

Designing Topology-Aware Collective Communication Algorithms for Large Scale InfiniBand Clusters: Case Studies with Scatter and Gather

Krishna Kandalla ⁽¹⁾, Hari Subramoni ⁽¹⁾, Abhinav
Vishnu⁽²⁾, and Dhabaleswar. K. Panda⁽¹⁾

⁽¹⁾Computer Science & Engineering Department,
The Ohio State University

⁽²⁾ High Performance Computing Group,
Pacific Northwest National Laboratory

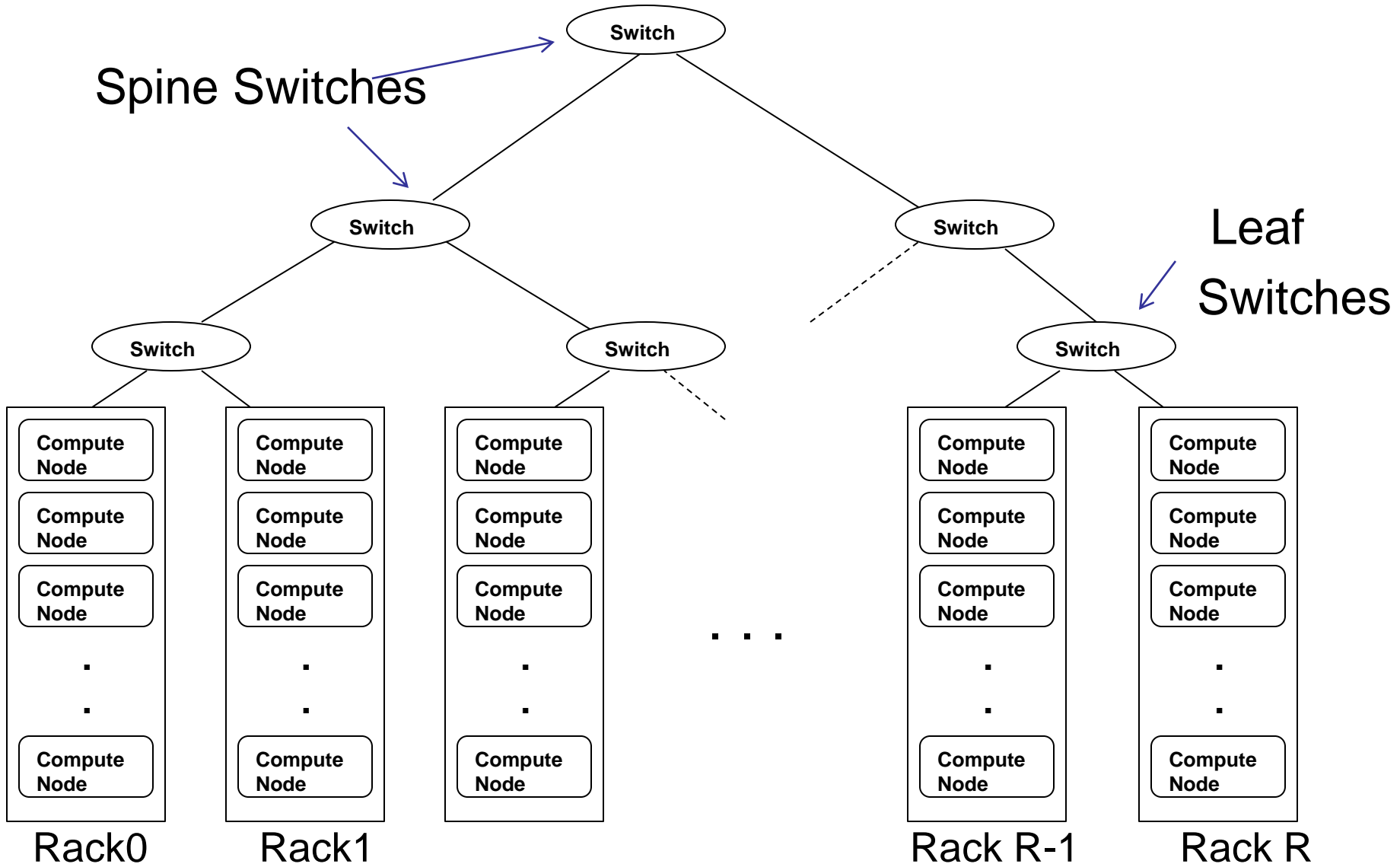
Introduction and Background

- Compute nodes are getting fatter with increasing core-count
- Communication networks have varied topologies, also getting faster and fatter
- Message Passing Interface (MPI) is the de-facto parallel programming model
- Many parallel applications can now scale up to tens of thousands of processes

Introduction and Background

- But, Wait! Are we really making the best use of these systems?
- Applications spend a considerable amount of time in communication operations
- How does the network topology affect the communication costs?
- *Can we design communication operations in a “topology-aware” manner?*

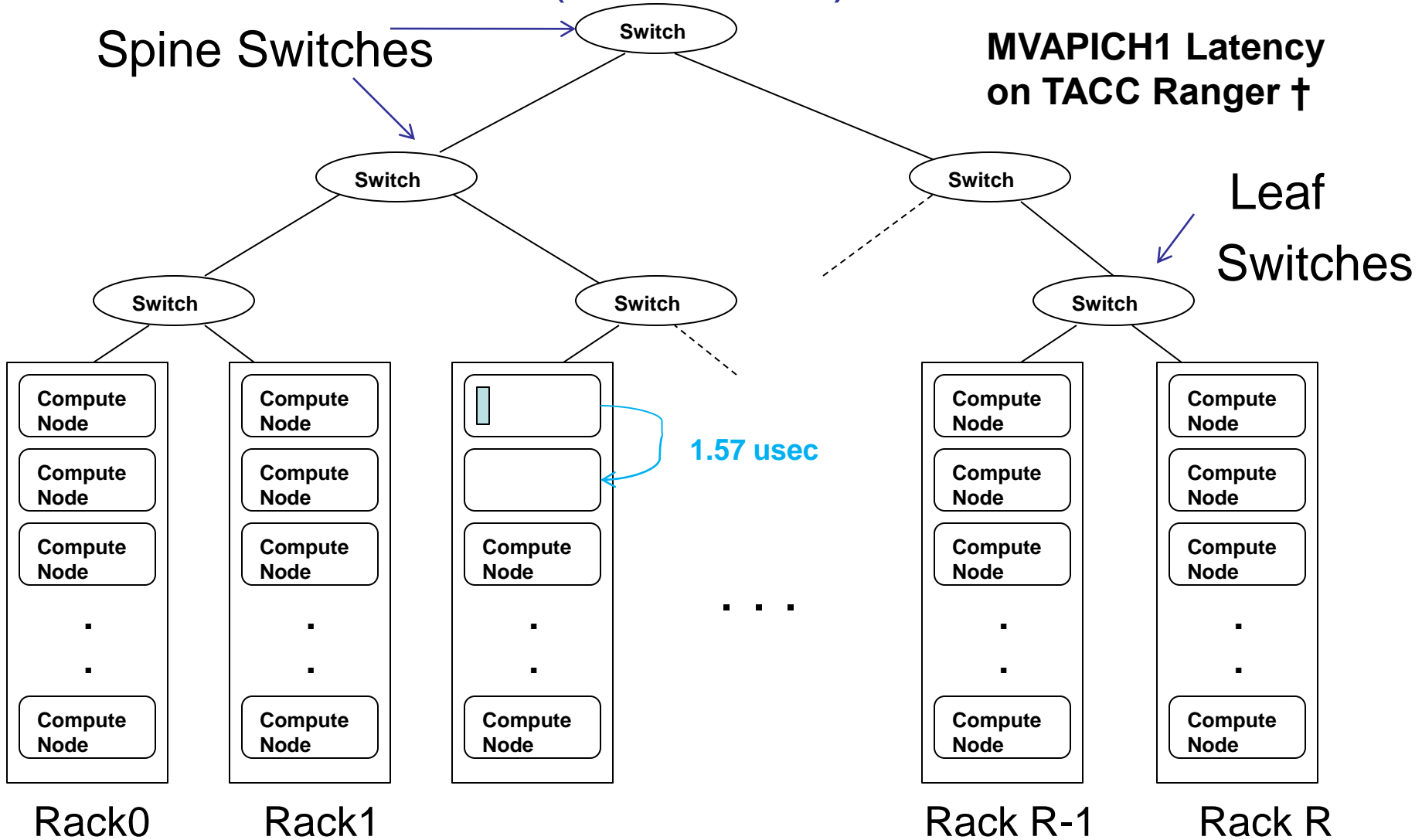
Typical Topology of Large Scale Clusters



