The Case for an RDMA Extension to MPI

Ron Brightwell

Sandia National Laboratories

Matthew L. Curry

University of Alabama at Birmingham

Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

Remote Direct Memory Access (RDMA)

- One-sided data movement
  - Allows a process to read and write remote memory without explicit interaction during data transfer
- Provides several important benefits
  - True zero-copy data transfer
    - No intermediate host-side buffering
  - Low CPU overhead
    - Decouples host processor from network
  - Fixed memory resources
    - No “unexpected” messages
  - Supports unstructured, non-blocking data transfer
    - Completion of transfer is a local event

RDMA-Capable Networks

- InfiniBand
- InfiniPath
- Myrinet
- Quadrics
- Ethernet (iWarp, RNIC-PI)
- Cray XT3 and XD1
- IBM systems
- (just about every message passing system)

Limitations of MPI Point-to-Point Over RDMA

- Two-copy data transfer for small messages
  - Memory registration costs lead to memcpy()'s at sender and receiver
- Zero-copy data transfer for large messages
  - Requires MPI-level flow control
- CPU overhead on every transfer
  - Host processor must do MPI context/tag matching
- Non-persistent user buffers need to be managed by MPI library
  - Buffering strategies do not scale

Limitations of MPI One-Sided Over RDMA (1)

- MPI one-sided is not one-sided
- Three different synchronization models
  - Fence and Post models must explicitly synchronize with target
  - Lock model requires active lock manager at target
- Performance overhead of synchronization is significant
- Cost of explicit synchronization usually greater than implicit synchronization of point-to-point operations

Limitations of MPI One-Sided Over RDMA (2)

- Synchronization overhead can be optimized by aggregating transfers until synchronization point is reached
  - Synchronization cost goes down at the expense of latency, overlap, and overhead
- Does not support unstructured, non-blocking data movement with local completion

Mismatch Between RDMA and MPI

- RDMA does not support MPI point-to-point well
  - Most of the significant benefits of RDMA are lost
- MPI does not support RDMA at all
  - MPI one-sided semantics do not fully exploit RDMA
- But RDMA is a reasonable model for many applications

Proposed MPI Extension for RDMA

- **RDMA write operation**
  - `MPI_RDMA_Write(buf,count,datatype,dest,offset,window);`
  - `MPI_RDMA_Iwrite(buf,count,datatype,dest,offset,window,request);`
- **RDMA read operation**
  - `MPI_RDMA_Read(buf,count,datatype,dest,offset,window);`
  - `MPI_RDMA_Iread(buf,count,datatype,dest,offset,window,request);`
- **Local window synchronization**
  - `MPI_RDMA_Win_test(buf,valp,datatype,boolean_op,win,flag);`
  - `MPI_RDMA_Win_wait(buf,valp,datatype,boolean_op,win);`
  - `Boolean_op = {MPI_OP_EQ,MPI_OP_NE,MPI_OP_LT,MPI_OP_GT, … }`

Benefits of MPI Extension for RDMA

- Maps directly to underlying RDMA calls
- Preserves all of the benefits of RDMA at the application level
- Provides a much-needed capability to MPI
- Adapting MPI to RDMA is arguably much easier than adapting RDMA to MPI

Why Not Just Mix RDMA with MPI?

- No way to coordinate RDMA resources
  - For some networks, this means
    - Duplicate connections
    - No common memory registration cache
- MPI behavior is undefined
  - Blocking on a RDMA completion event may stop all MPI communication from making progress

Evaluation Platforms

- Cray XT3 (Red Storm)
  - 2.0 GHz AMD Opteron
  - 4 GB main memory
  - SeaStar 1.2 network
  - MPICH2 0.97
- InfiniBand cluster
  - Dual 3.6 GHz Intel Xeon EM64T
  - 6 GB main memory
  - X8 PCI Express
  - MVAPICH2-0.6.5

Benchmarks

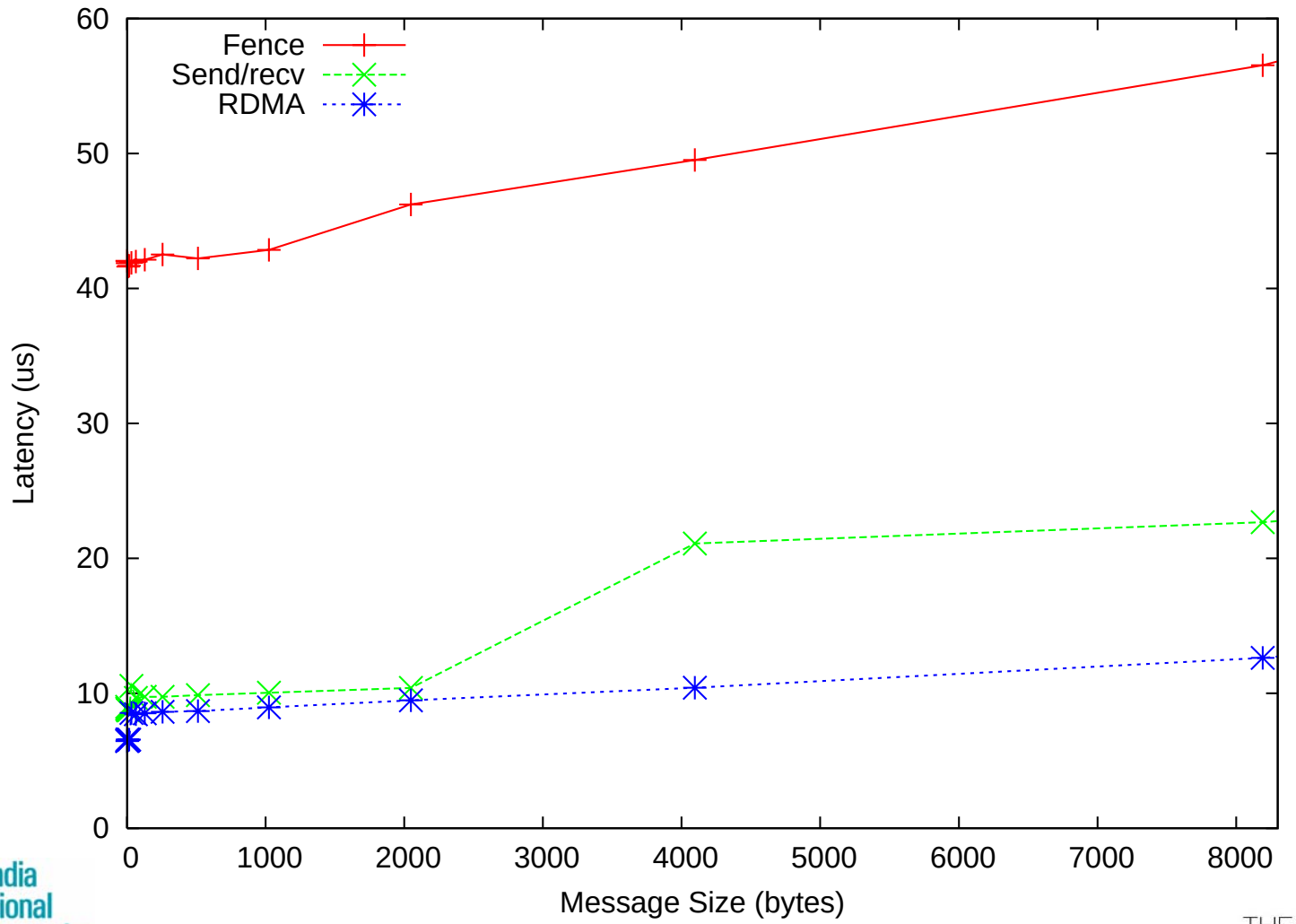
- Ping-pong latency
  - Ring-based ping-pong exchange between all nodes
- Nearest-neighbor ghost-area exchange
  - Test code from Argonne used to evaluate one-sided and point-to-point operations
- CPU availability
  - Calculates percentage of CPU available at receiver by doing a fixed amount of work during message arrival

Cray XT3 Results

Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

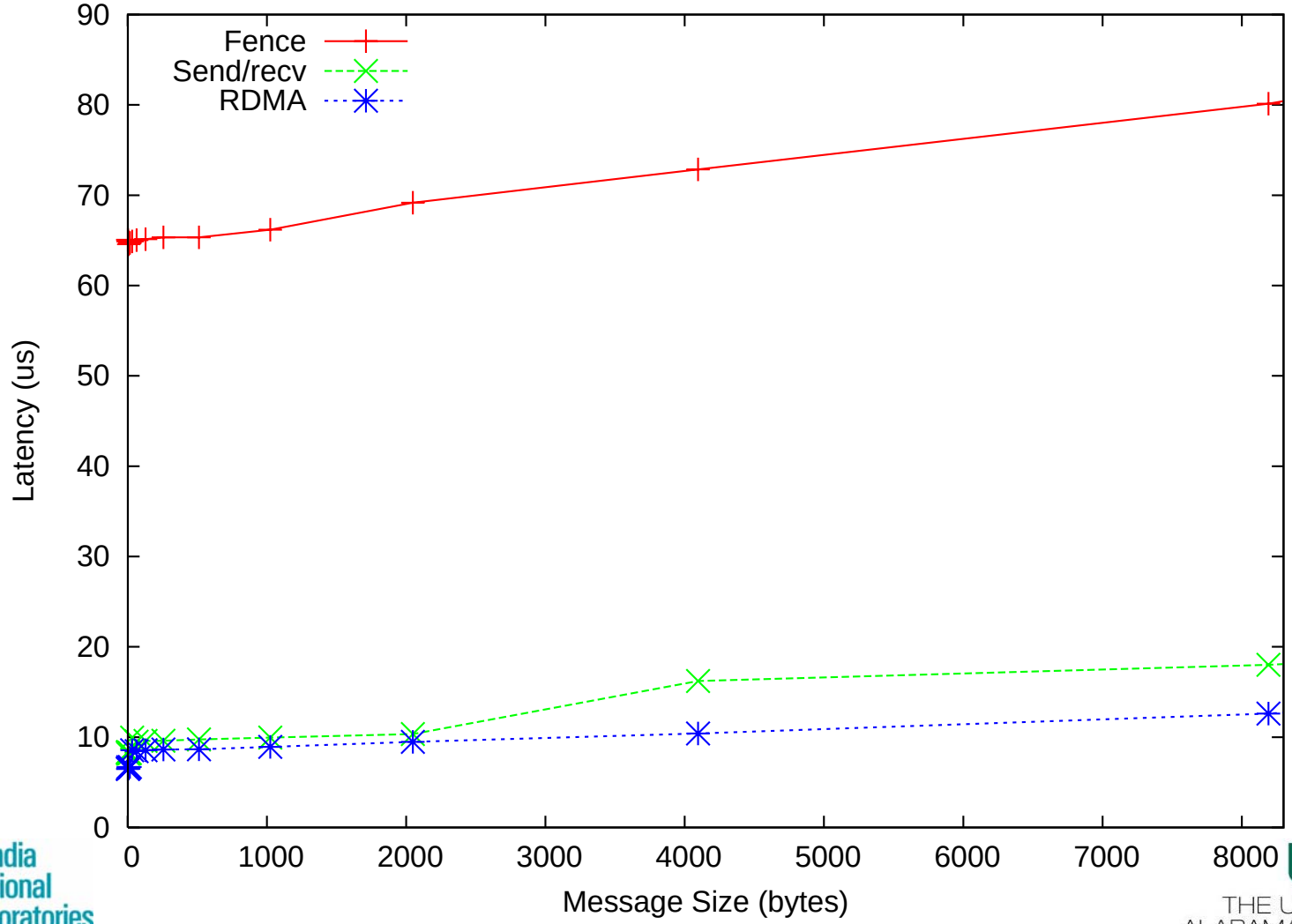
Ping-Pong Latency – 2 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

