Introduction to Parallel Processing

Lecture 19 : Advanced Performance Models

10/26/2022

Professor Amanda Bienz

Message Passing

- To communicate a message, all data is split up into packets and the packets are sent through the network to the destination process
- Also, have an envelope that describes the message (size, tag, etc)
- Different protocols for sending messages:

Message Passing

- **Short** : All message data fits in envelope, sent directly to process

Message Passing

- **Eager :** Message does not fit in envelope, but still relatively small
  - Can assume the receiving process has buffer space available for this message
  - Pack up and send directly

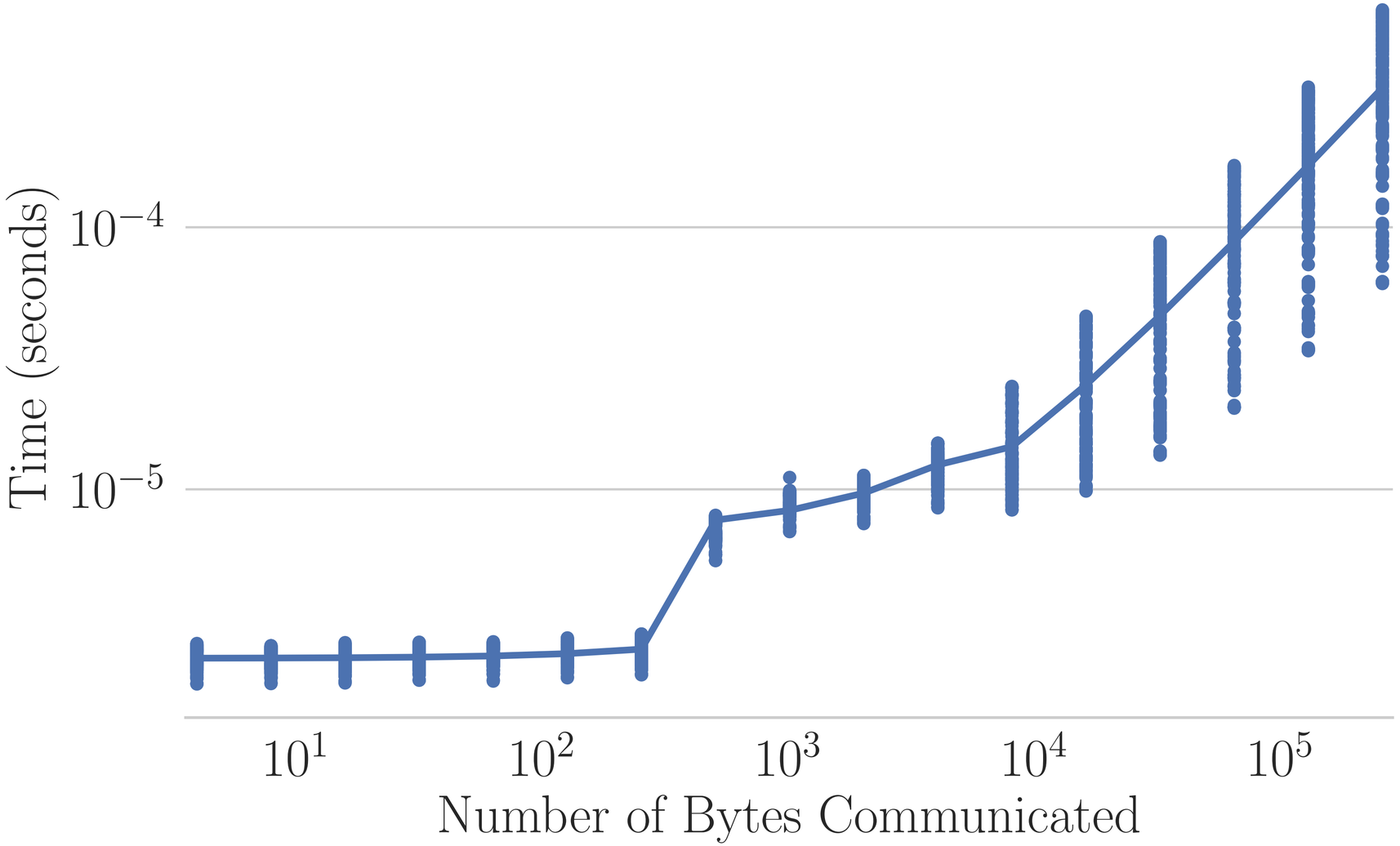
Message Passing

- **Rendezvous :** Largest messages
  - Cannot assume receiving process has buffer space for this message
  - Sending process sends a message to the receiving process, saying it wants to send a message of this size
  - Receiving process allocates the buffer space and sends back a message saying it is ready
  - Only then can sending process send the data