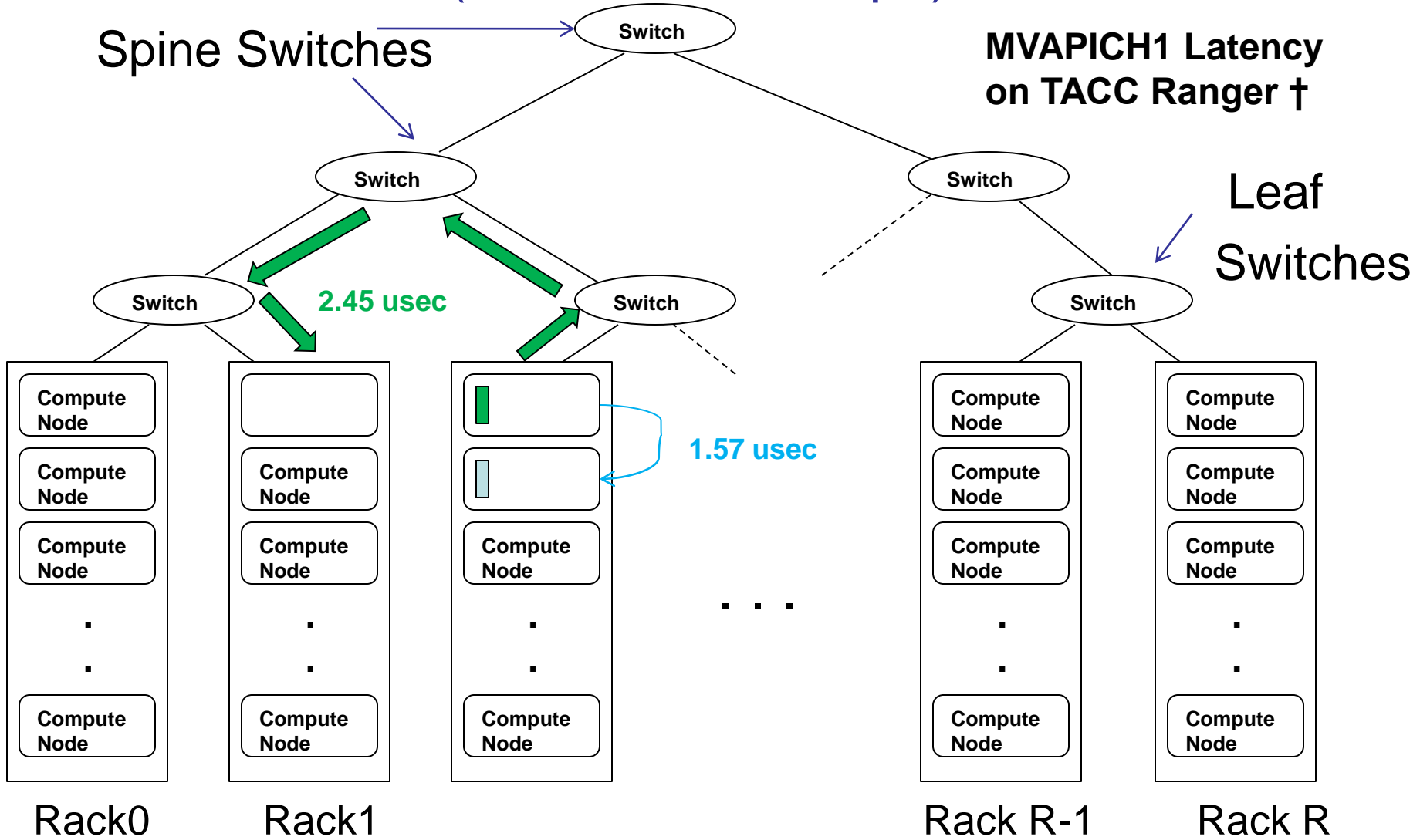
Outline

- Introduction and Background
- **Motivation**
- Problem Statement
- Designing “Topology-Aware” Collective Algorithms
- Experimental Evaluation
- Conclusions and Future Work

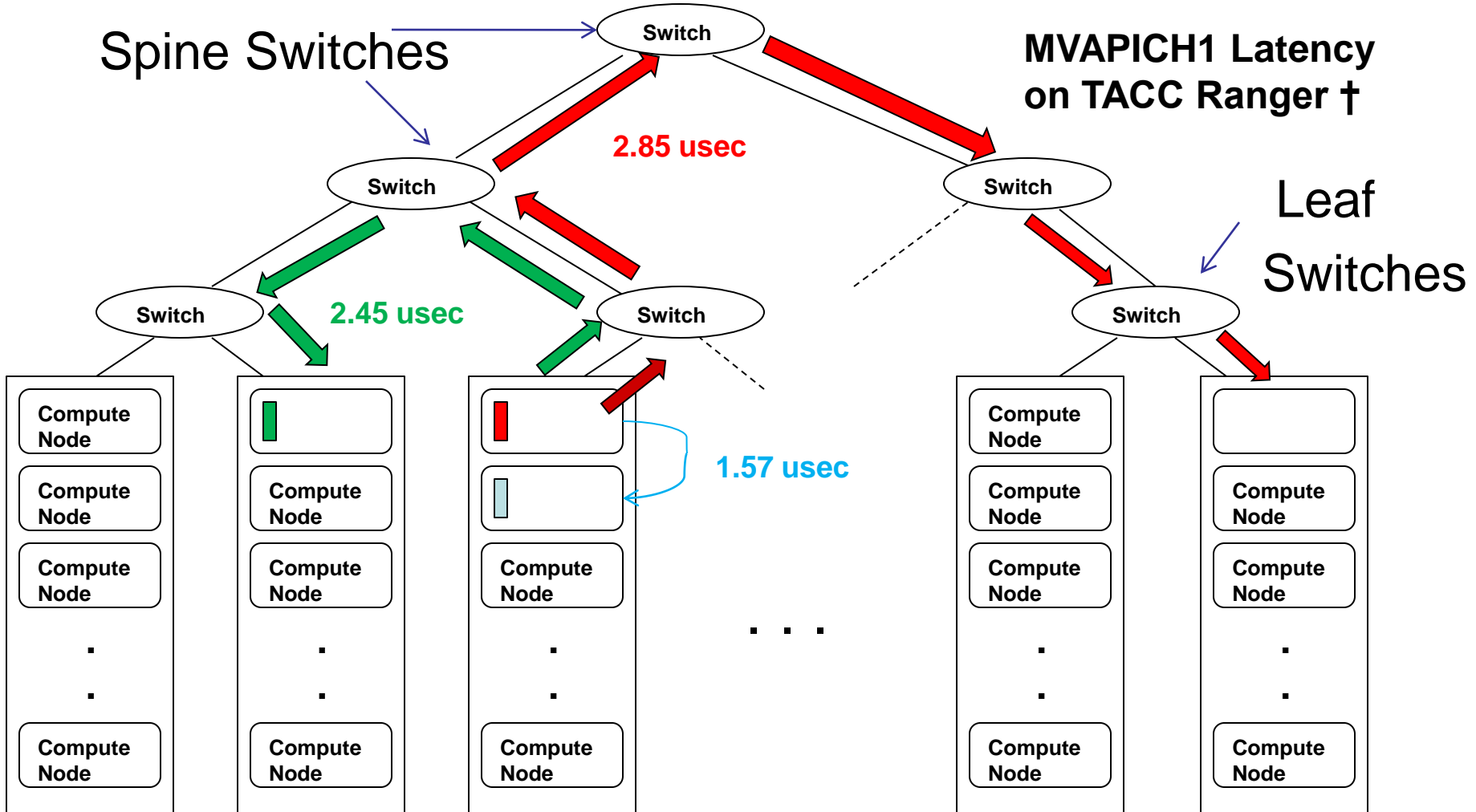
Cluster Topology and Communication Latency (Intra-Rack)



Cluster Topology and Communication Latency (Inter-Rack 3 Hops)

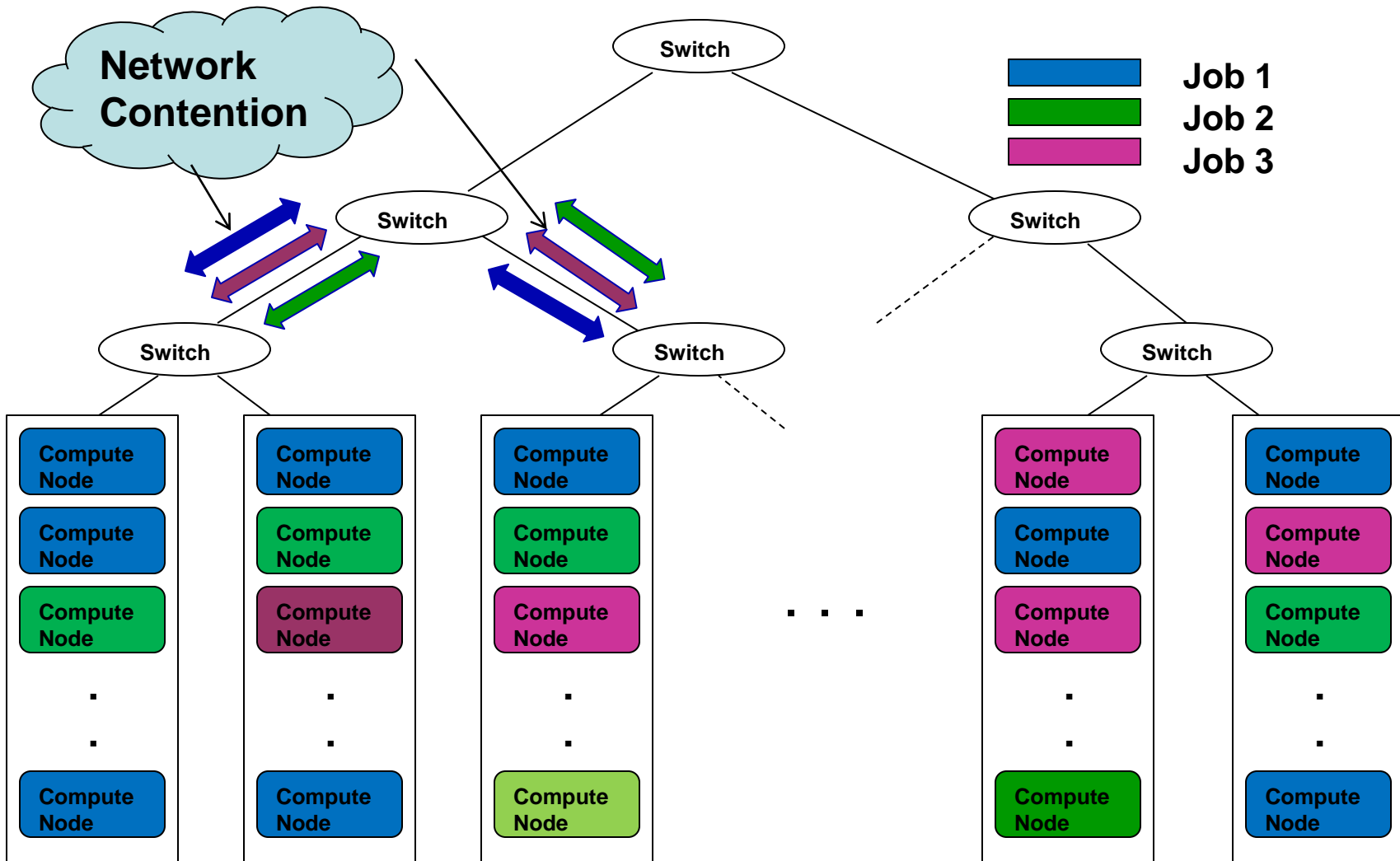


Cluster Topology and Communication Latency (Inter-Rack 5 Hops)