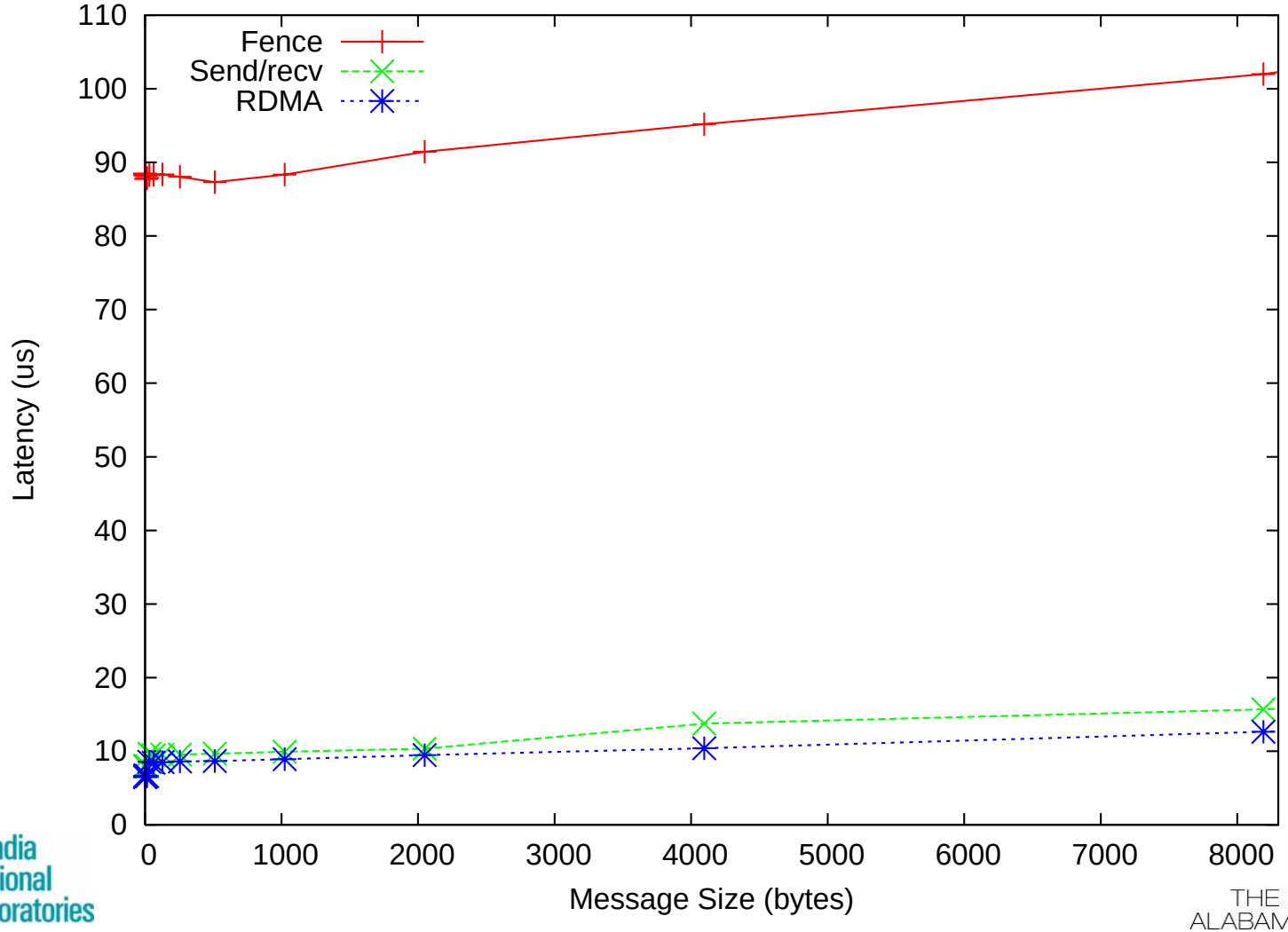
Ping-Pong Latency – 4 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

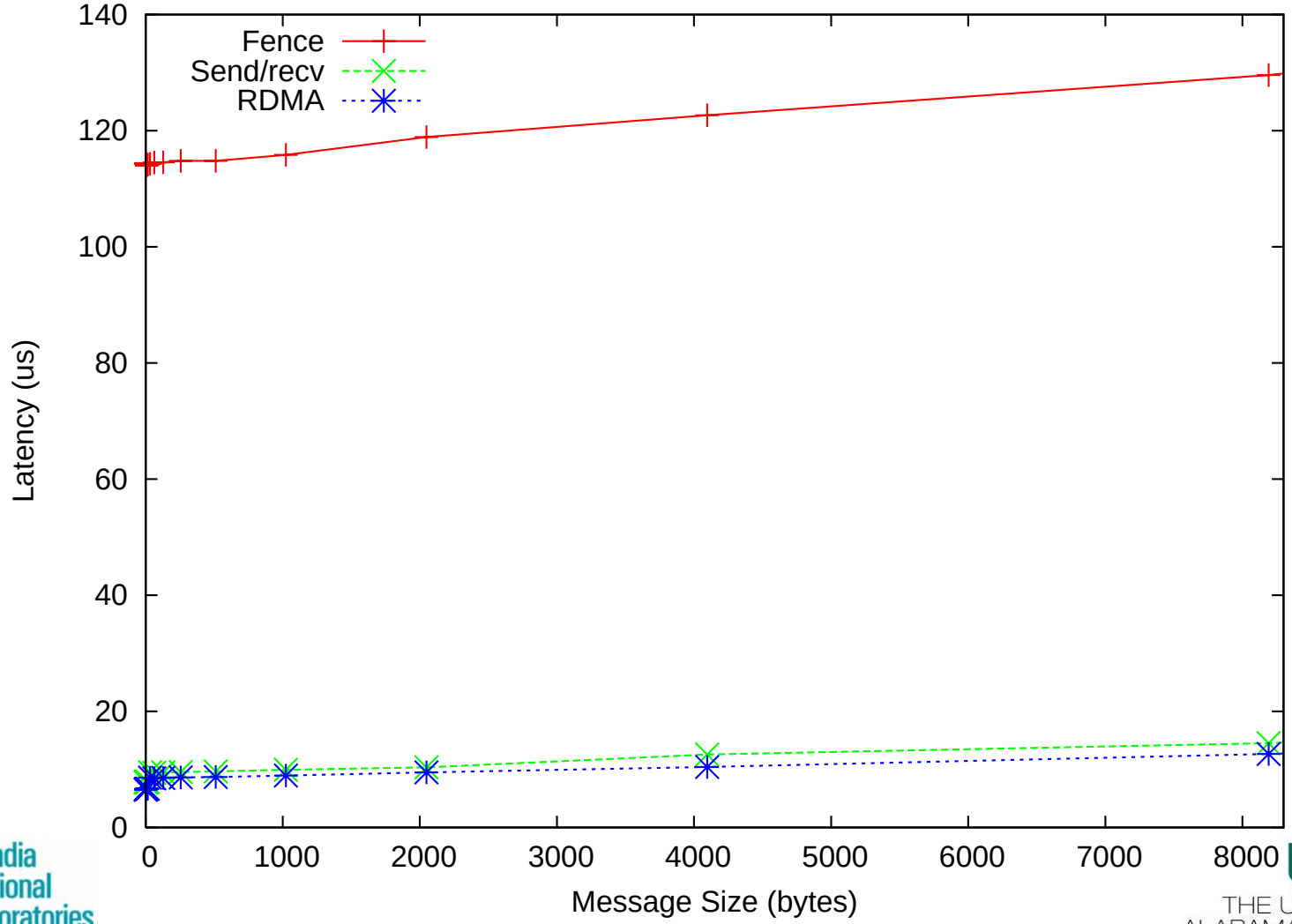
Ping-Pong Latency – 8 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

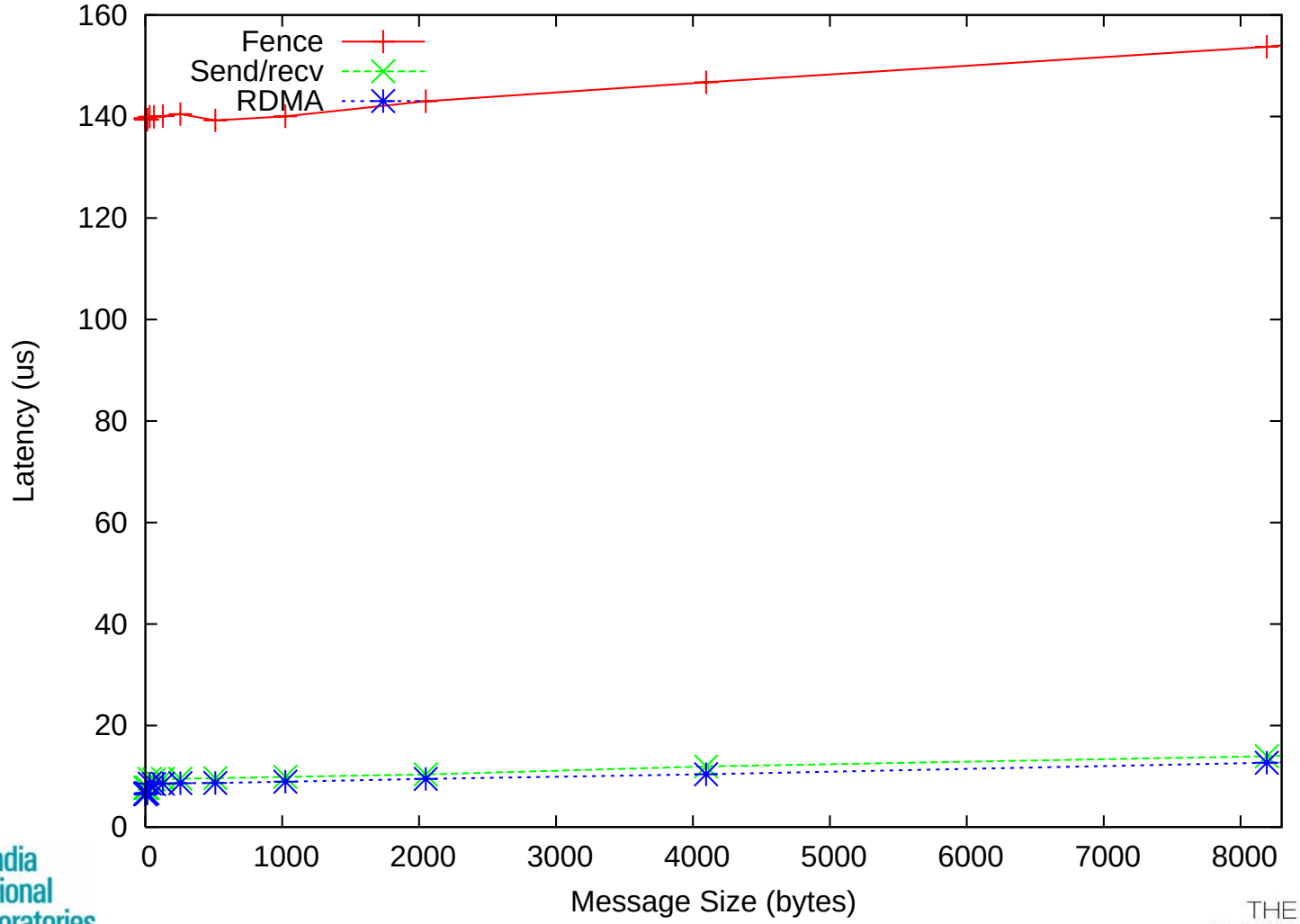
Ping-Pong Latency – 16 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

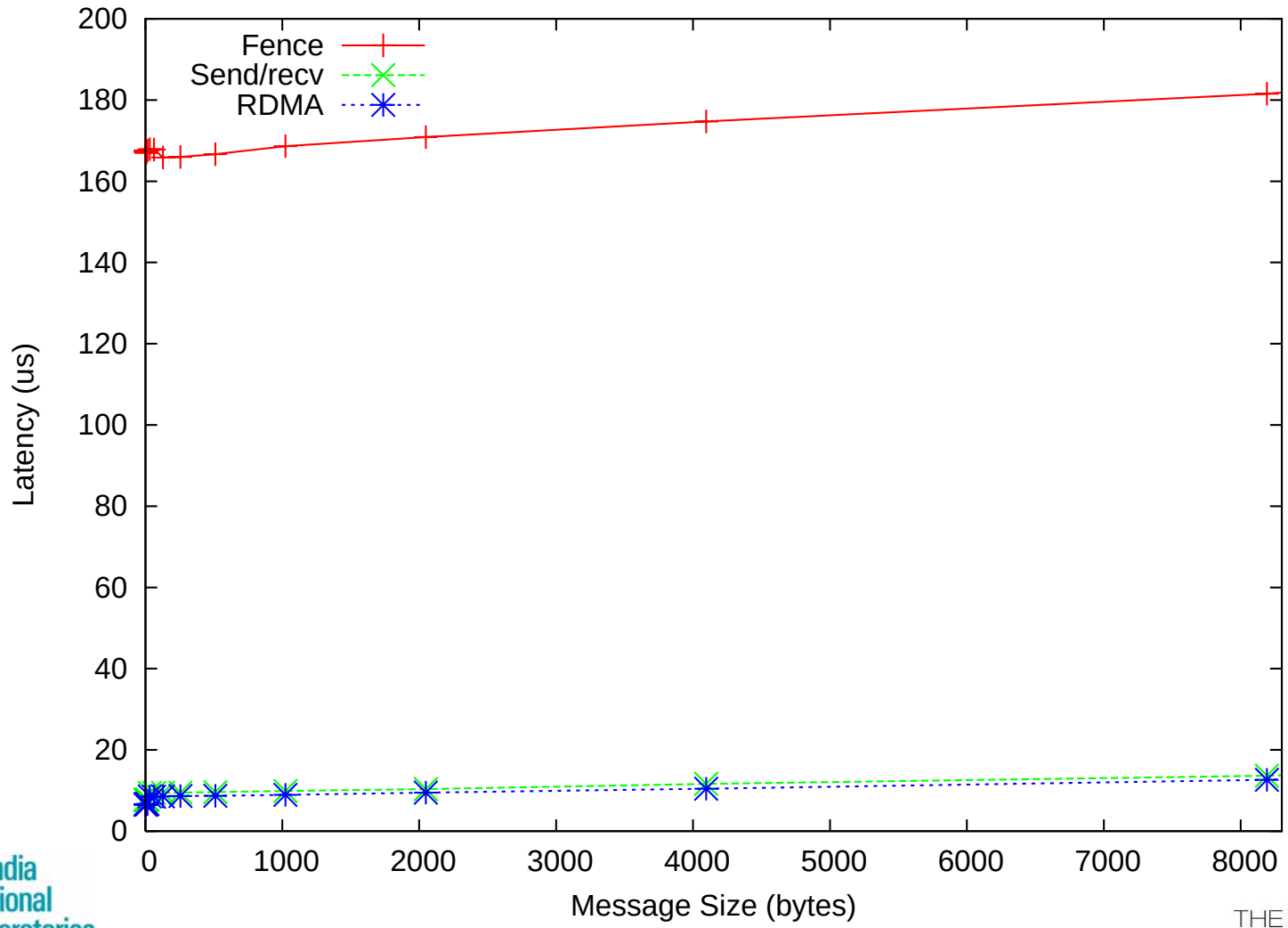
Ping-Pong Latency – 32 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