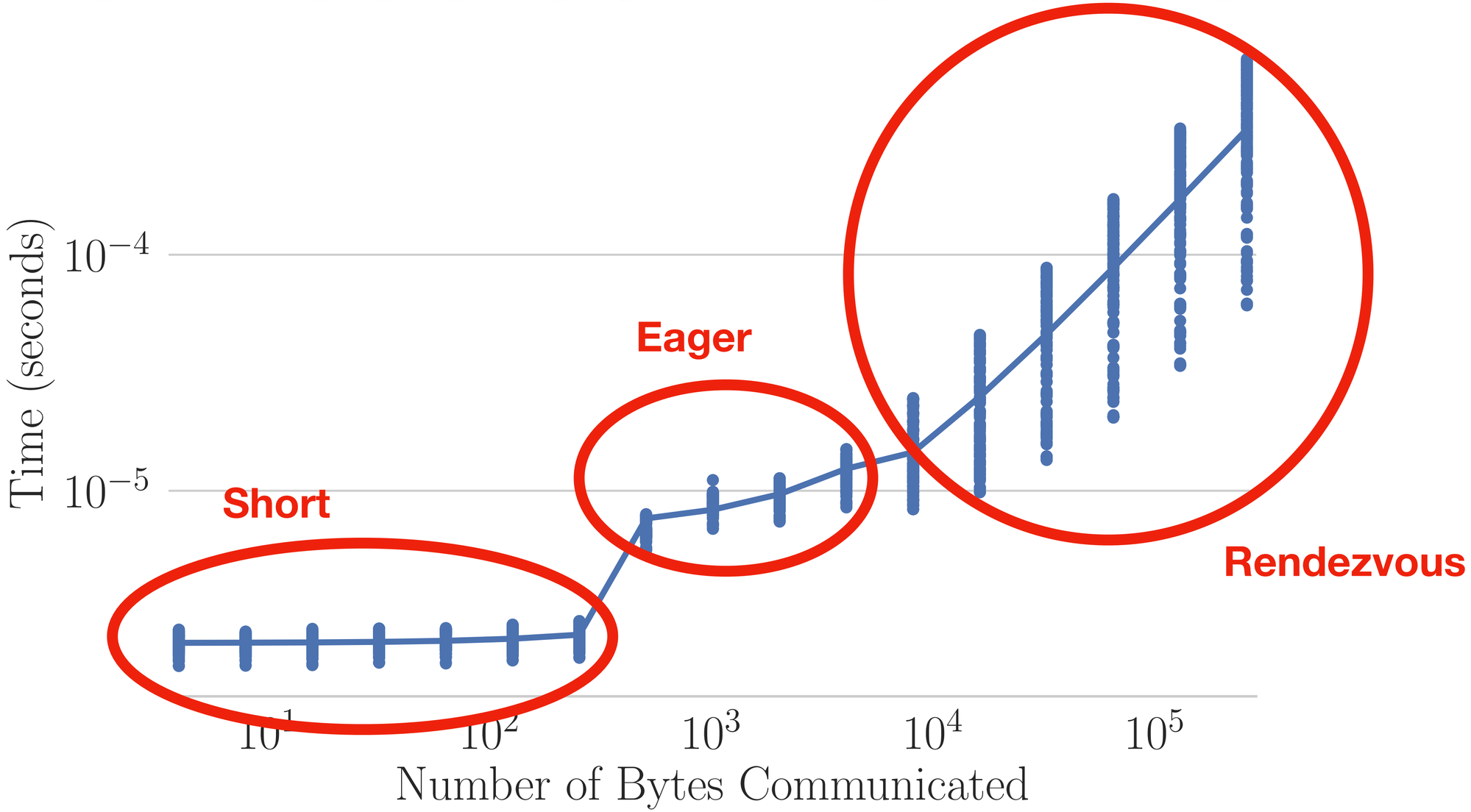
How do we measure cost of communication?

- Standard Approach : The postal model
  - T = alpha * n + beta * s
    - alpha = start-up time (latency) for sending each message
    - beta = per-byte transport cost
    - n = number of messages
    - s = number of bytes communicated
- Need one of these models for each of short, eager, and rendezvous communication