† Thanks to Dr. Bill Barth and Dr. Karl W. Schulz @ TACC

Network Sharing between different jobs



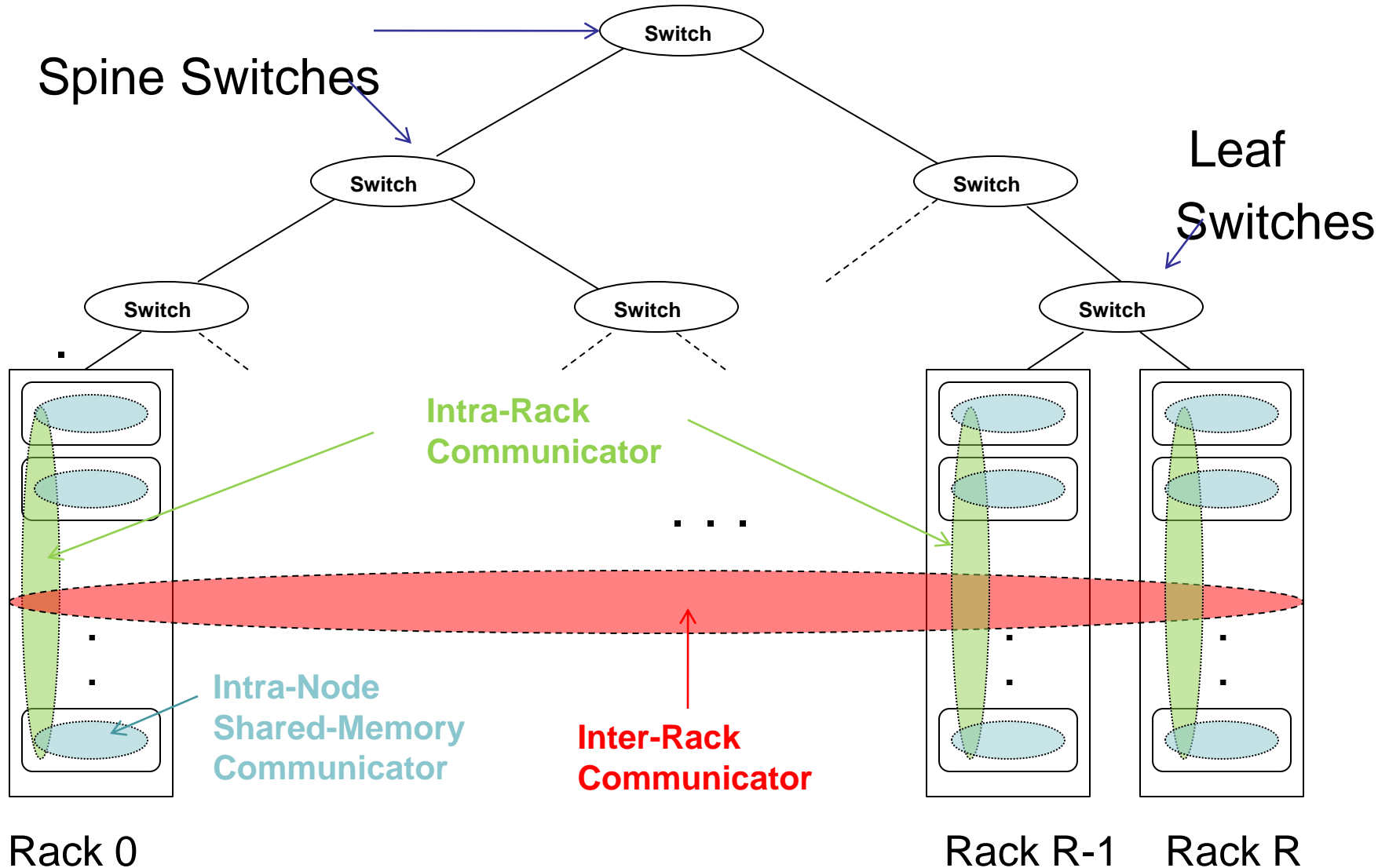
Problem Statement

- How do we design collective algorithms in a “*Topology-Aware*” manner to minimize the communication costs?
- What is the impact of background traffic on the performance of the existing collective algorithms?
- Can we design collective algorithms to be “*resilient*” to network traffic?

Outline

- Introduction and Background
- Motivation
- Problem Statement
- Designing “Topology-Aware” Collective Algorithms
- Experimental Evaluation
- Conclusions and Future Work