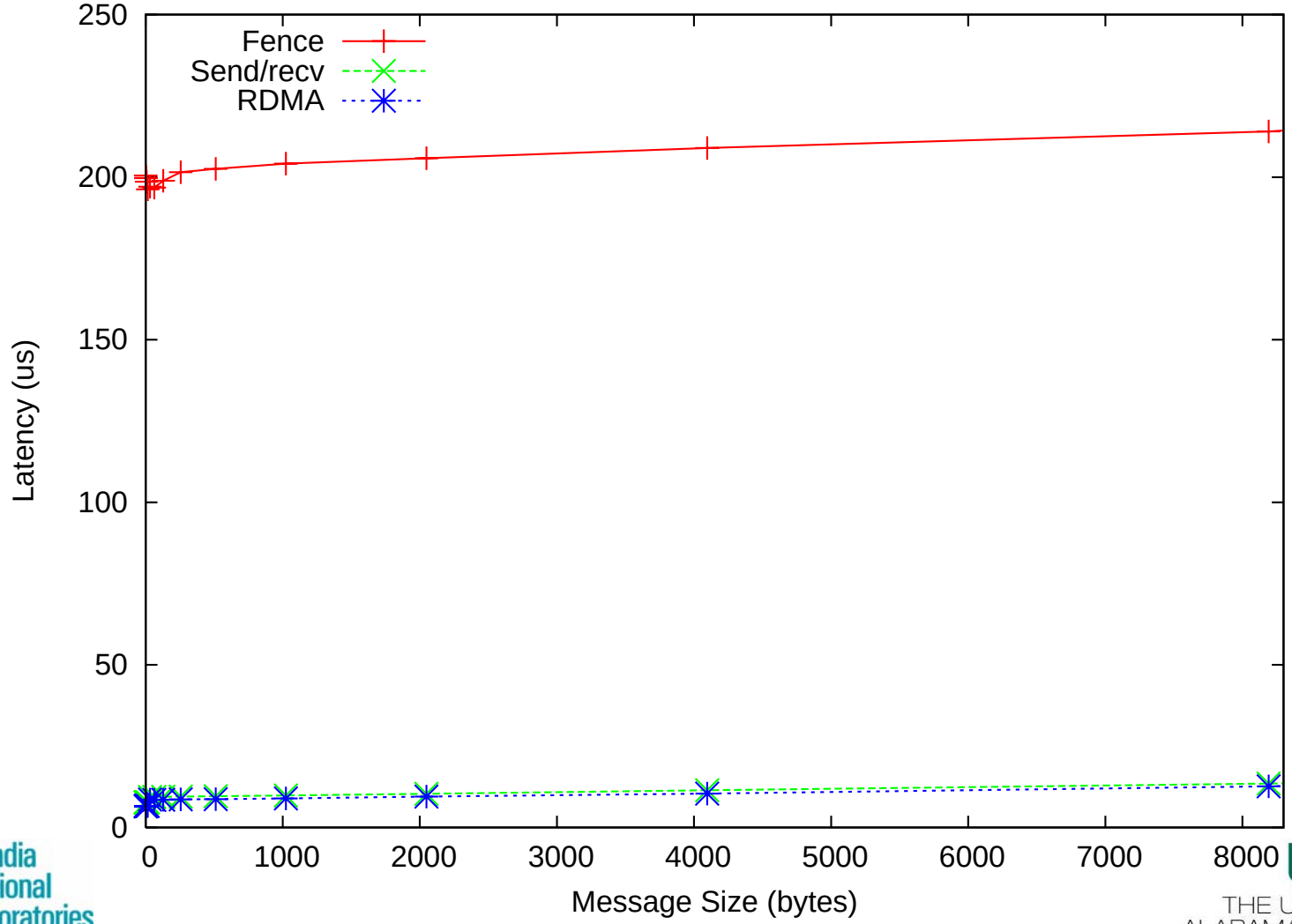
Ping-Pong Latency – 64 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

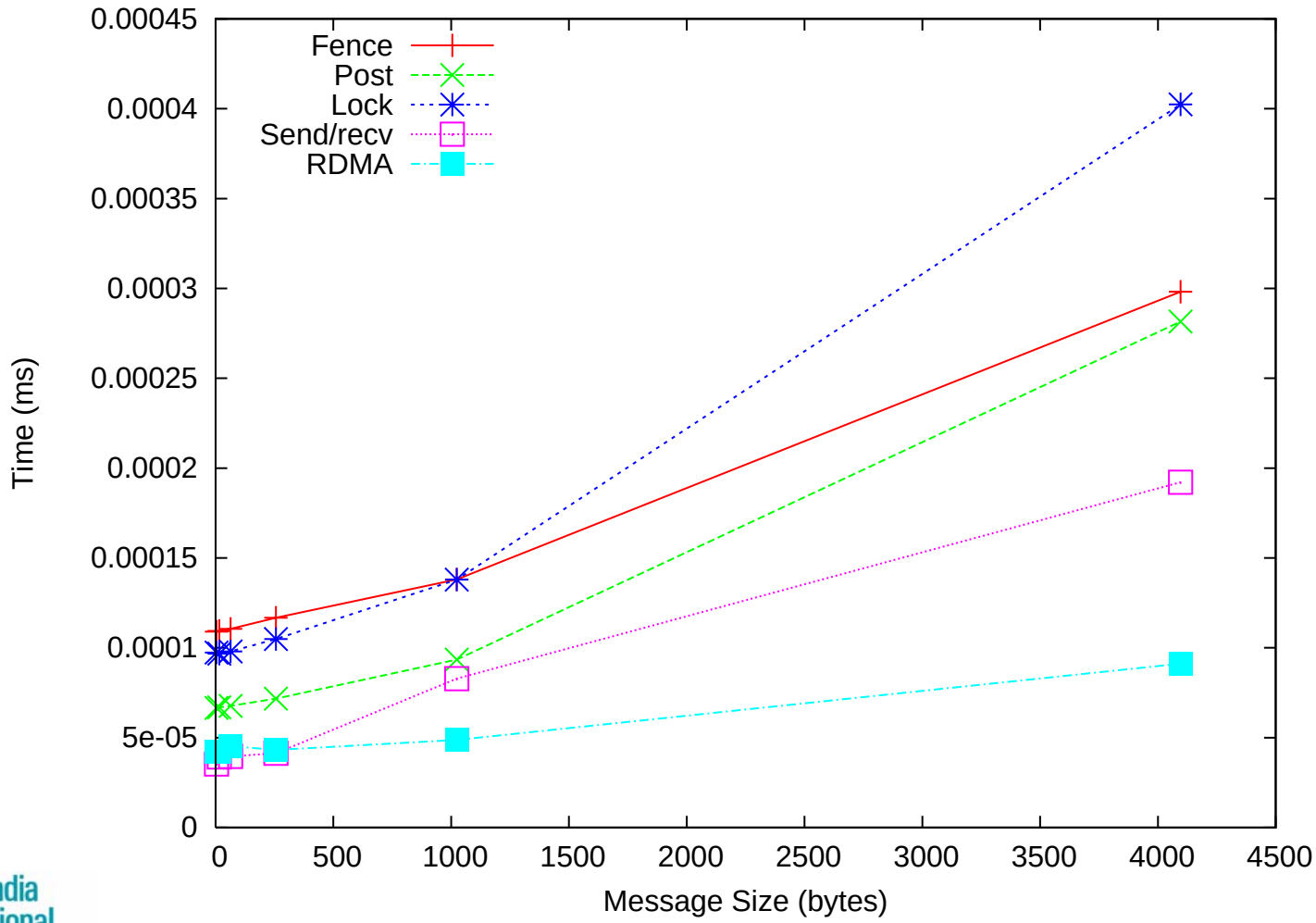
Ping-Pong Latency – 128 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

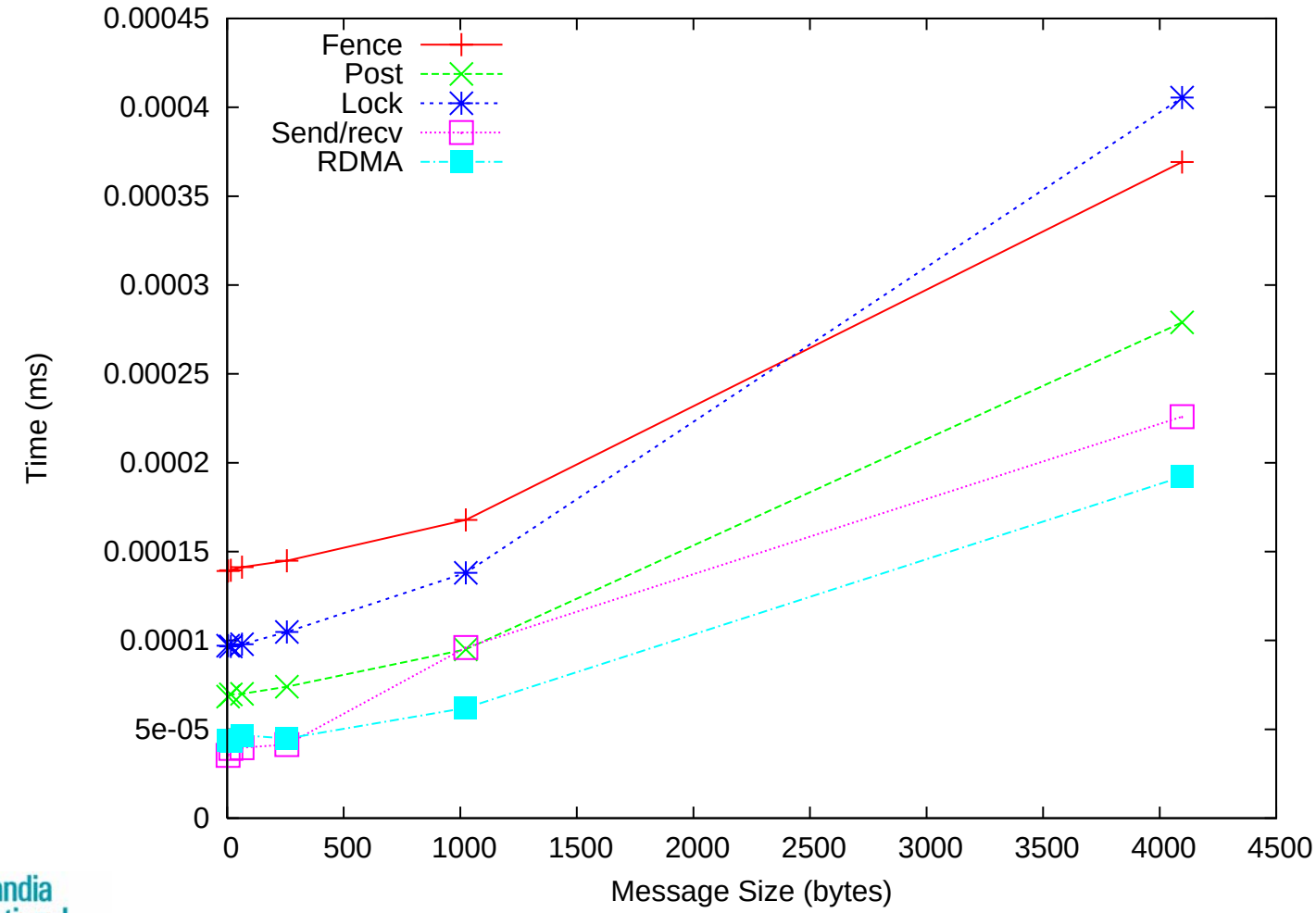
Ghost-Area Exchange – 25 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