Inter-Node Communication

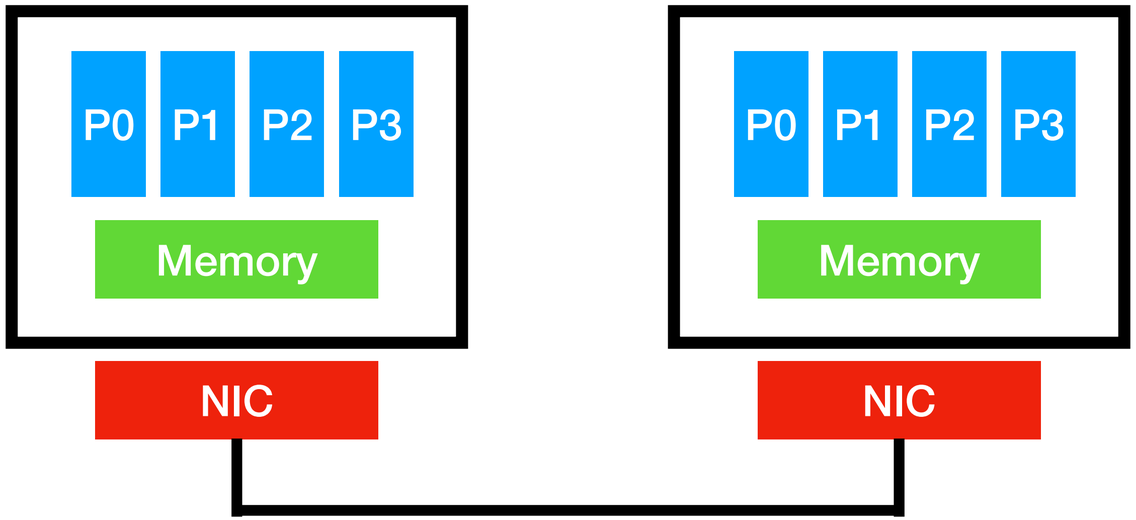


Inter-Node Communication



What are we not capturing?

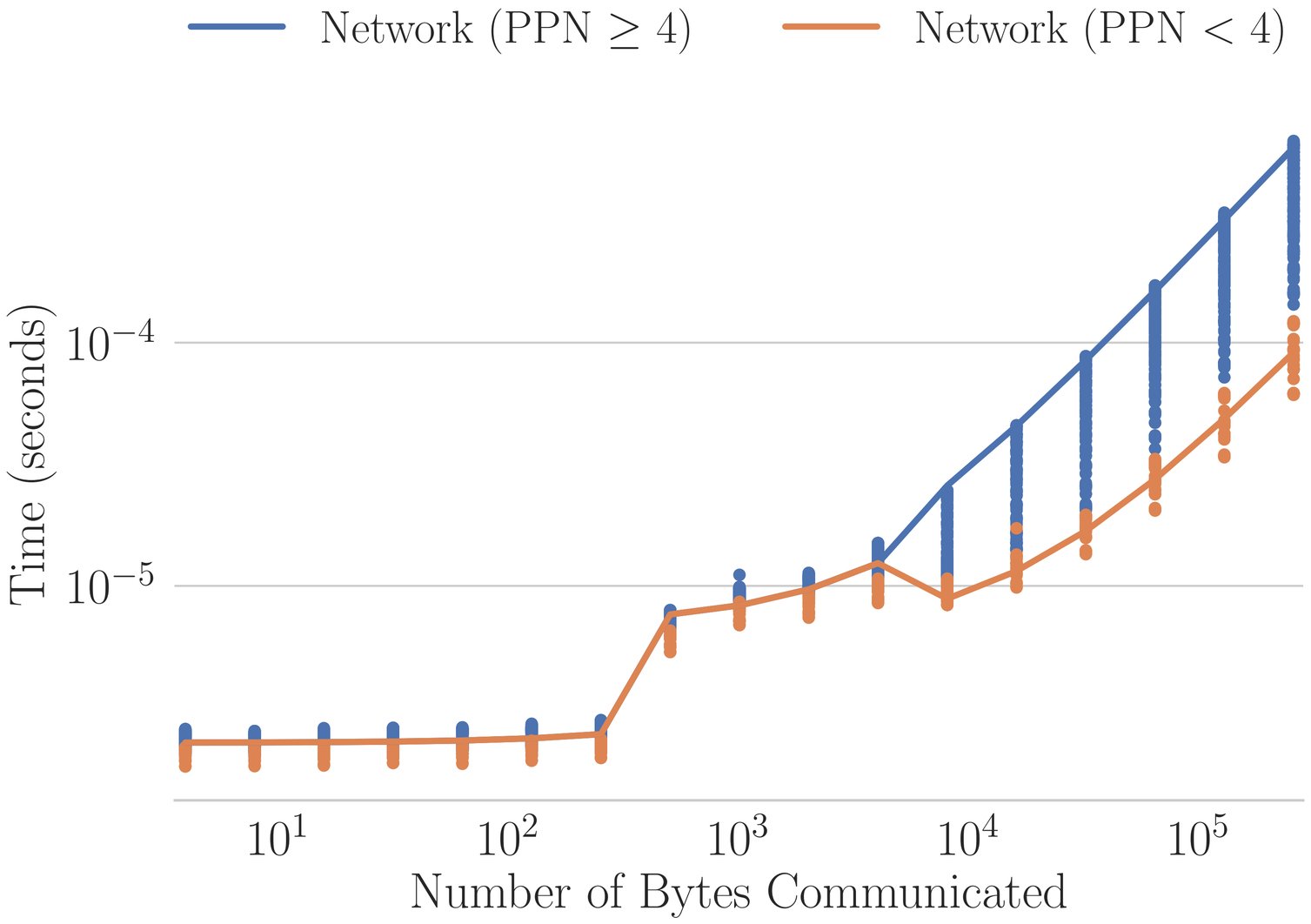
- Injection bandwidth limits
- Have multiple processes per node all sending data
- Network interface card (NIC) can only push so much data into the network at any time



How do we measure cost of communication?