Designing Topology-Aware Collective Algorithms – Sub-Communicator Creation

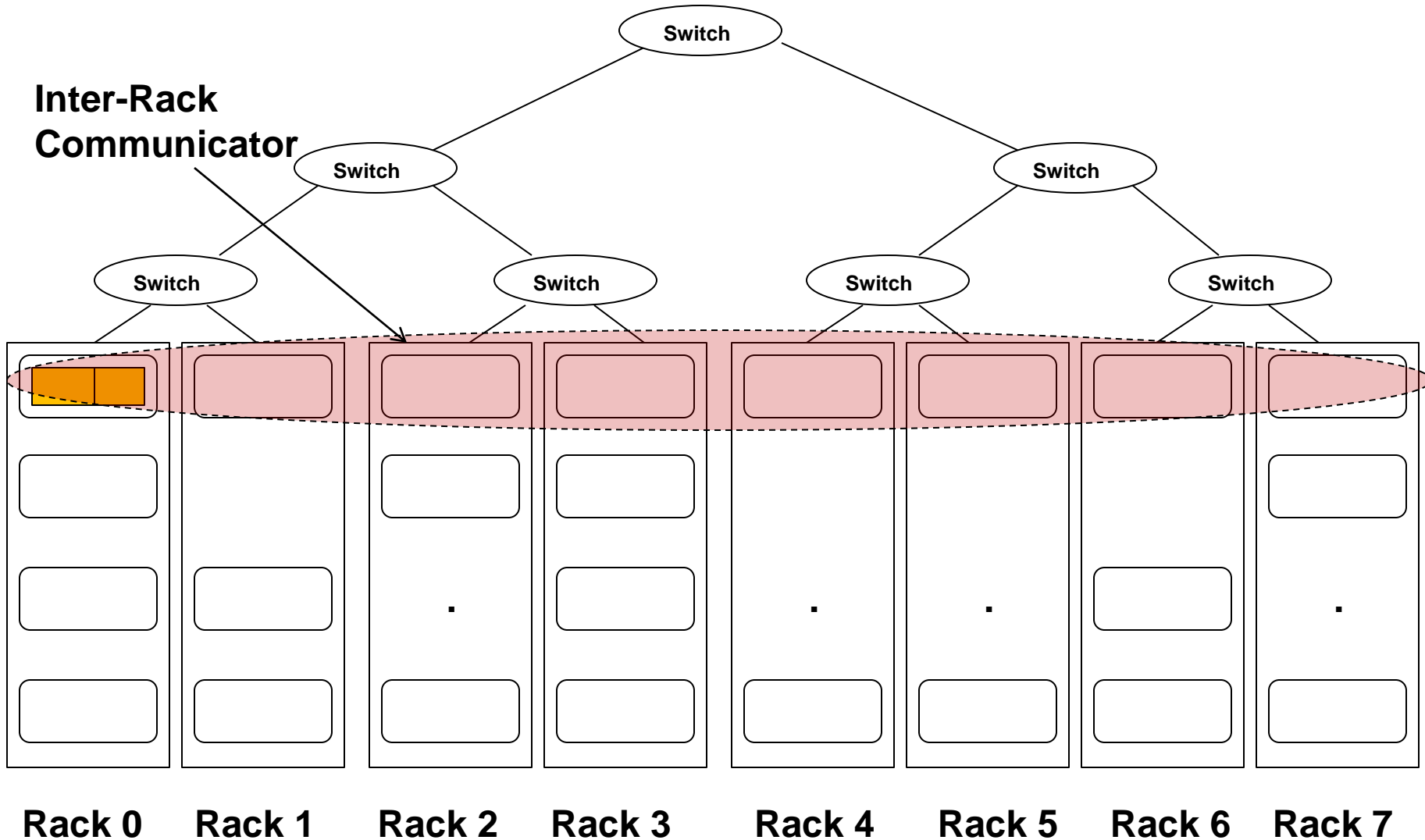


Rack 0

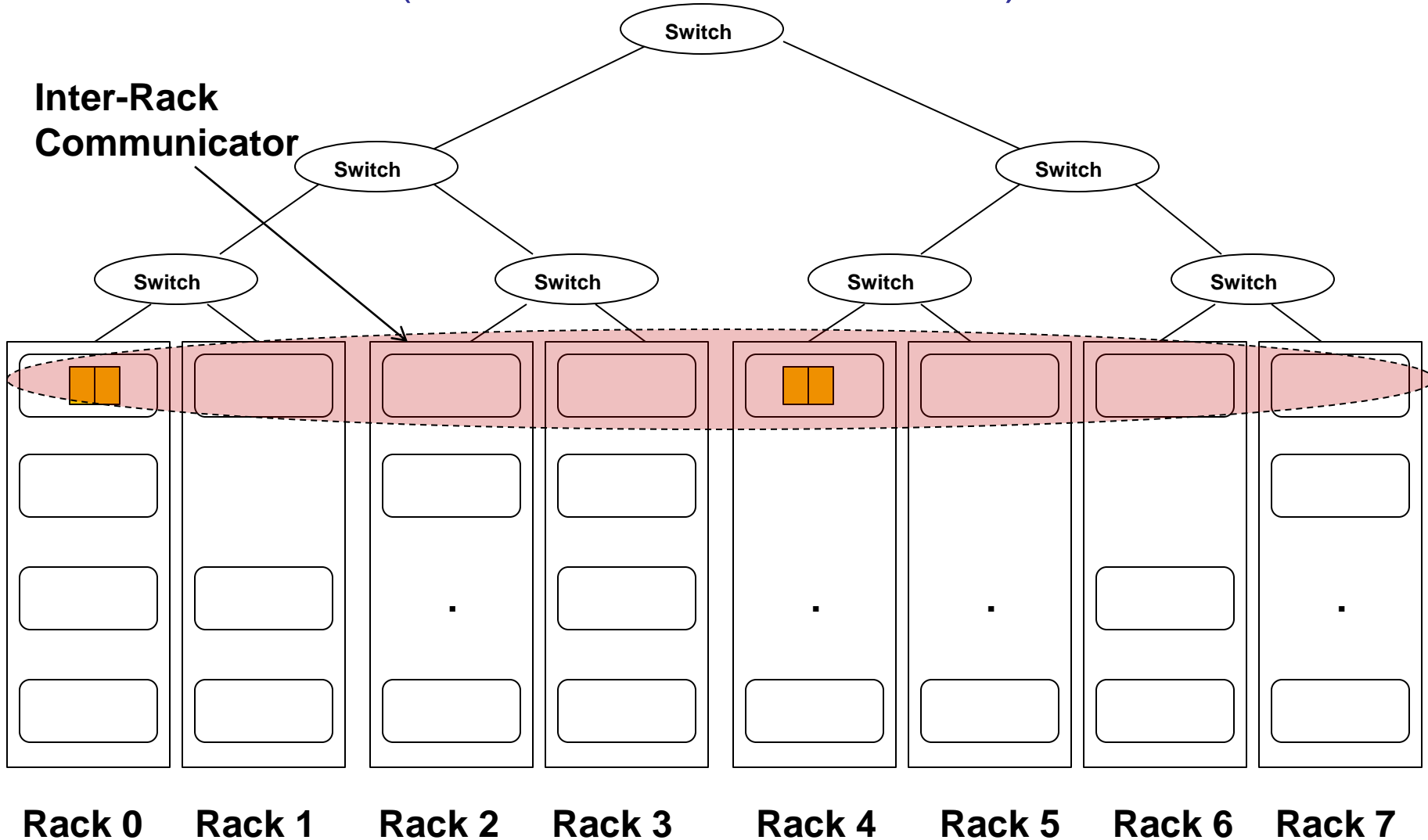
Rack R-1

Rack R

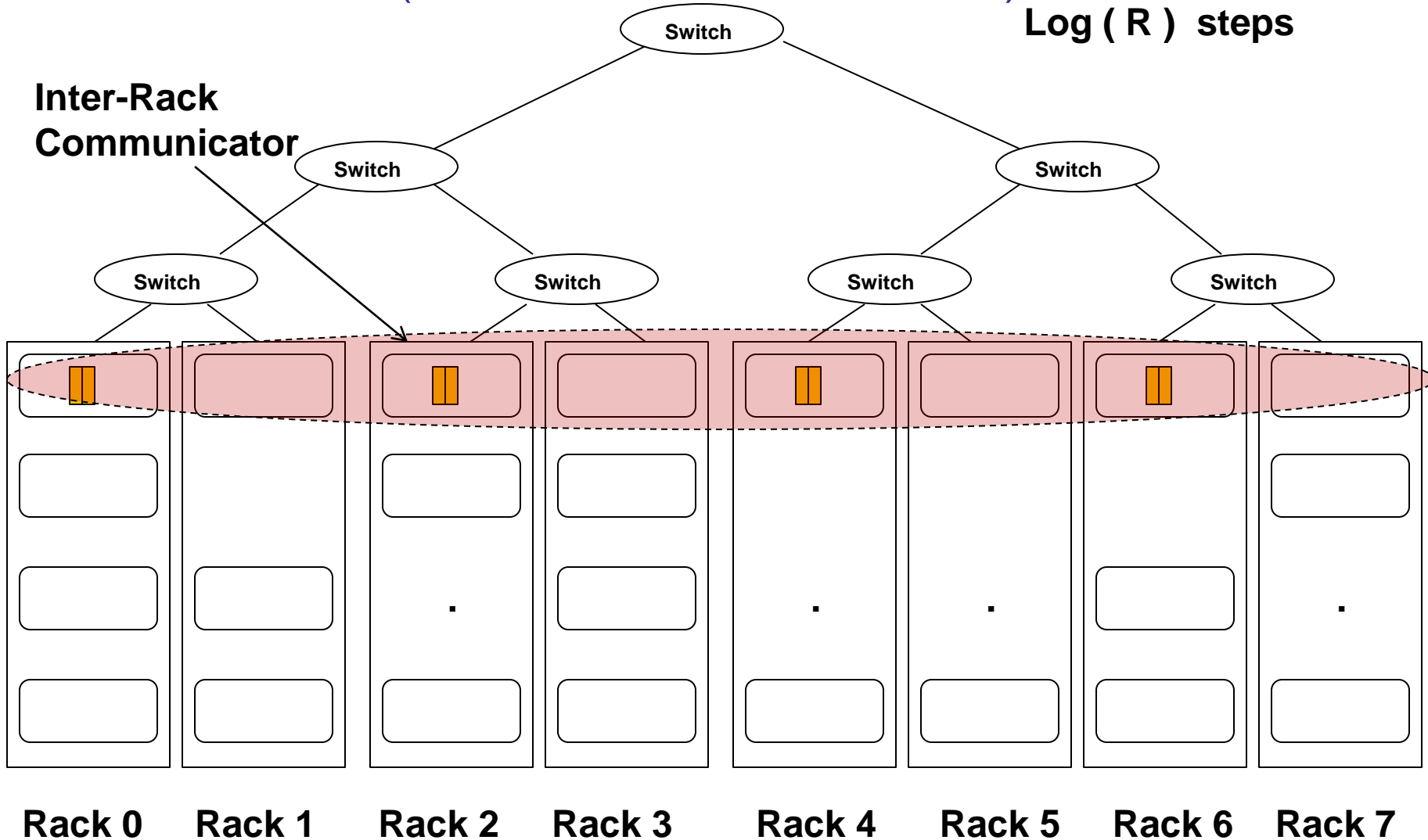
Communication Pattern of Proposed Topology-Aware One-to-All Operations (Inter-Rack Communication)



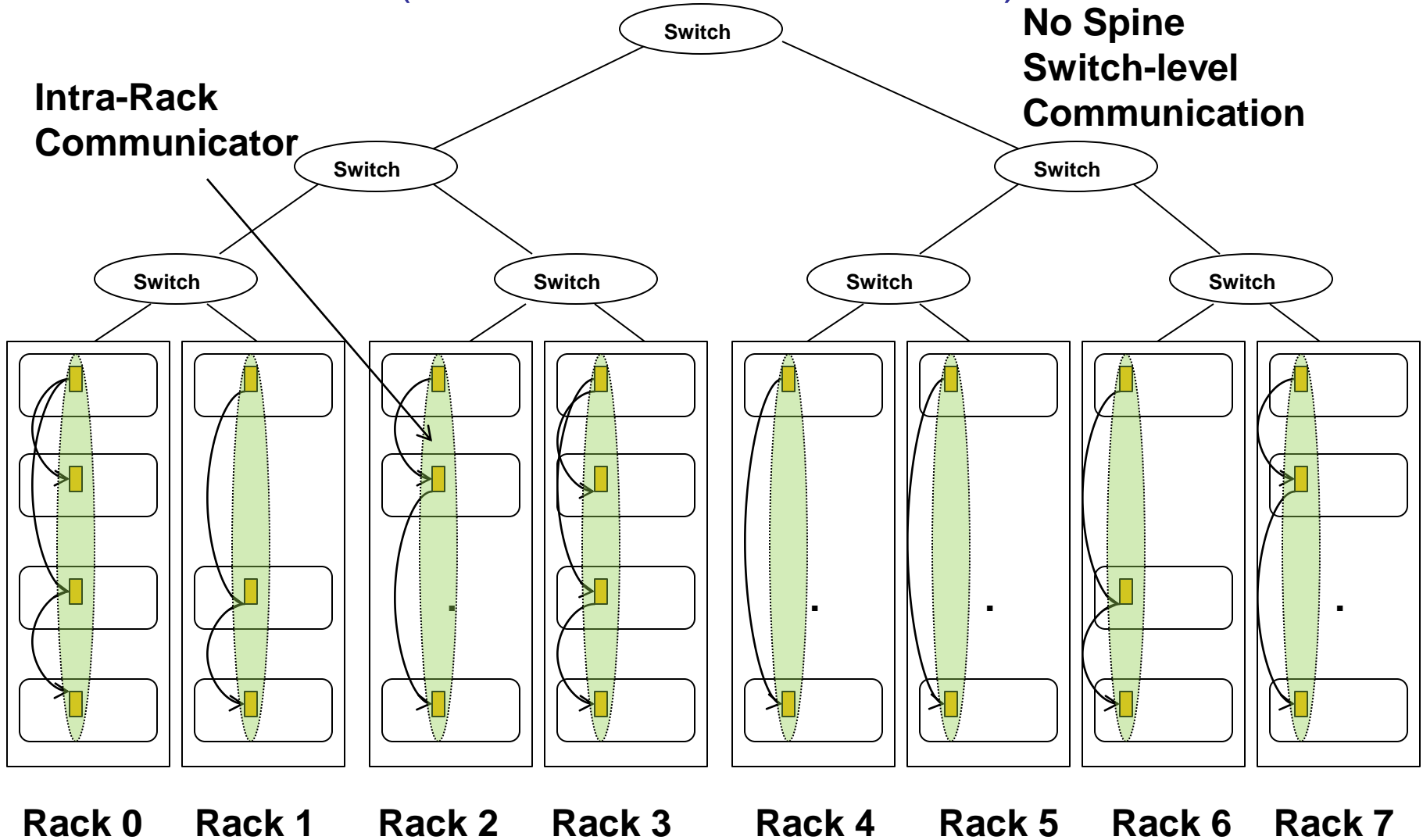
Communication Pattern of Proposed Topology-Aware One-to-All Operations (Inter-Rack Communication)