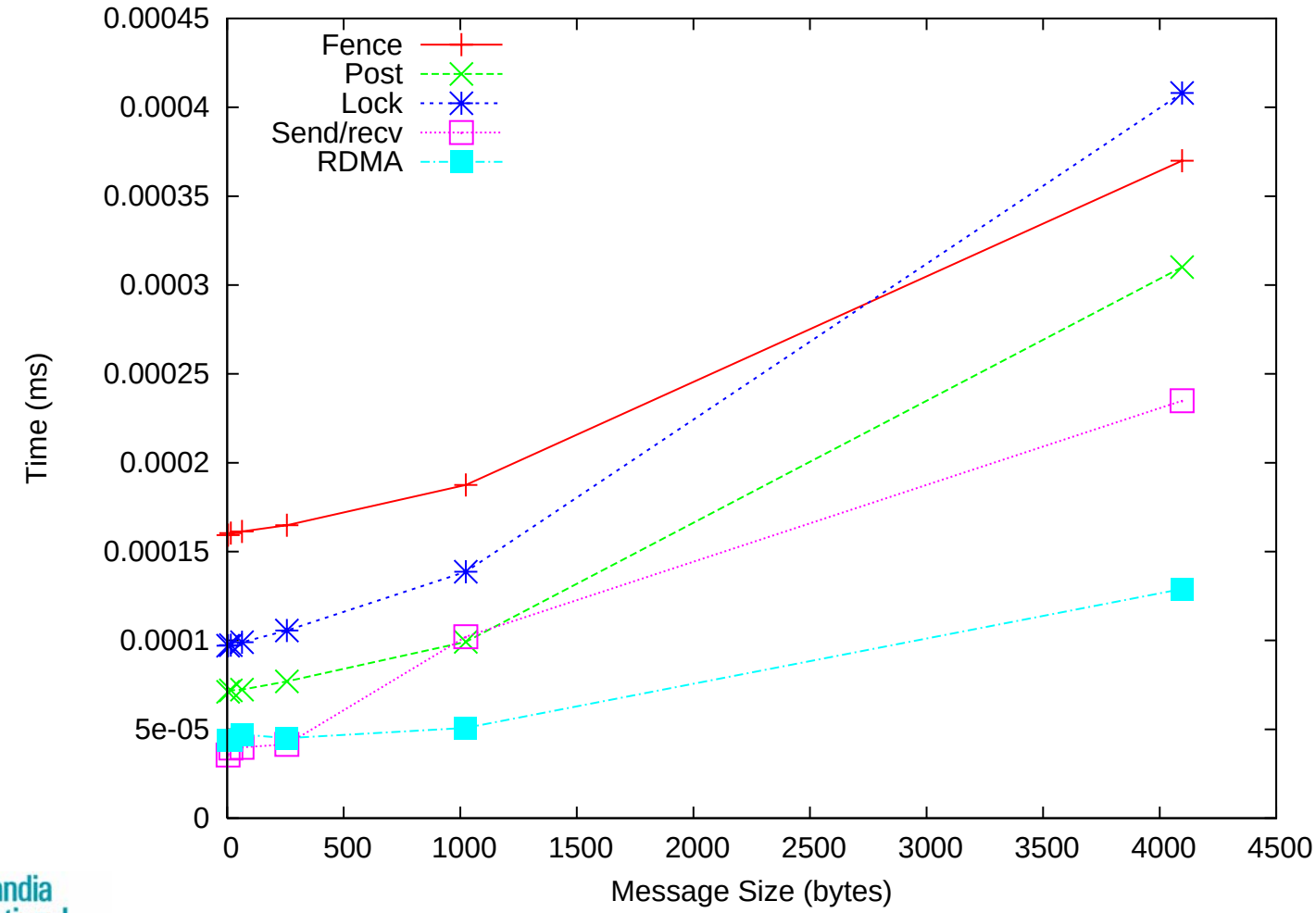
Ghost-Area Exchange – 100 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

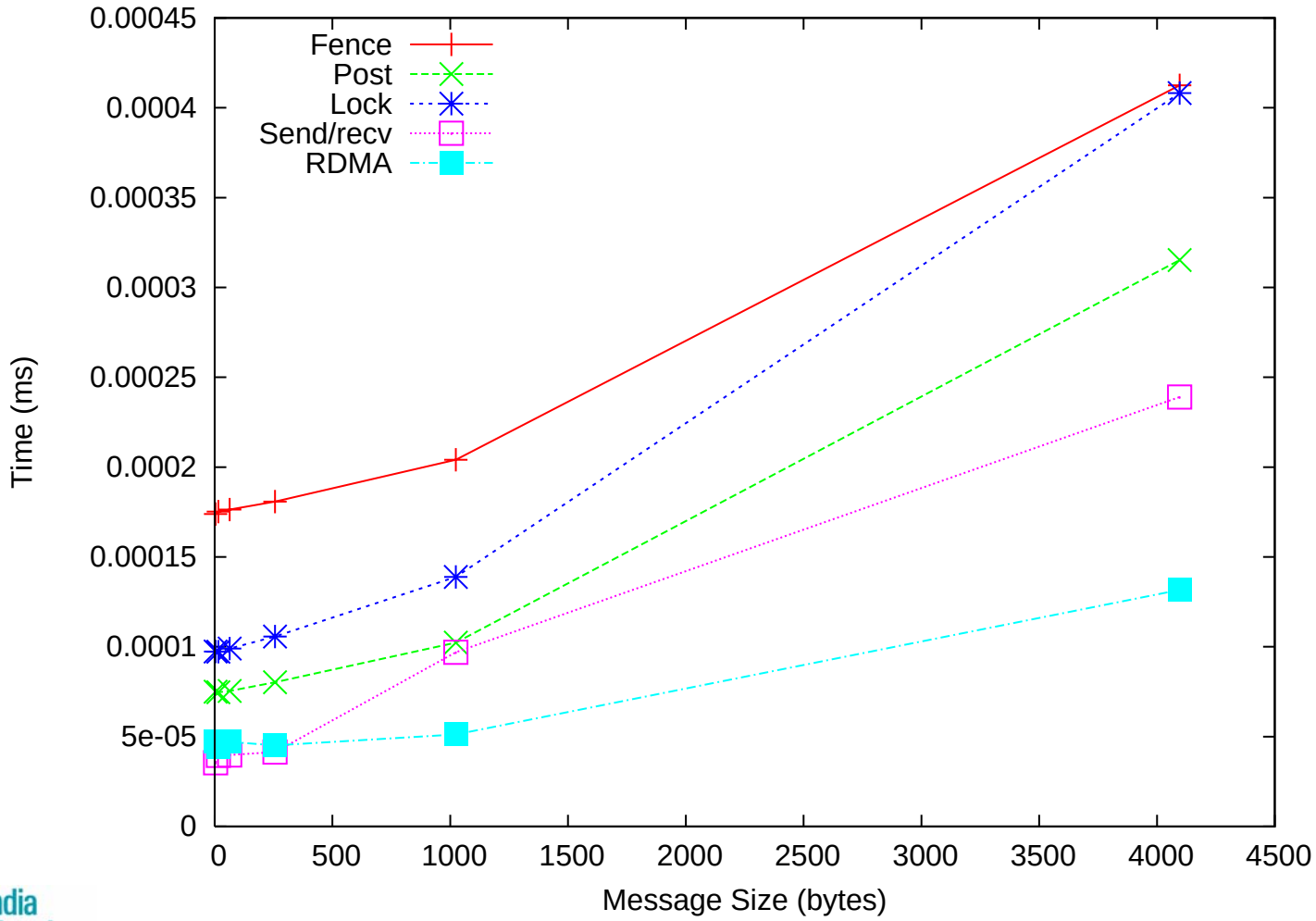
Ghost-Area Exchange – 225 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

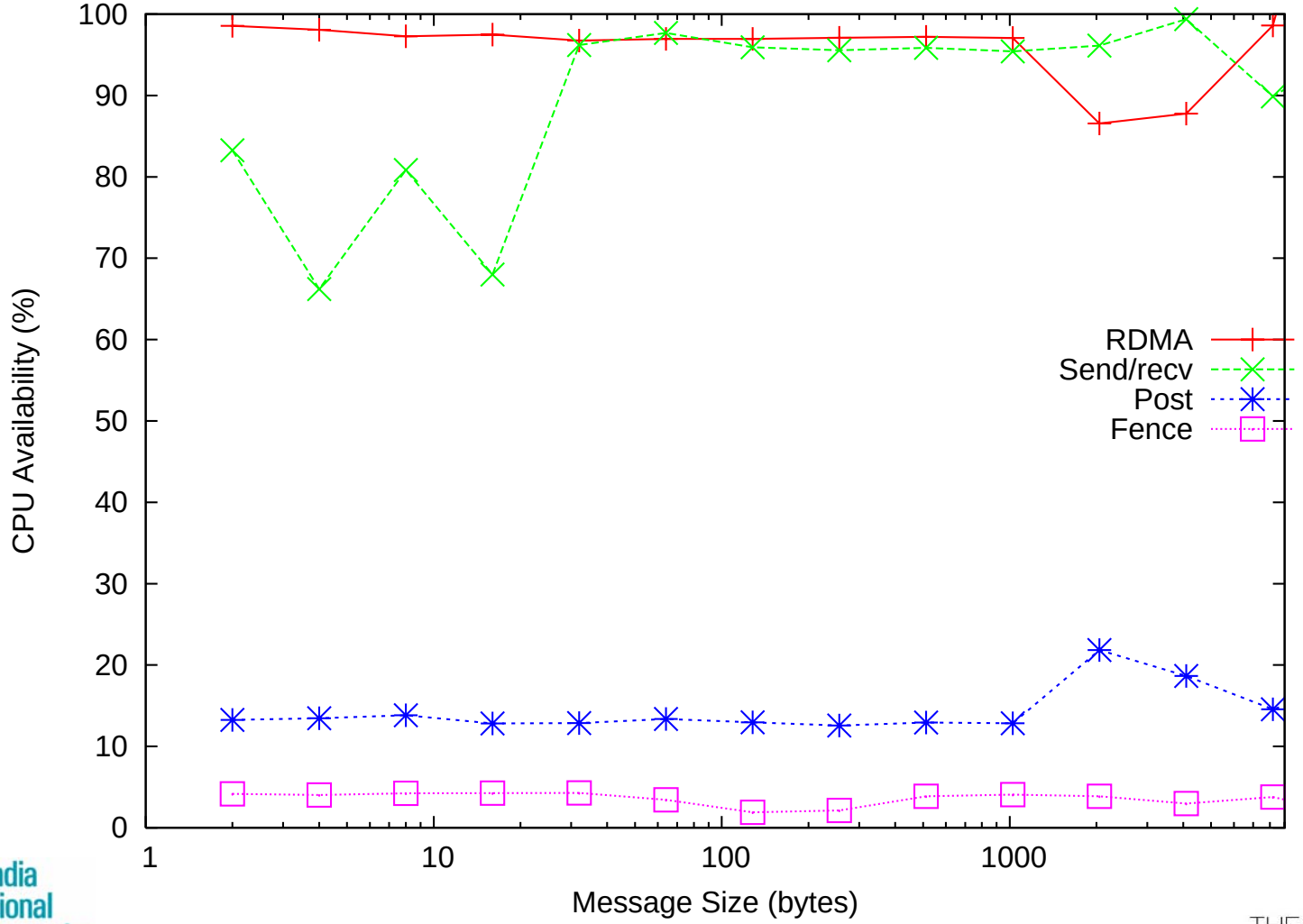
Ghost-Area Exchange – 361 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

CPU Availability



Sandia National Laboratories

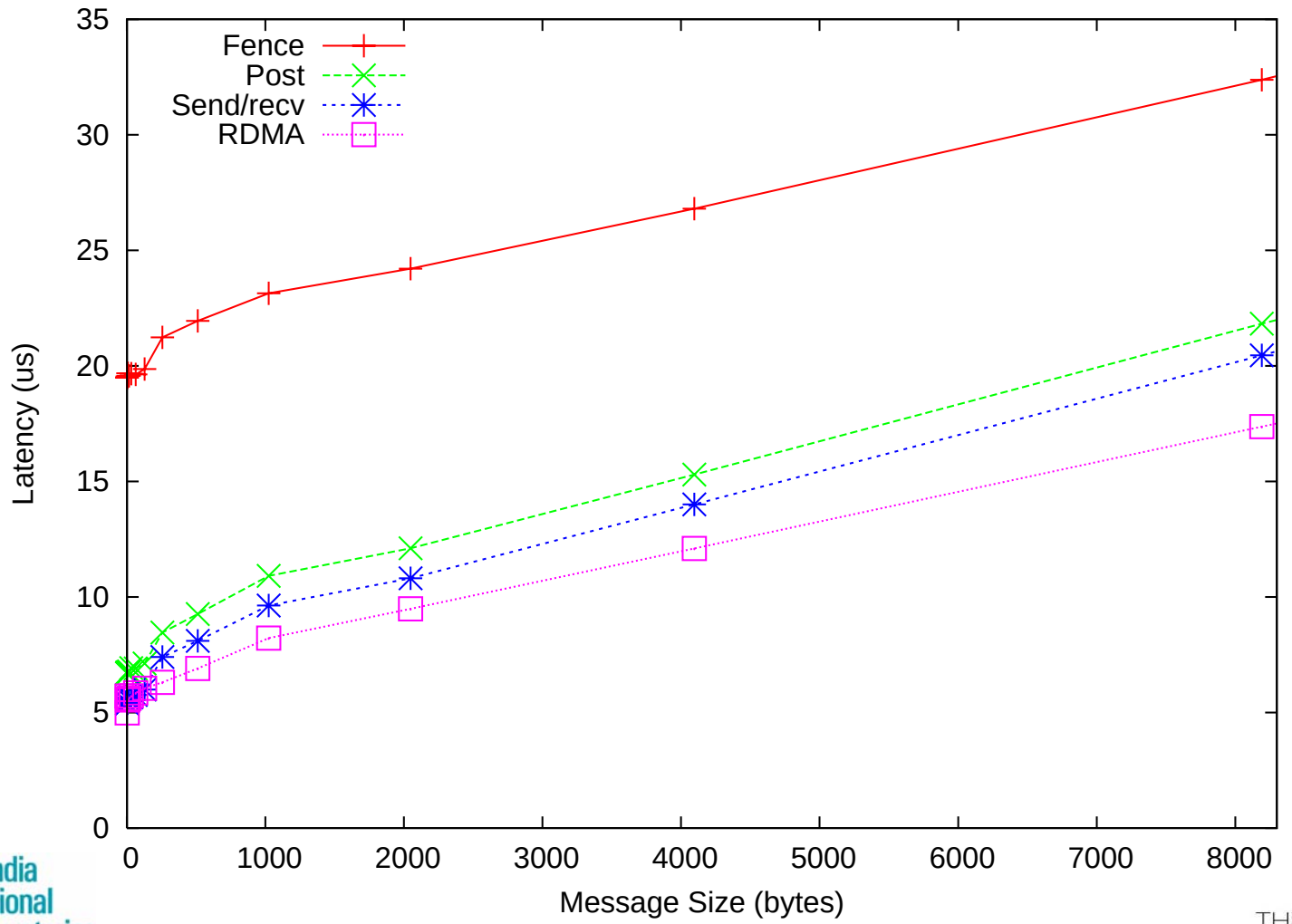
THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

