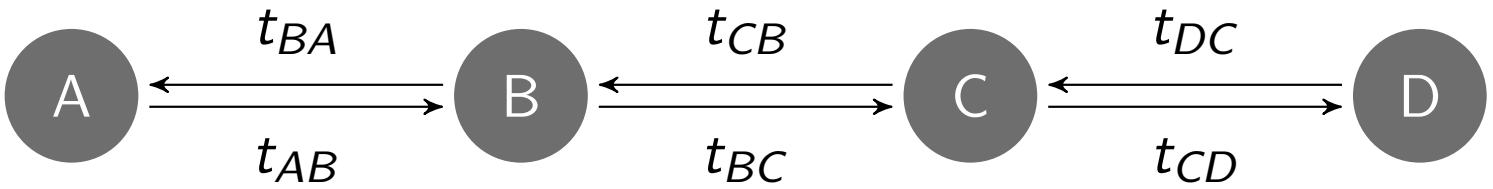
Multi-Hop PingPong – Throughput

- Latencies accumulate
- Average throughput equals that of the bottleneck link



Multi-Hop PingPong – Throughput

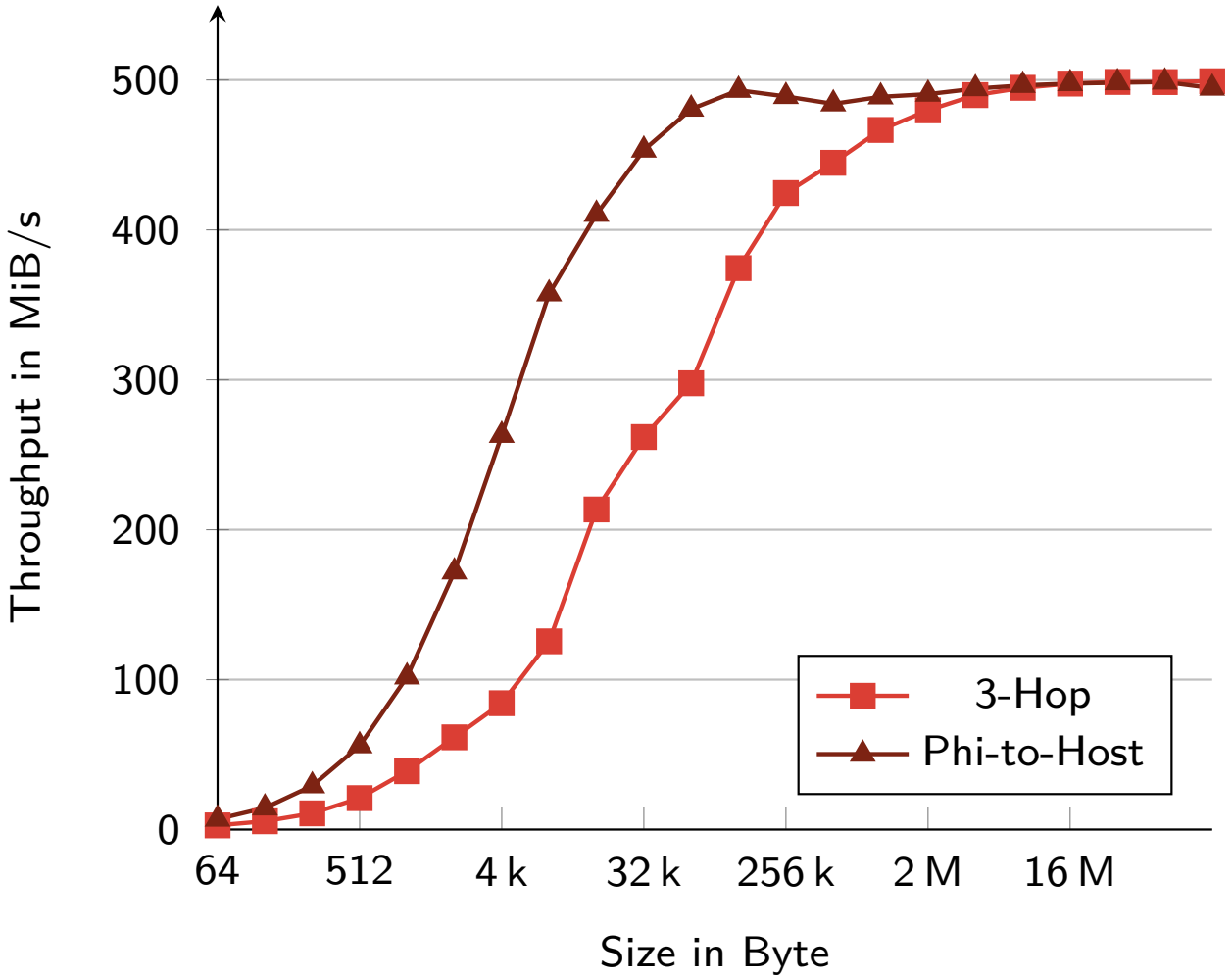
- Latencies accumulate
- Average throughput equals that of the bottleneck link