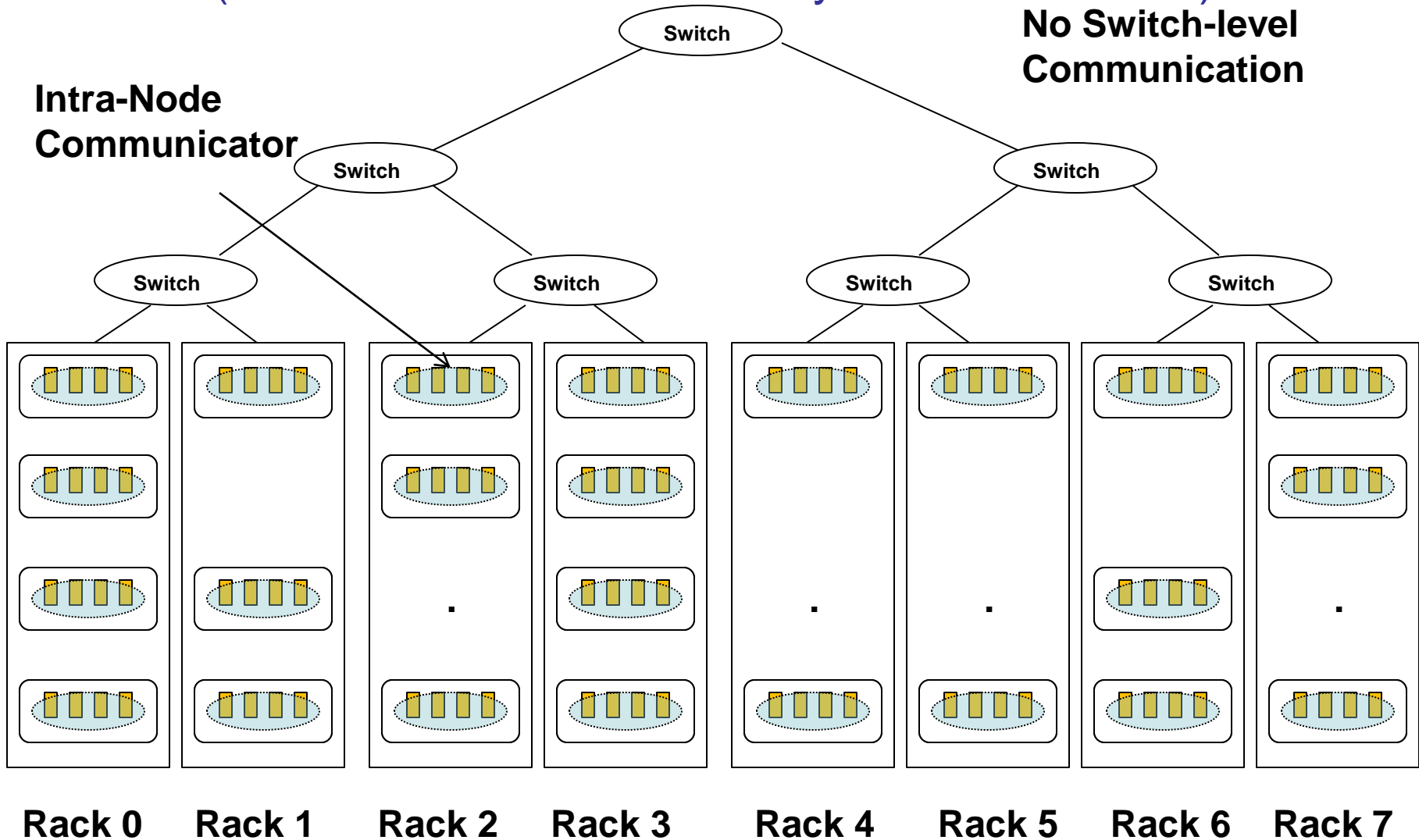
Communication Pattern of Proposed Topology-Aware One-to-All Operations (Inter-Rack Communication)



Communication Pattern of Proposed Topology-Aware One-to-All Operations (Intra-Rack Communication)



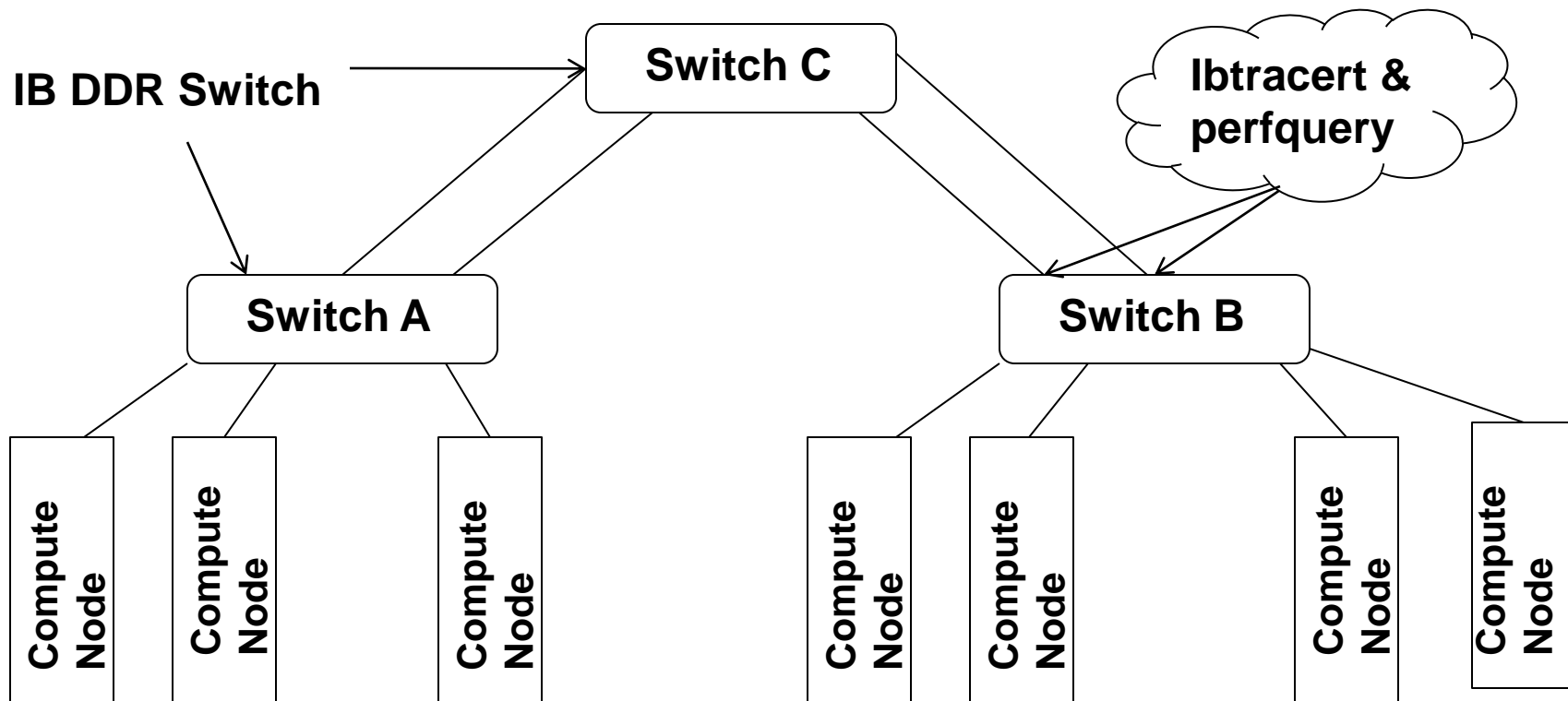
Communication Pattern of Proposed Topology-Aware One-to-All Operations (Intra-Node Shared-Memory Communication)



Outline

- Introduction and Background
- Motivation
- Problem Statement
- Designing “Topology-Aware” Collective Algorithms
- Experimental Evaluation
- Conclusions and Future Work