InfiniBand Cluster Results

Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

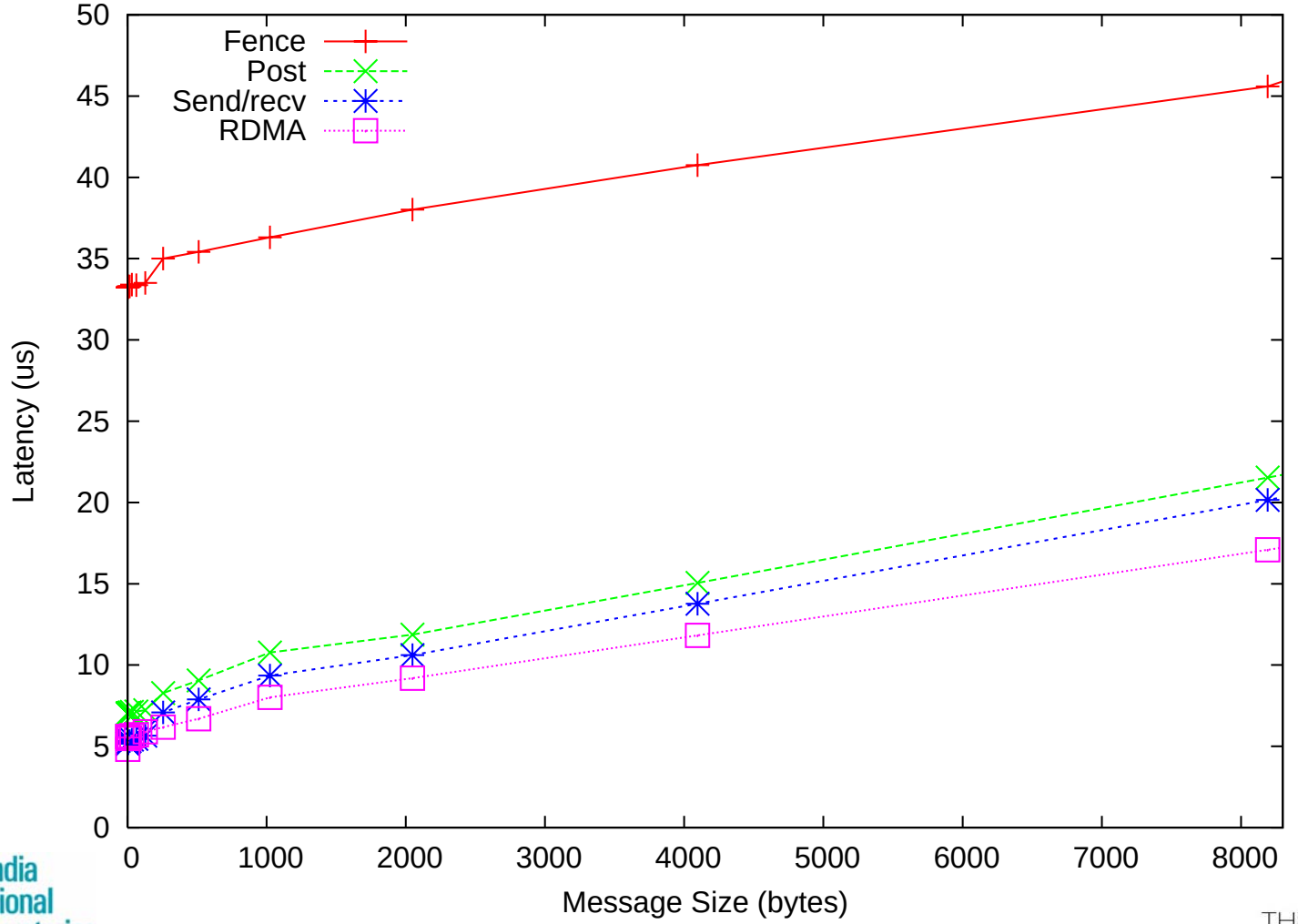
Ping-Pong Latency – 2 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

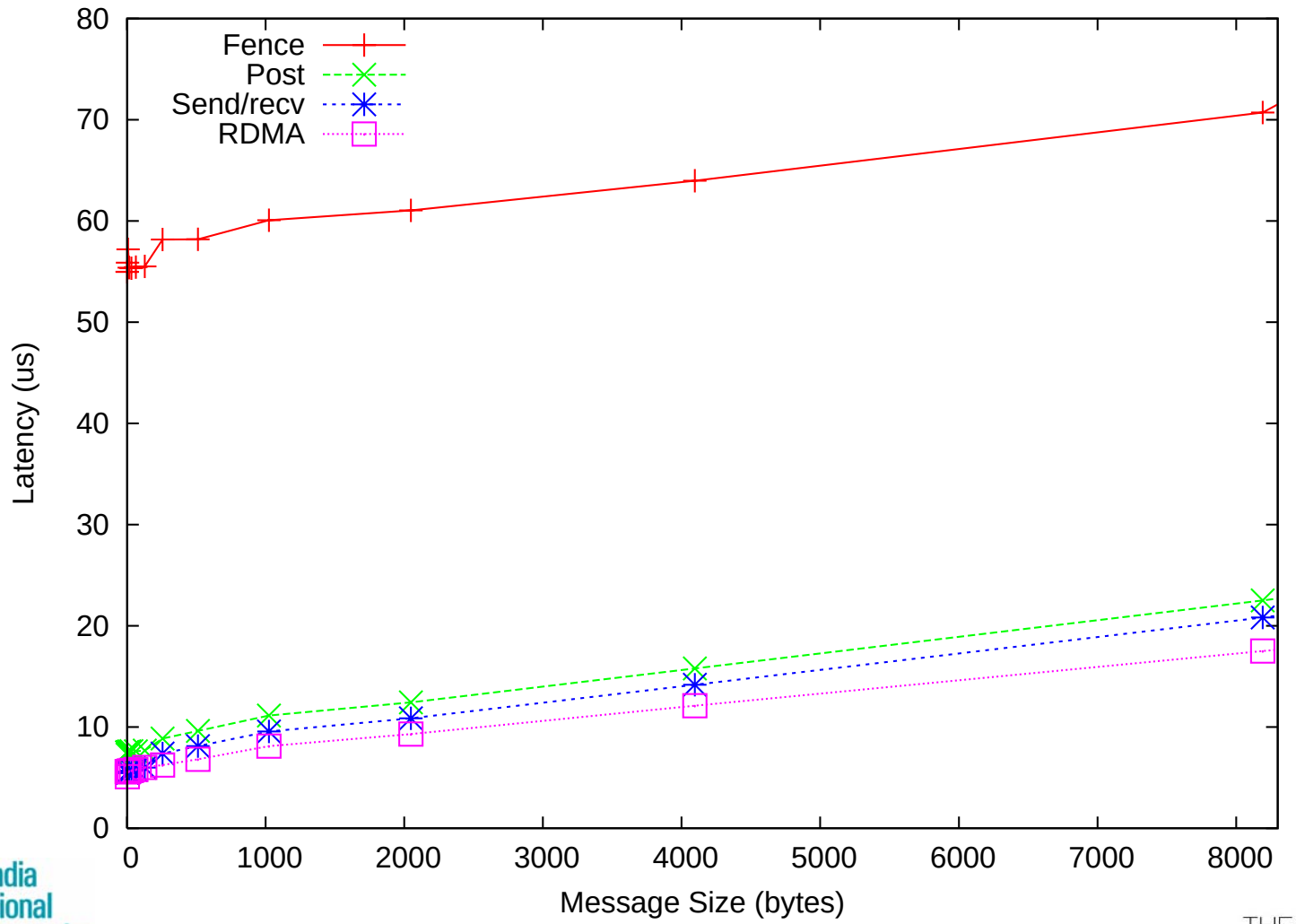
Ping-Pong Latency – 4 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

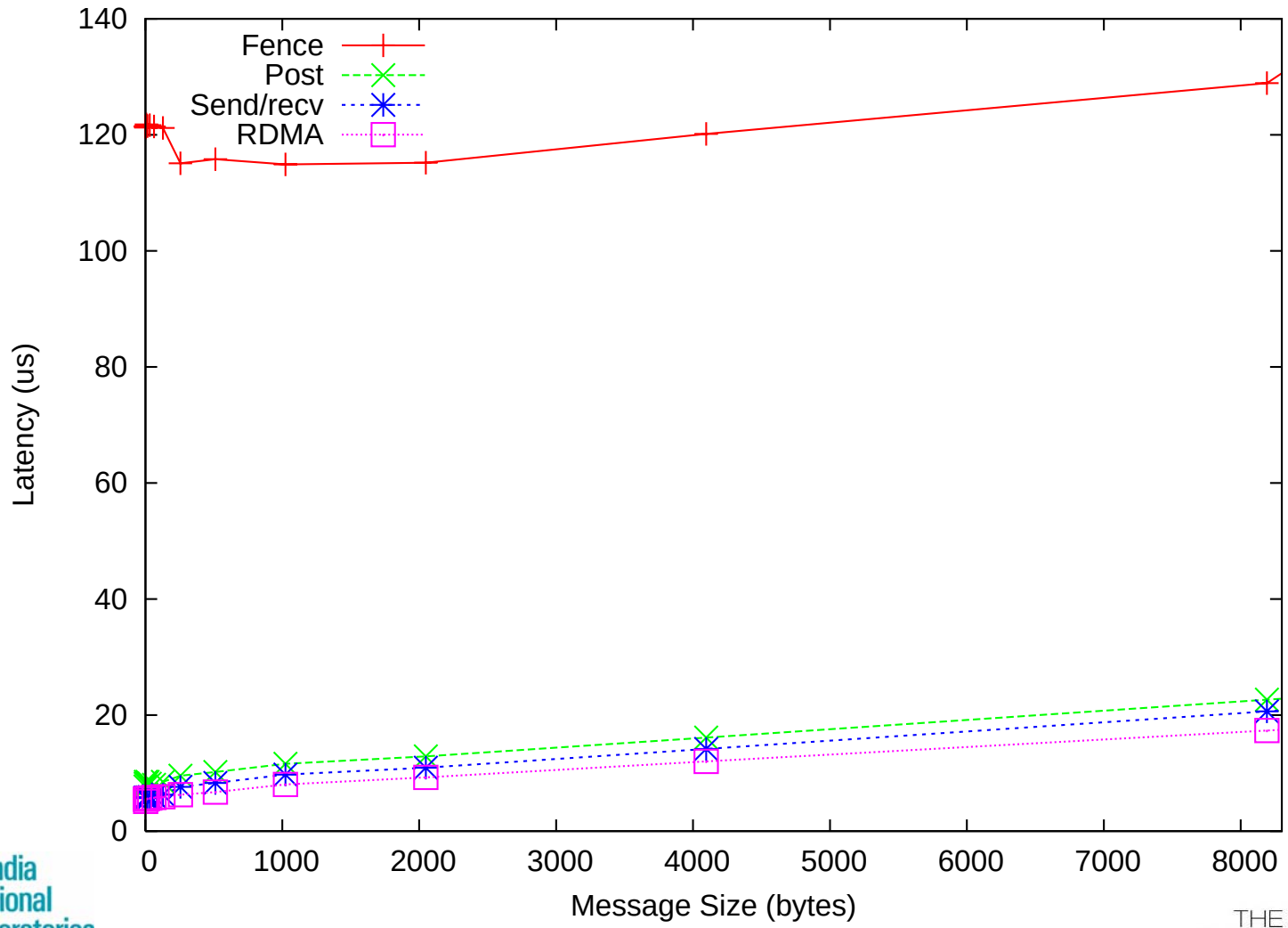
Ping-Pong Latency – 8 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