→ Asymetric links

→ Average throughput corresponds to the *harmonic* mean of the two bottleneck links

Multi-Hop PingPong – Throughput



What we have . . .

- Support for heterogeneous network landscapes
- High portability

- Supply of different programming models

- Consideration of the hardware's RDMA and RMA capabilities

- High performance

What we have . . .

- Support for heterogeneous network landscapes
- High portability
  - Device abstraction
  - Three devices: SCIF, SISCI, and SHMEM
- Supply of different programming models

- Consideration of the hardware's RDMA and RMA capabilities

- High performance

What we have . . .

- Support for heterogeneous network landscapes
- High portability
  - Device abstraction
  - Three devices: SCIF, SISCI, and SHMEM
- Supply of different programming models
  - Three-layered topology
  - Automatic node ID assignment
- Consideration of the hardware's RDMA and RMA capabilities

- High performance

What we have . . .

- Support for heterogeneous network landscapes
- High portability
  - Device abstraction
  - Three devices: SCIF, SISCI, and SHMEM
- Supply of different programming models
  - Three-layered topology
  - Automatic node ID assignment
- Consideration of the hardware's RDMA and RMA capabilities
  - One-sided communication
  - Zero-copy forwarding
- High performance

What we have . . .

- Support for heterogeneous network landscapes
- High portability
  - Device abstraction
  - Three devices: SCIF, SISCI, and SHMEM
- Supply of different programming models
  - Three-layered topology
  - Automatic node ID assignment
- Consideration of the hardware's RDMA and RMA capabilities
  - One-sided communication
  - Zero-copy forwarding
- High performance
  - Good latency results (asynchronous signaling)
  - Promising multi-hop throughput results

What we need . . .

- DMA over the whole platform
  - Implementation of `swift_put()/_get()` (w. i. p.)
- Multicast or PUB/SUB communication mode
- Dynamic routing (e. g. Bellman-Ford)

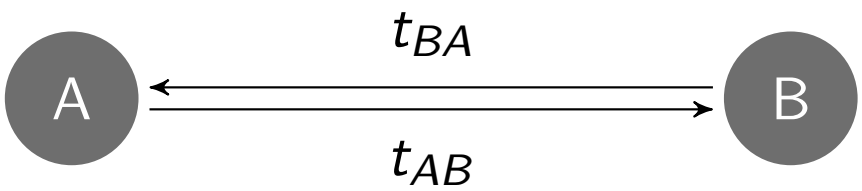
Thank you for your kind attention!

Simon Pickartz – spickartz@eonerc.rwth-aachen.de

Institute for Automation of Complex Power Systems
E.ON Energy Research Center, RWTH Aachen University
Mathieustraße 10
52074 Aachen

www.eonerc.rwth-aachen.de

Asymmetric Channels



- time to send a message of length L from A to B and back

$$T = \frac{L}{t_{AB}} + \frac{L}{t_{BA}}$$

- resulting throughput

$$t_{res} = \frac{L}{\frac{T}{2}} = \frac{2L}{\frac{L}{t_{AB}} + \frac{L}{t_{BA}}} = 2 \cdot \frac{t_{AB} \cdot t_{BA}}{t_{AB} + t_{BA}}$$

RDMA Results