Experimental Evaluation



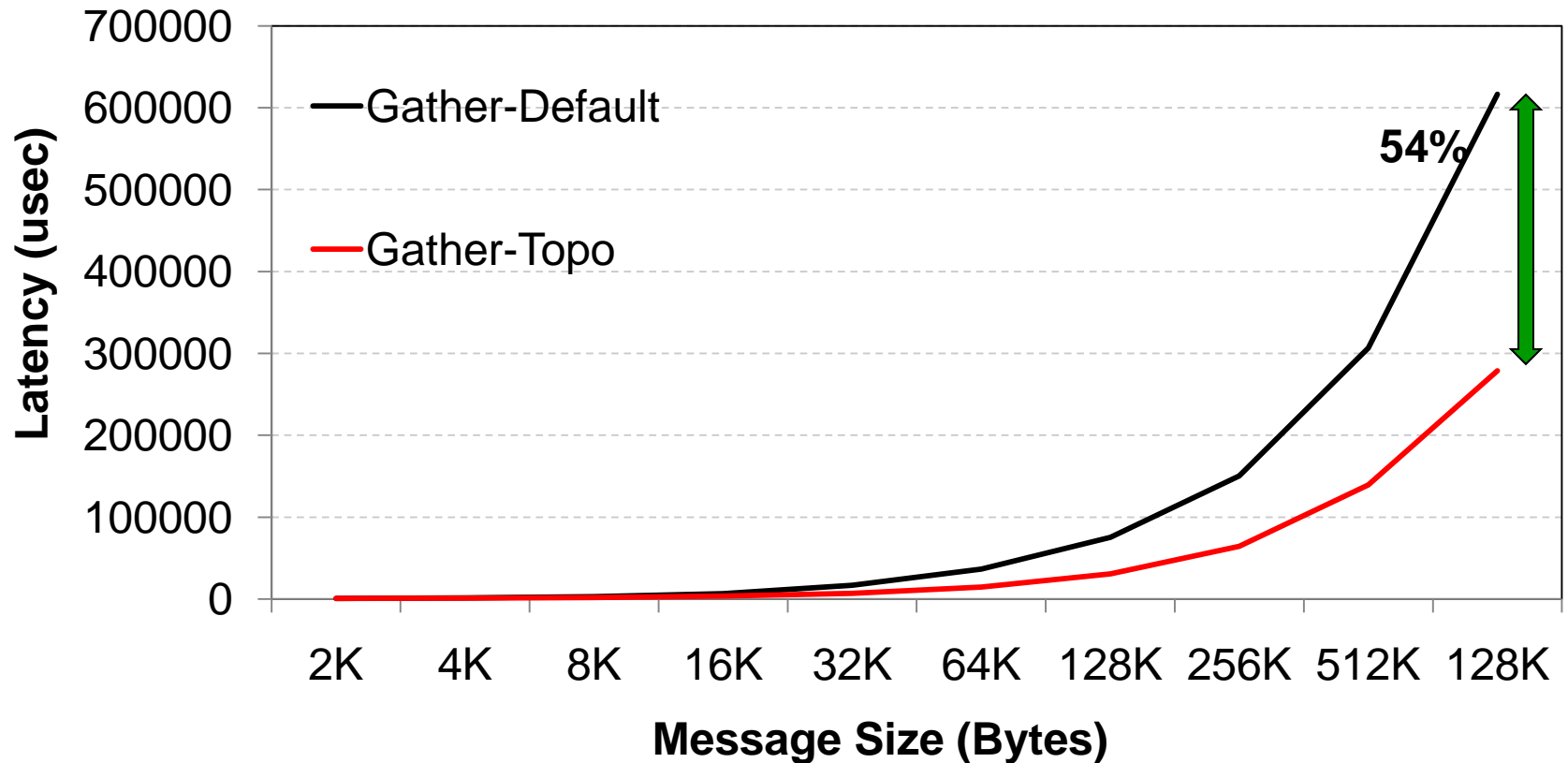
Cluster A : AMD Barcelona

- 8 nodes
- 16 cores/node
- ConnectX DDR HCA's

Cluster B : Intel Clovertown

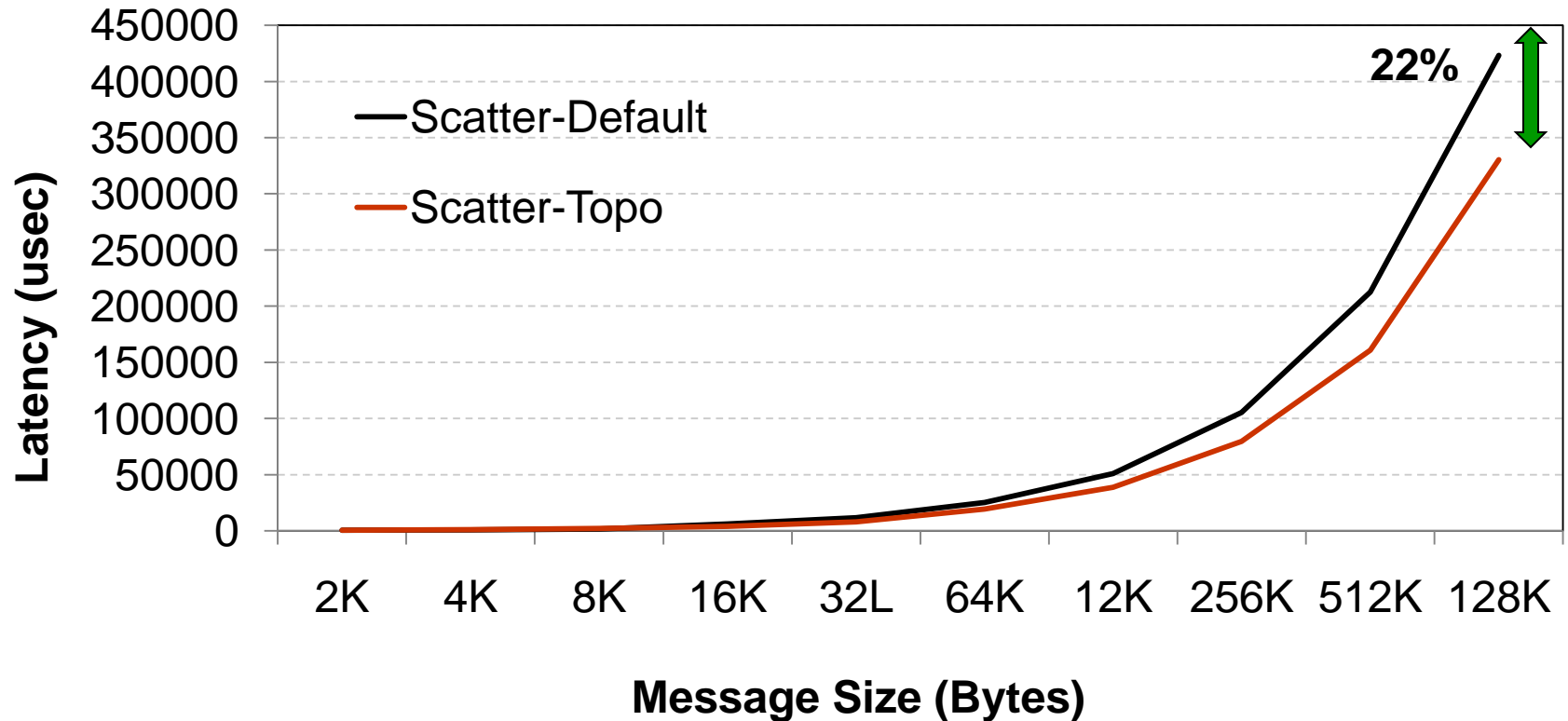
- 29 nodes
- 8 cores/node
- ConnectX DDR HCA's

Topology-Aware MPI_Gather



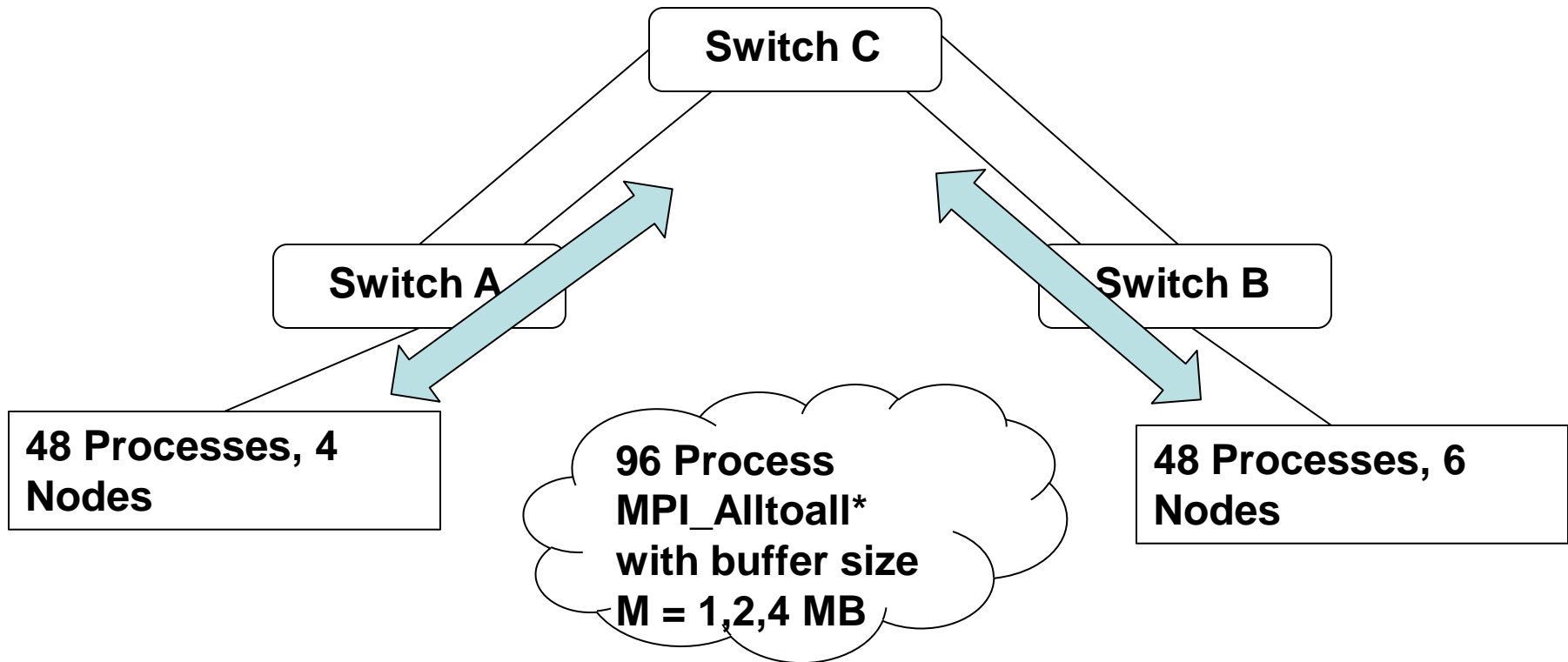
Topology-Aware MPI_Gather across 296 Processes : Performance improvement of almost 54% (Quiet Conditions)