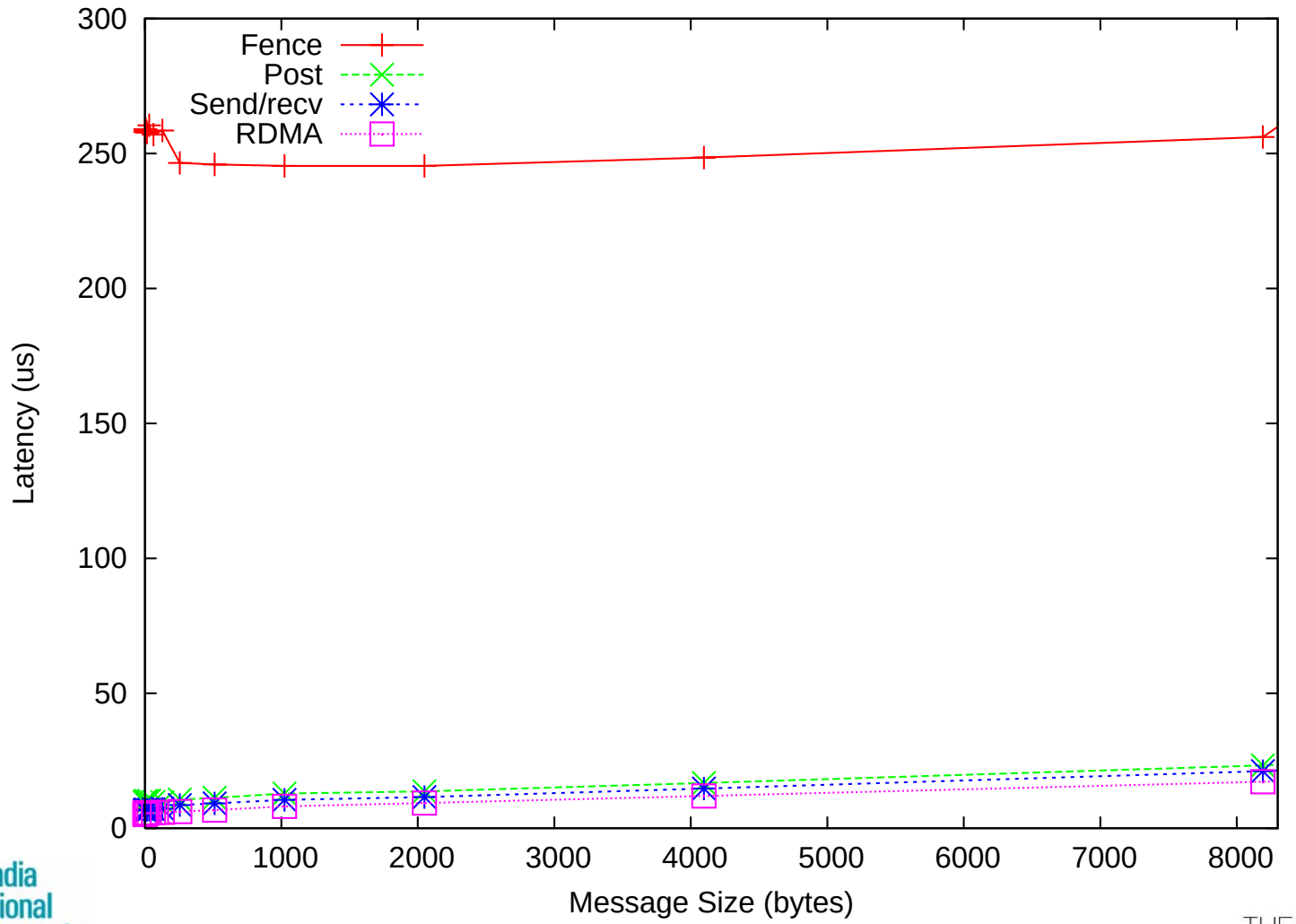
Ping-Pong Latency – 16 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

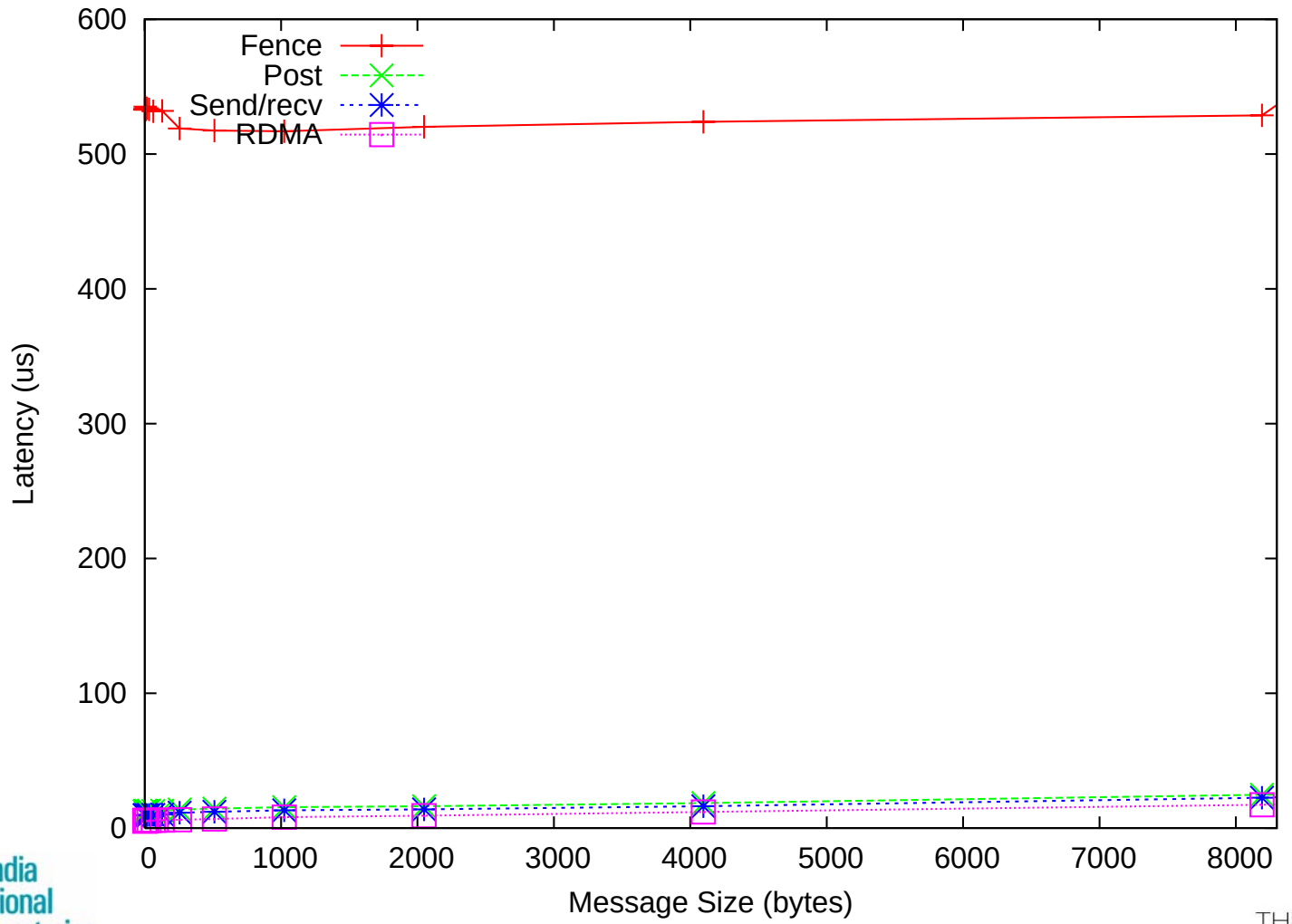
Ping-Pong Latency – 32 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

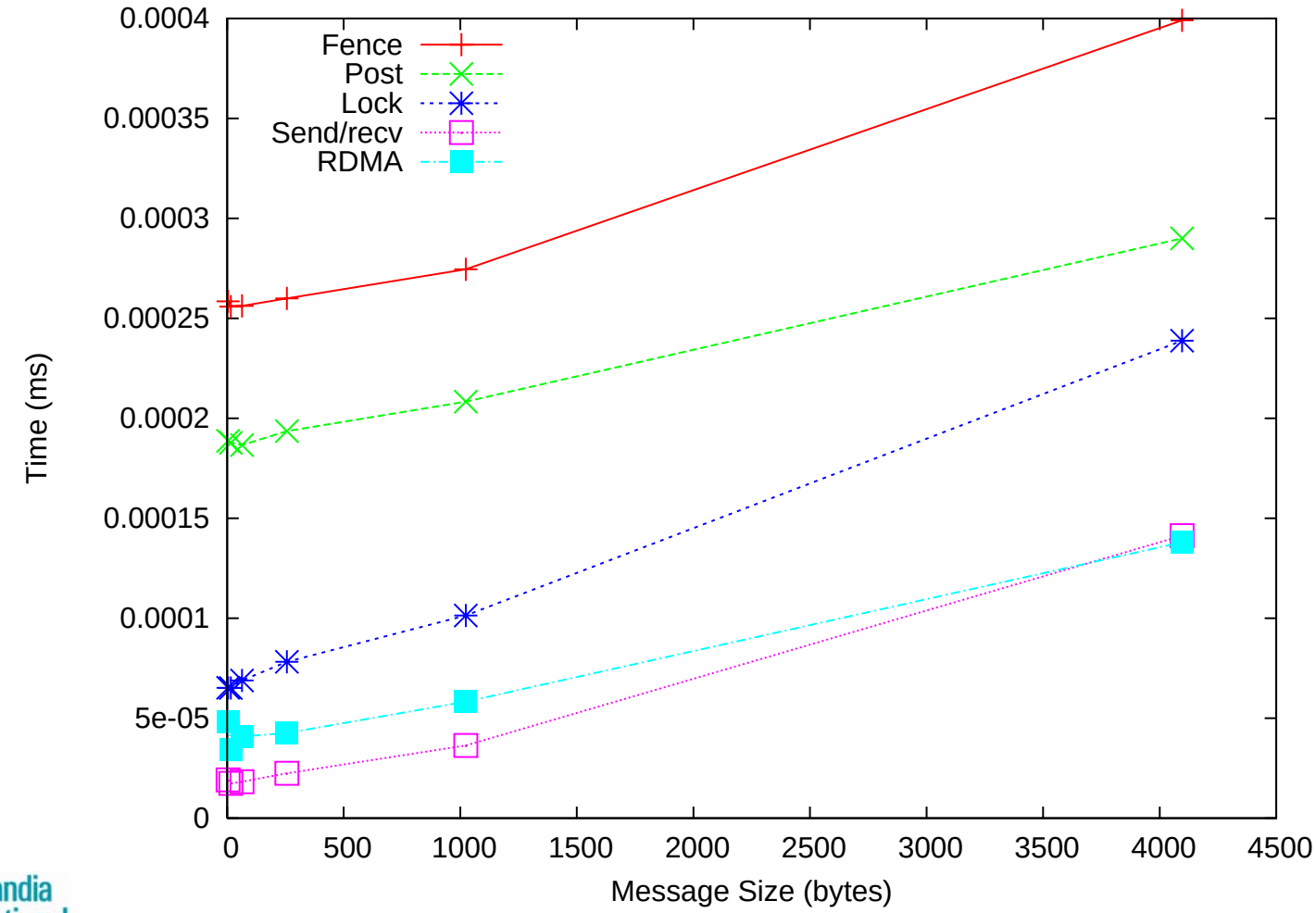
Ping-Pong Latency – 64 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

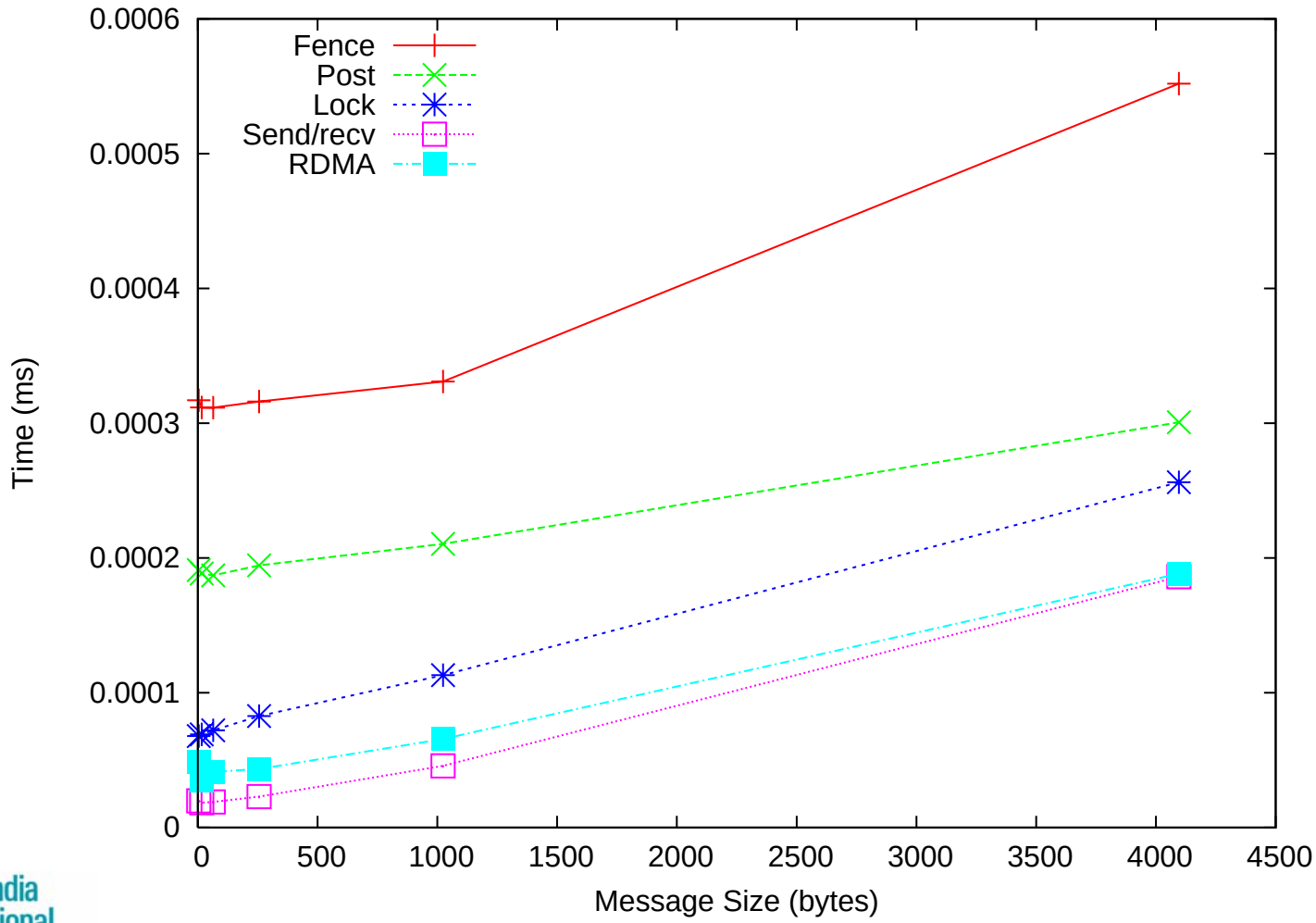
Ghost-Area Exchange – 9 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

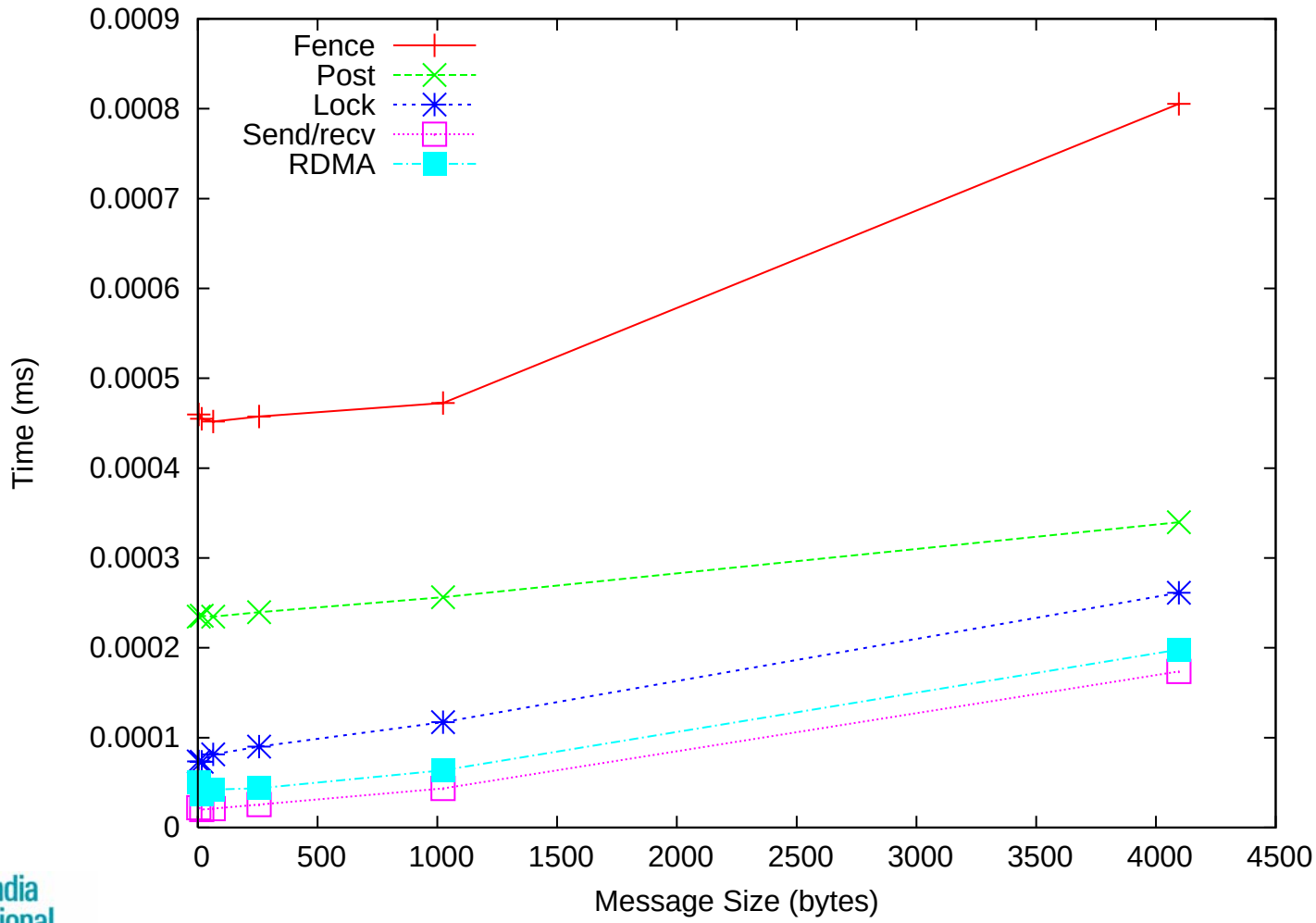
Ghost-Area Exchange – 16 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

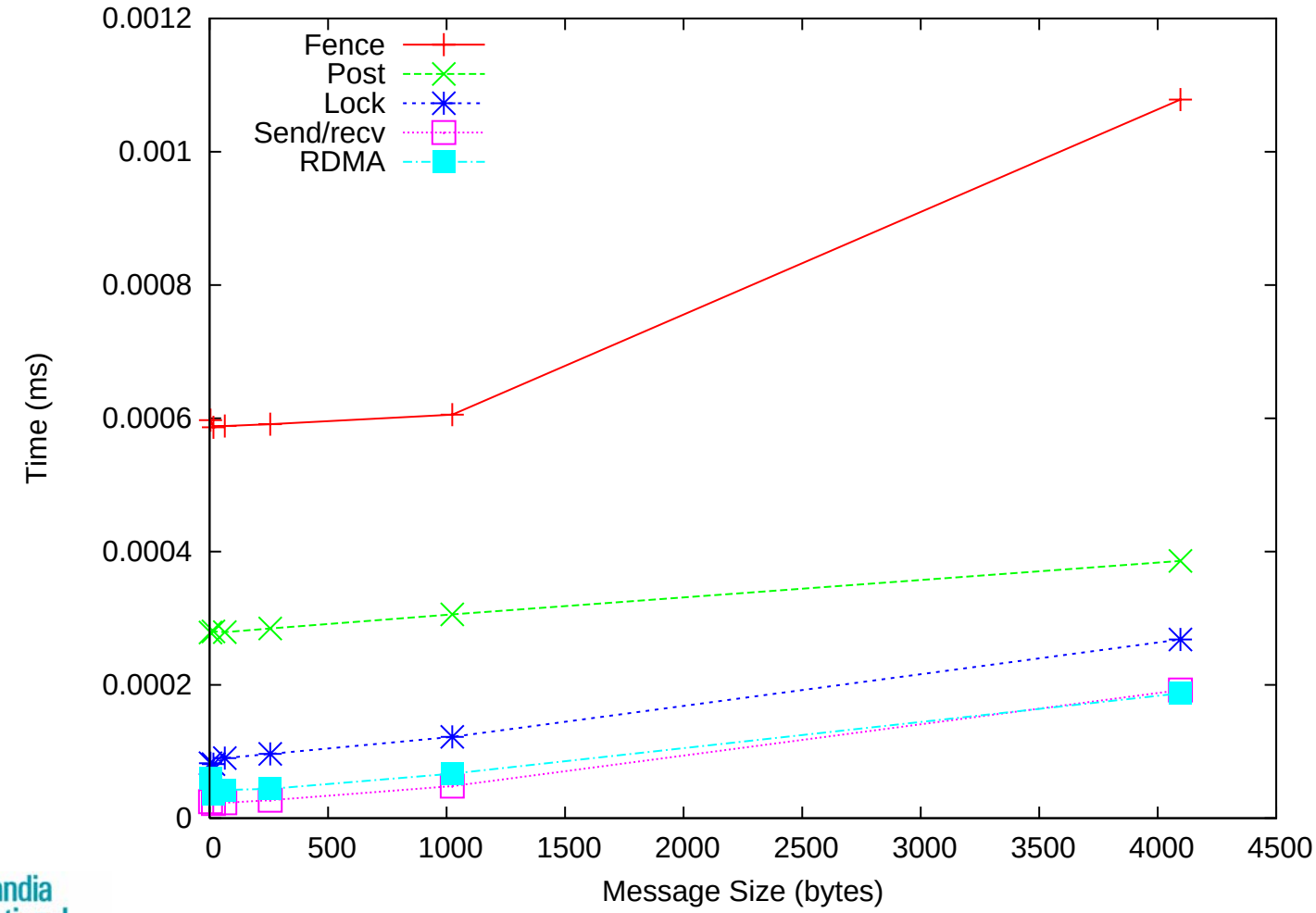
Ghost-Area Exchange – 25 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

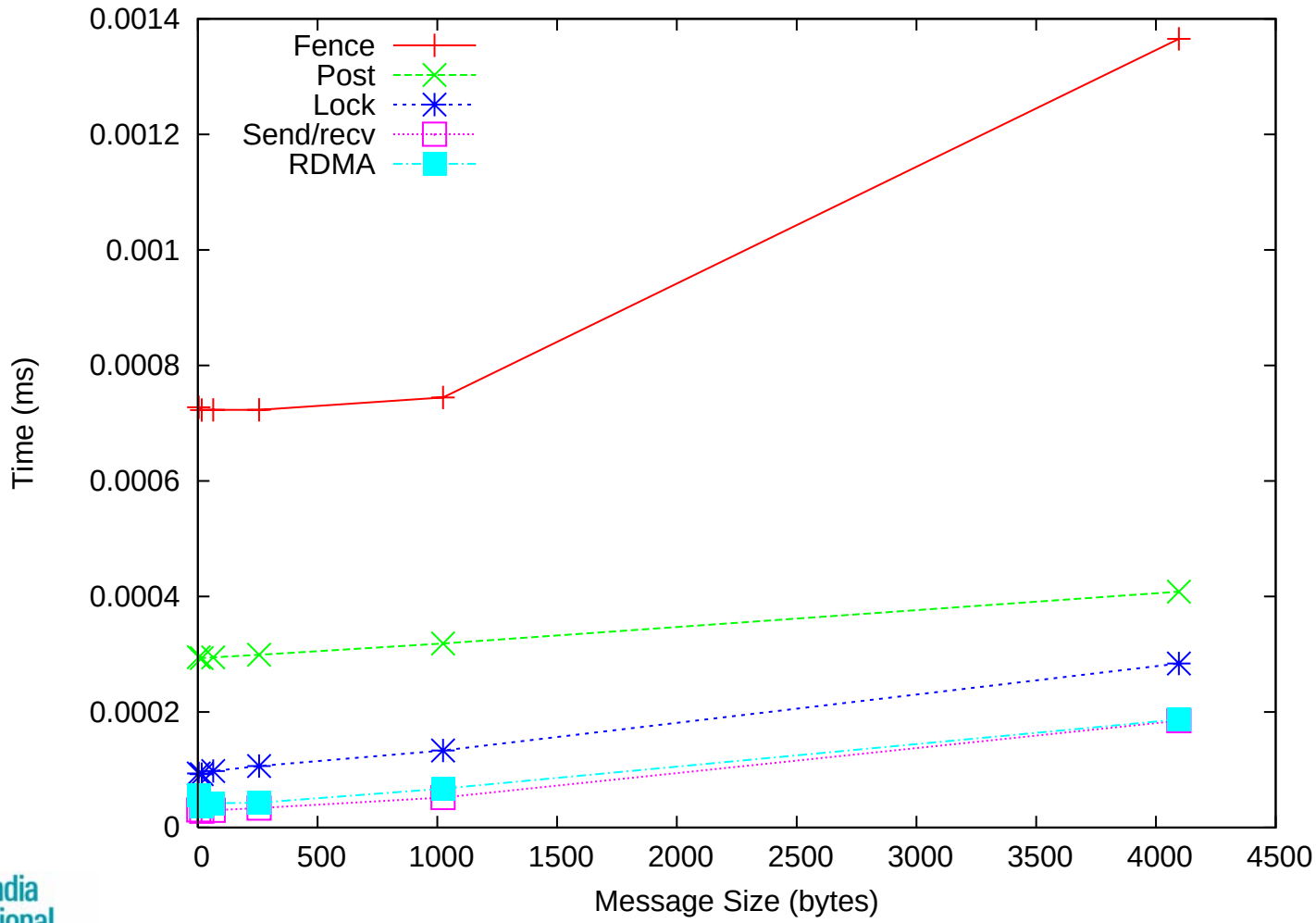
Ghost-Area Exchange – 36 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