Topology-Aware MPI_Scatter



Topology-Aware MPI_Scatter across 296 Processes : Performance improvement of almost 22% (Quiet Conditions)

Experimental Methodology – Background Traffic



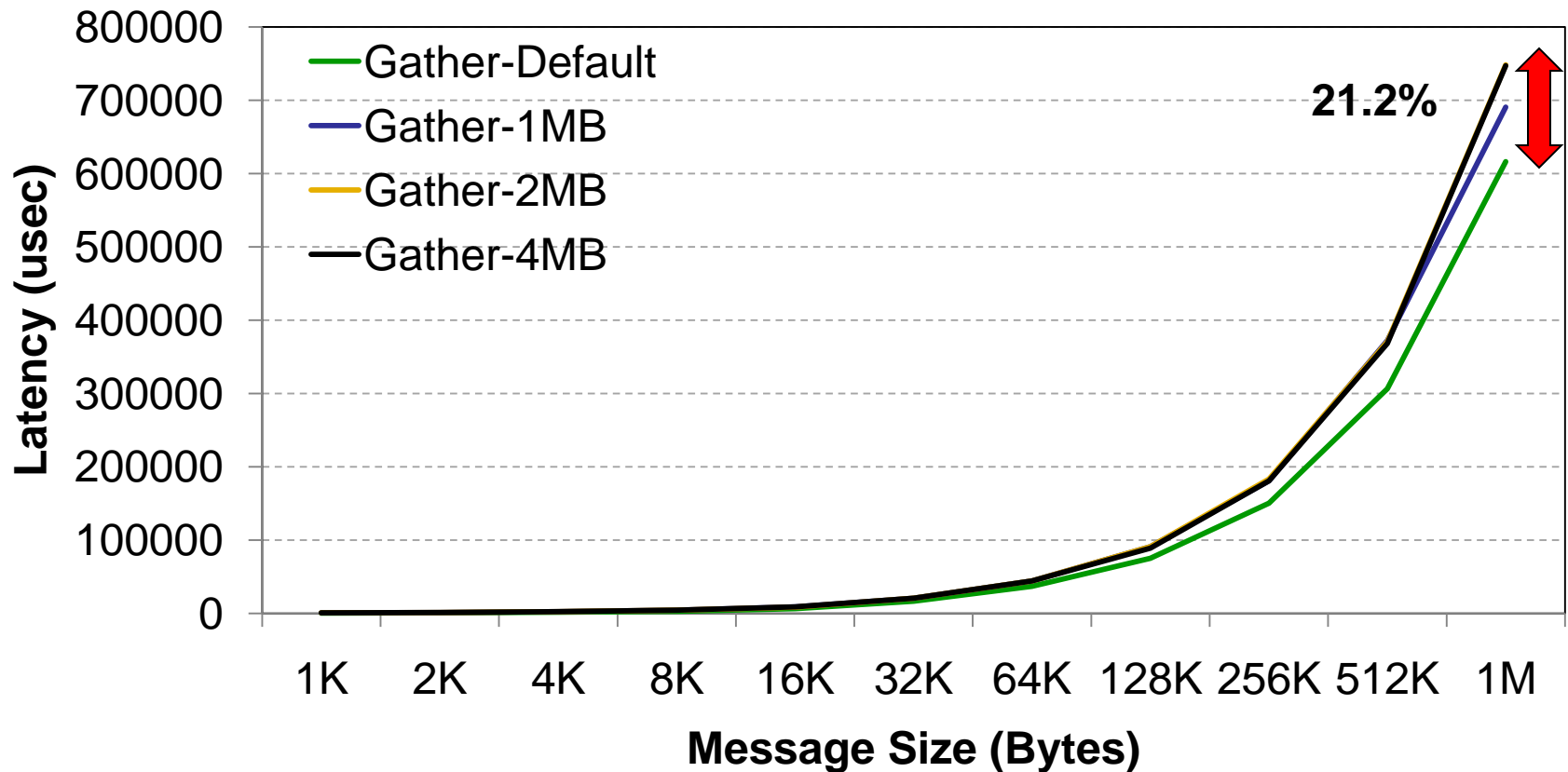
Cluster A : AMD Barcelona

- 8 nodes
- 16 cores/node
- ConnectX DDR HCA's

Cluster B : Intel Clovertown

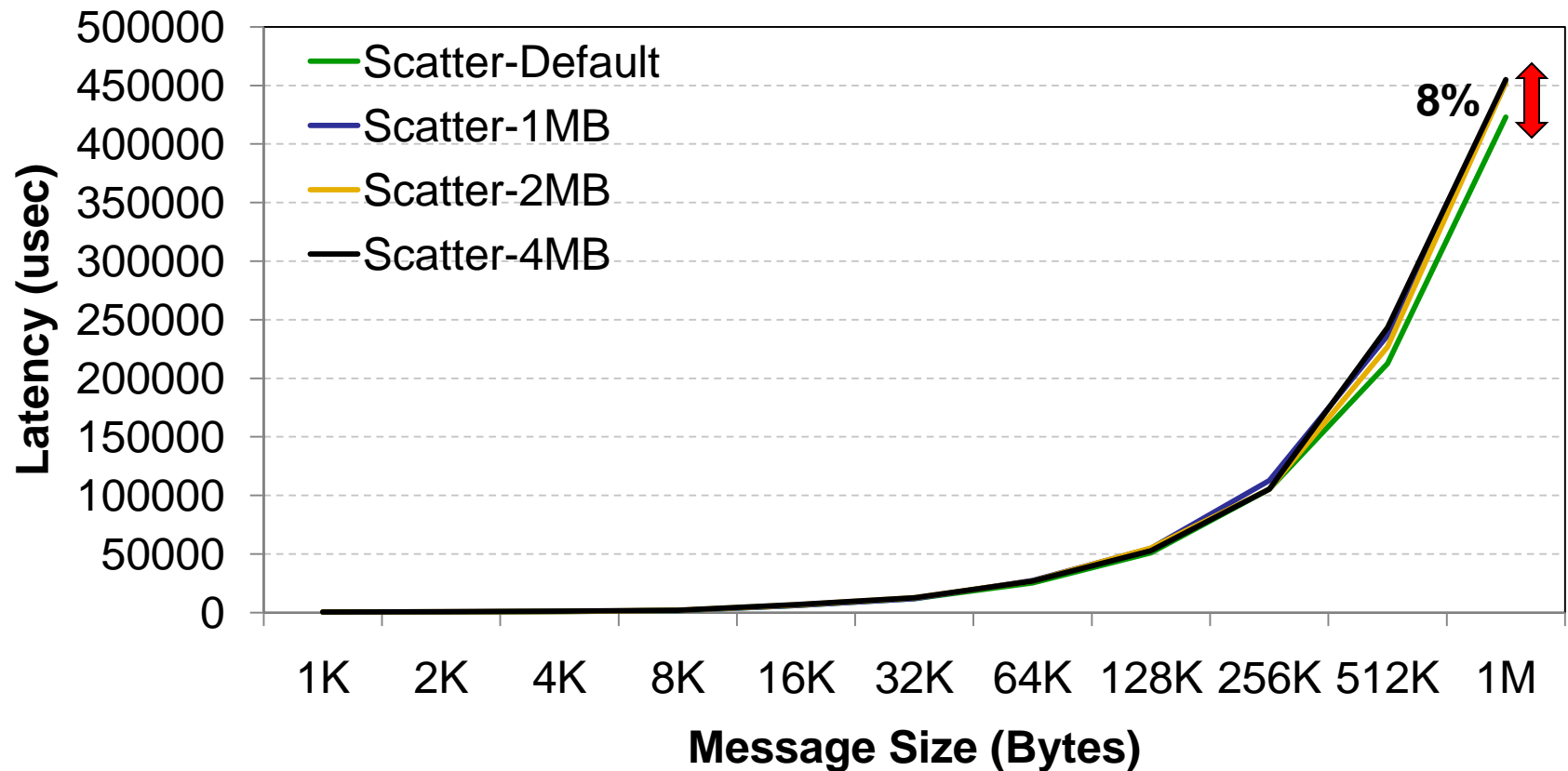
- 29 nodes
- 8 cores/node
- ConnectX DDR HCA's