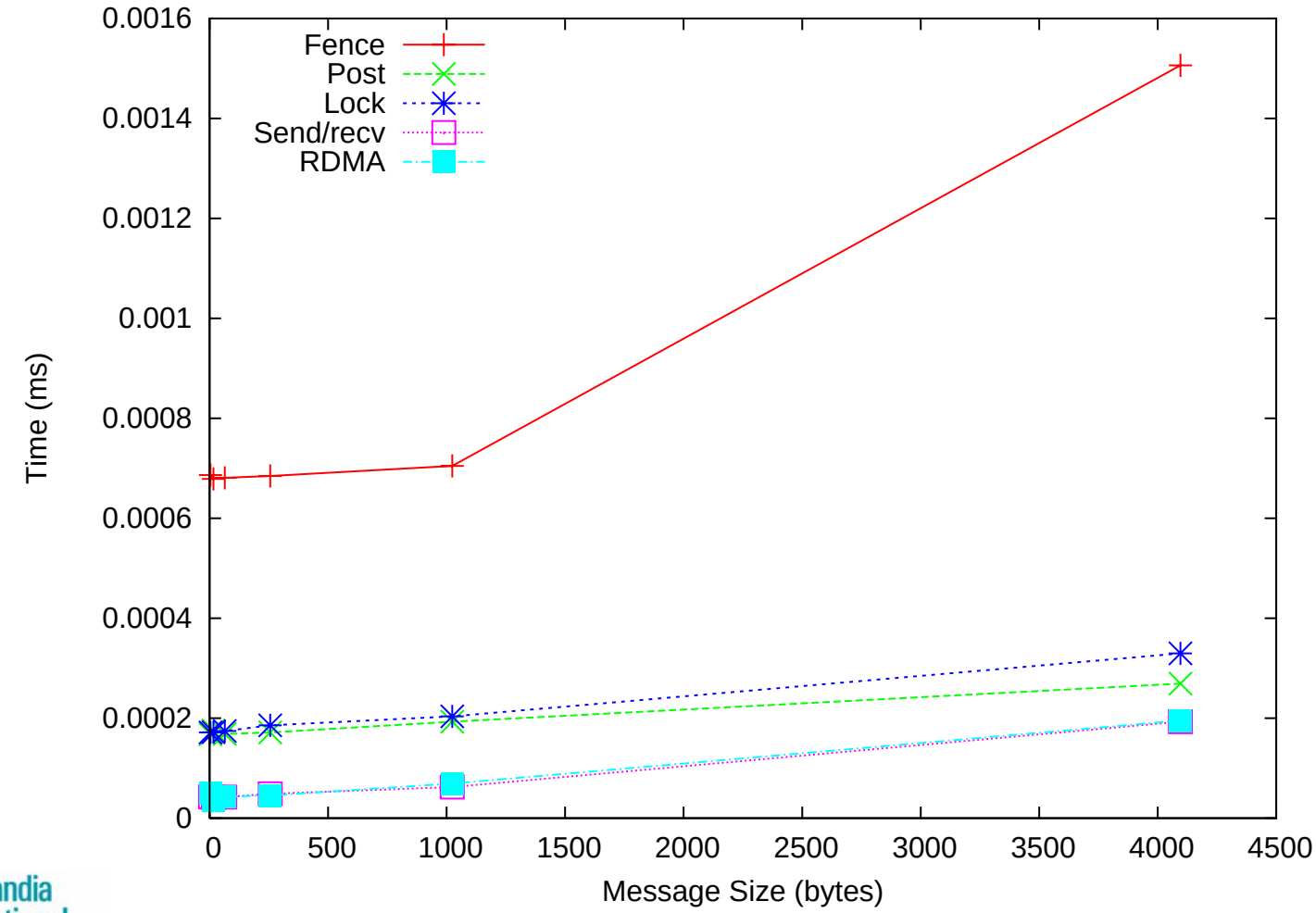
Ghost-Area Exchange – 49 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

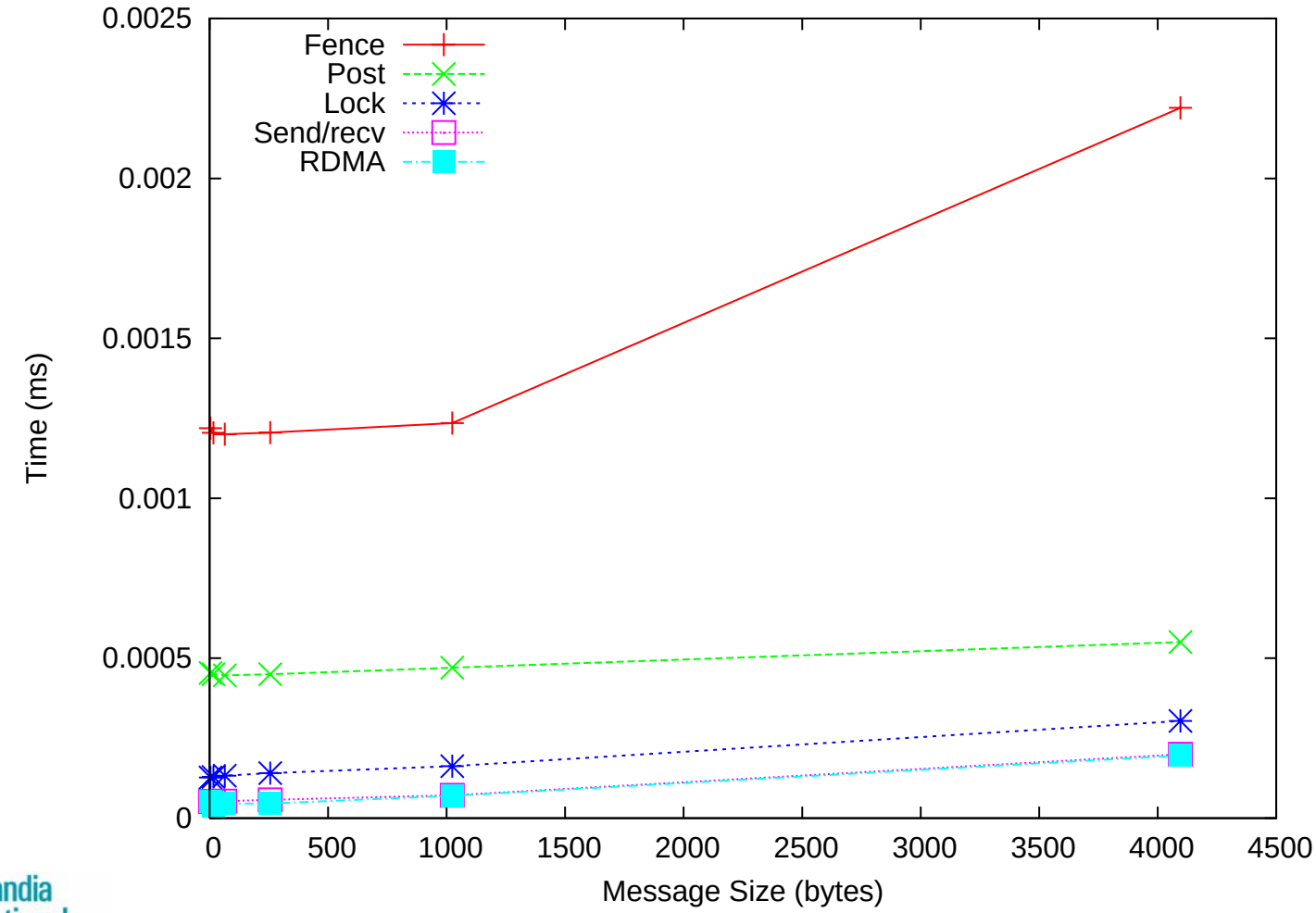
Ghost-Area Exchange – 64 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

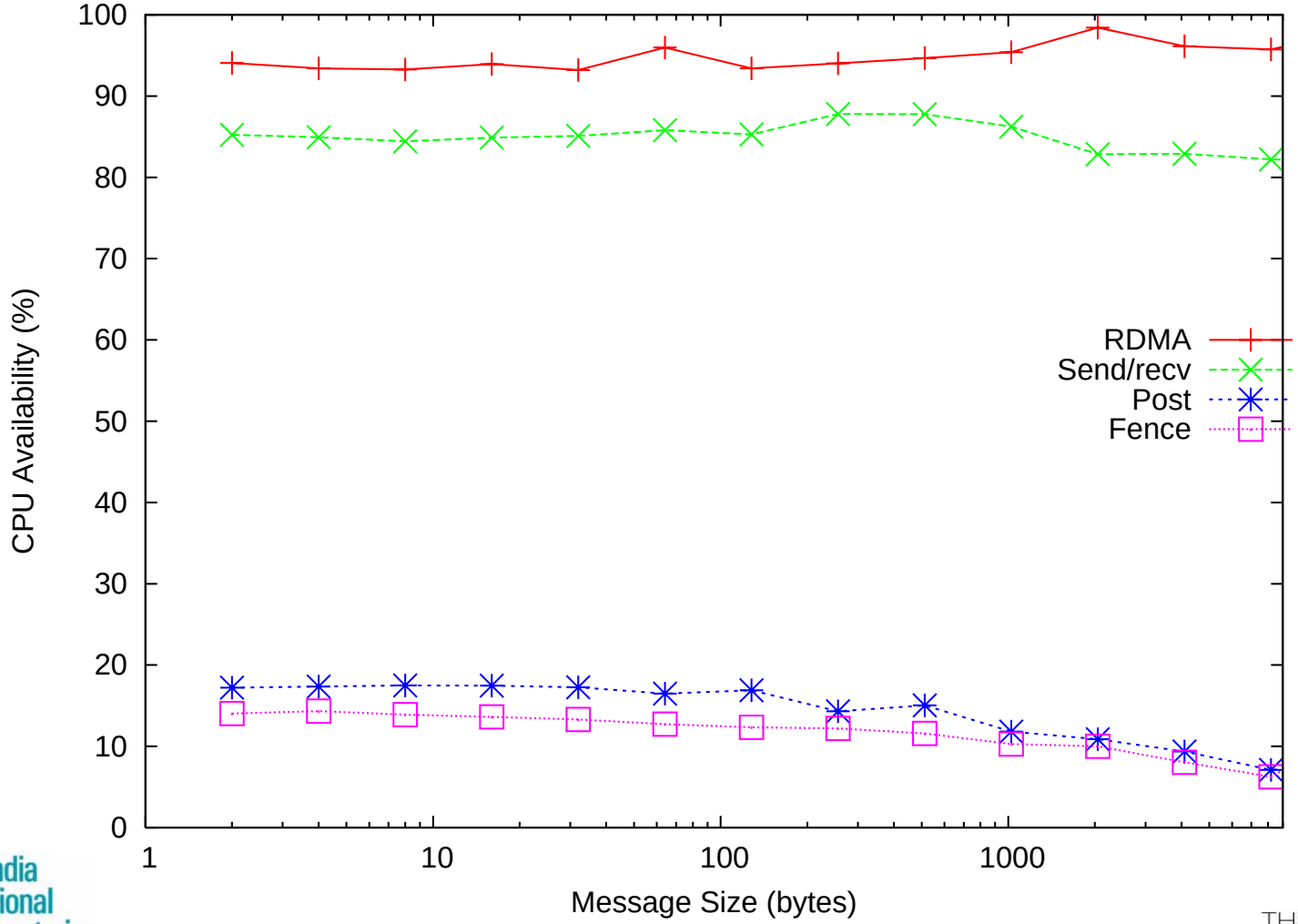
Ghost-Area Exchange – 81 Nodes



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

CPU Availability



Sandia National Laboratories

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

Conclusion

- We have designed and implemented an MPI extension to better support RDMA operations
- This extension allows for local completion of all RDMA operations, avoiding the synchronization penalty of MPI one-sided
- Implementations of this RDMA extension outperform MPI point-to-point and MPI one-sided operations on two different RDMA capable networks on several benchmarks
- Our results also demonstrate the performance problems of MPI one-sided versus point-to-point

Future Work

- Implementations for more RDMA-capable networks
- Evaluate using more and better one-sided micro-benchmarks
  - MPI one-sided benchmarks from Iowa State
  - SKaMPI 5.0
- Evaluate using real applications