Impact of Network Traffic on MPI_Gather



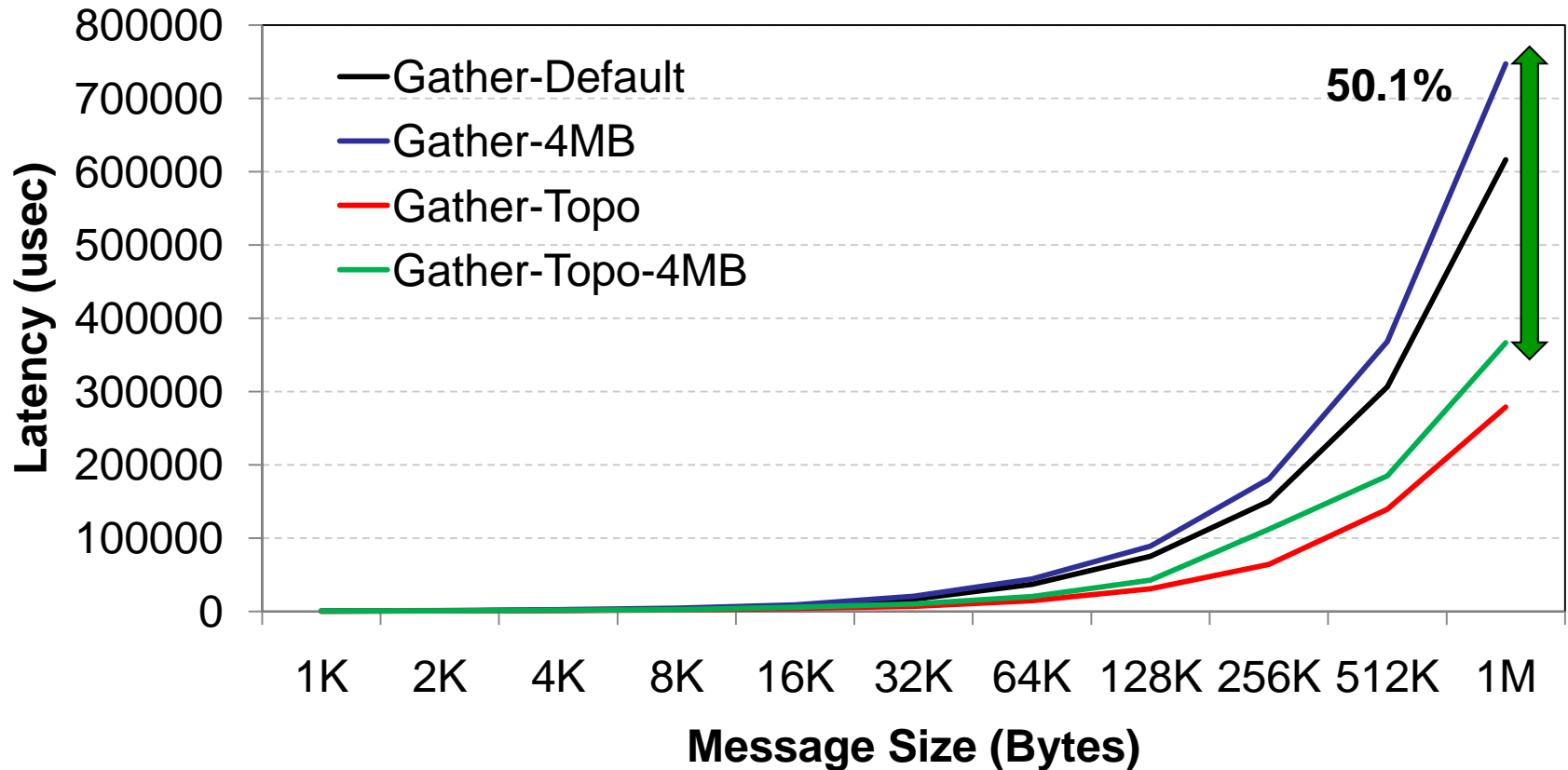
MPI_Gather across 296 Processes : Performance degradation of 21% when the background process is a 4MB MPI_Alltoall job

Impact of Network Traffic on MPI_Scatter



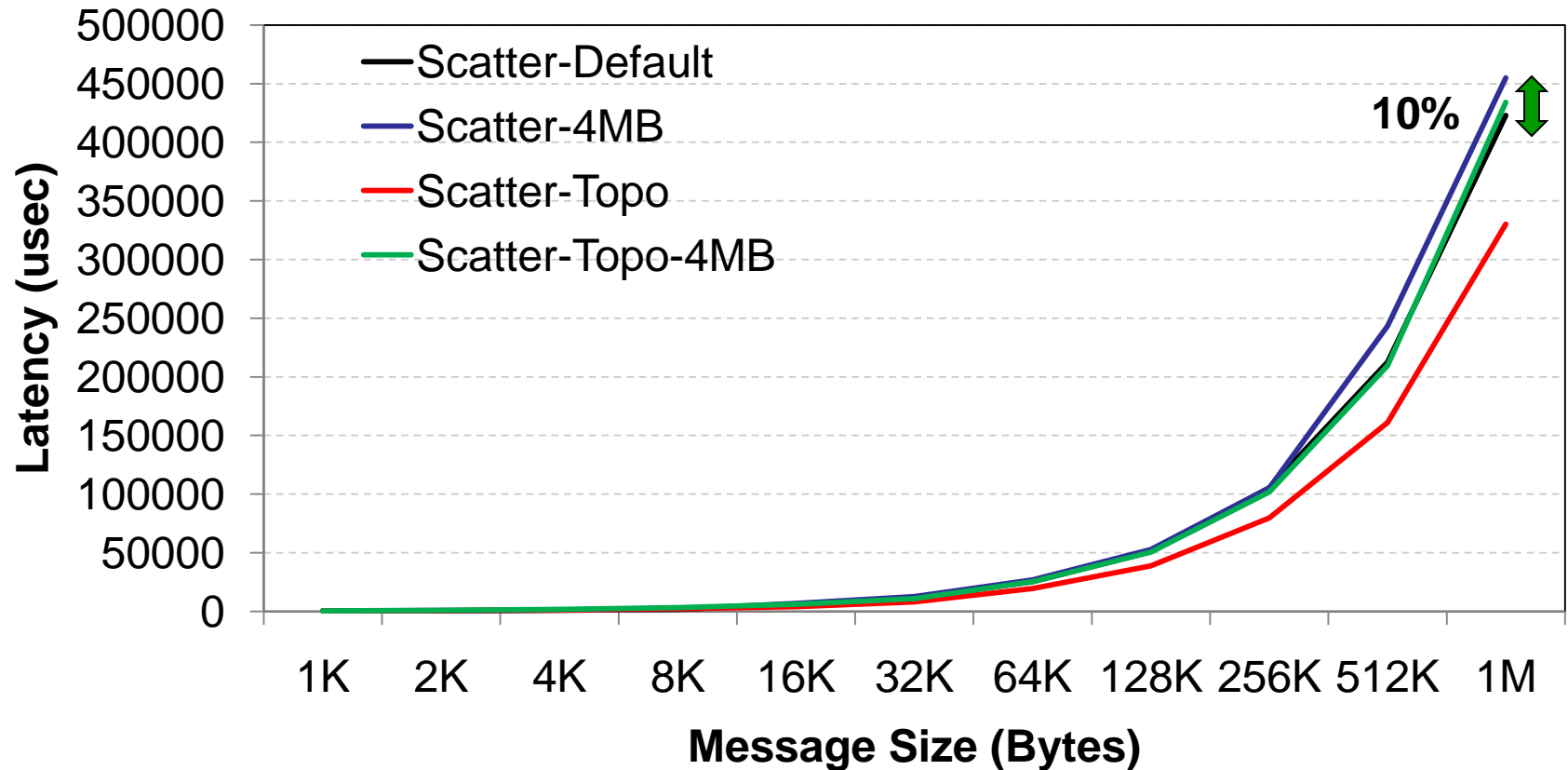
MPI_Scatter across 296 Processes : Performance degradation of 8% when the background process is a 4MB MPI_Alltoall job

Topology-Aware MPI_Gather with Background Traffic



Topology-Aware MPI_Gather across 296 Processes when the background process is a 4 MB Alltoall job : Performance improvement of about 50%

Topology-Aware MPI_Scatter with Background Traffic



Topology-Aware MPI_Scatter across 296 Processes when the background process is a 4 MB Alltoall job : Performance improvement of about 10%

Outline

- Introduction and Background
- Motivation
- Problem Statement
- Designing “Topology-Aware” Collective Algorithms
- Experimental Evaluation
- Conclusions and Future Work

Conclusions

- Process Topology significantly affects the performance of point-to-point and collective operations
- Background traffic due to the presence of other jobs in the system can affect the performance of dense collective operations by almost **21%**
- Our proposed topology-aware collective algorithms can outperform the default algorithms by almost **50%** under both quiet and busy conditions

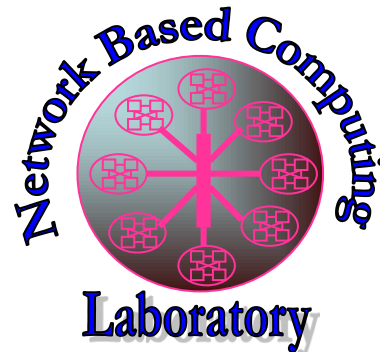
Future Work

- Explore designs to dynamically detect topology in an efficient manner
- Extend these designs to other collectives and study the performance benefits at large scales
- Evaluate the performance benefits with applications at large scales
- Incorporate the proposed “topology-aware” designs in the upcoming MVAPICH/MVAPICH2 releases



<http://mvapich.cse.ohio-state.edu>

Thank you !



¹{kandalla, subramon, panda}@cse.ohio-state.edu

²abhinav.vishnu@pnl.gov

¹Network-Based Computing Laboratory, Ohio State University

²High Performance Computing Group, Pacific Northwest National Laboratory

Back-up Slides

Related Work

- Topology-Awareness in Collective algorithms have been proposed for Grids [Coti et al, Craig A. Lee, B. Jakimovski et. al]
- Topology Detection in Ethernet based clusters was proposed by L. Lawrence et al
- Patarasuk et al proposed Bandwidth efficient Allreduce algorithms on Tree based networks
- In our study, we focus on topology detection and topology-aware collective algorithms for tightly-coupled InfiniBand networks

Detecting InfiniBand Topology

- InfiniBand Subnet Manager allocated unique LID for each active device in the network
- “ibnetdiscover” provides a mapping between all the active ports and their connections
- During MPI_Init, we parse this information to learn about the set of processes that are connected to the same leaf-level switch
- Create sub-communicators within the MPI library to reflect the topology

Designing Topology-Aware Collective Algorithms

Sub Communicator Creation

- * Intra-Node Leader
- * Intra-Rack Leader