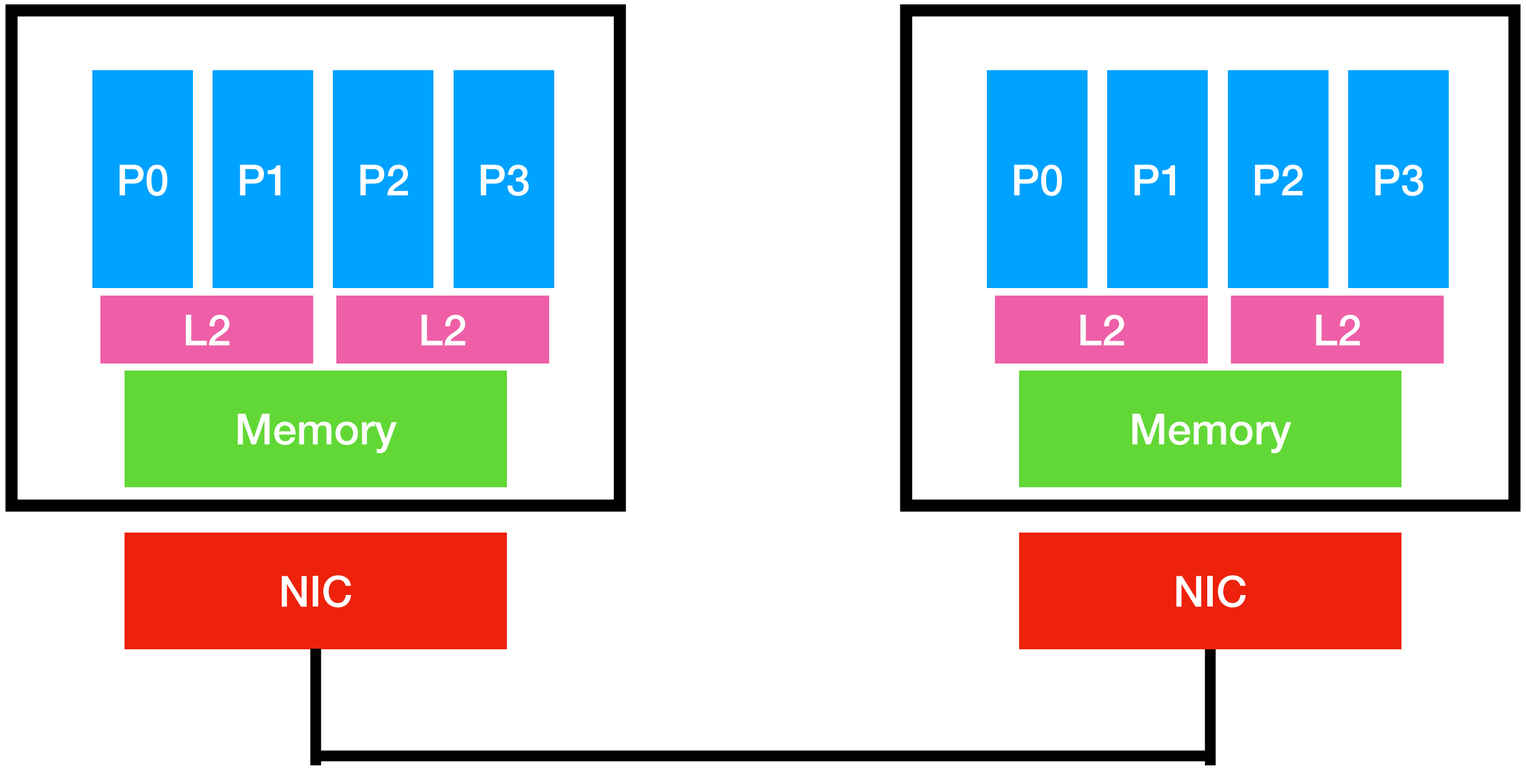
- Better approach : the max-rate model
  - $$T = \alpha \cdot n + \frac{\text{ppn} \cdot s}{\min(R_N, R_p \cdot \text{ppn})}$$
    - ppn = processes per node
    - R_{p} = inter-process bandwidth
    - R_{N} = injection bandwidth
    - s = number of bytes communicated
  - Beta from postal model is equal to inverse of R_{p}

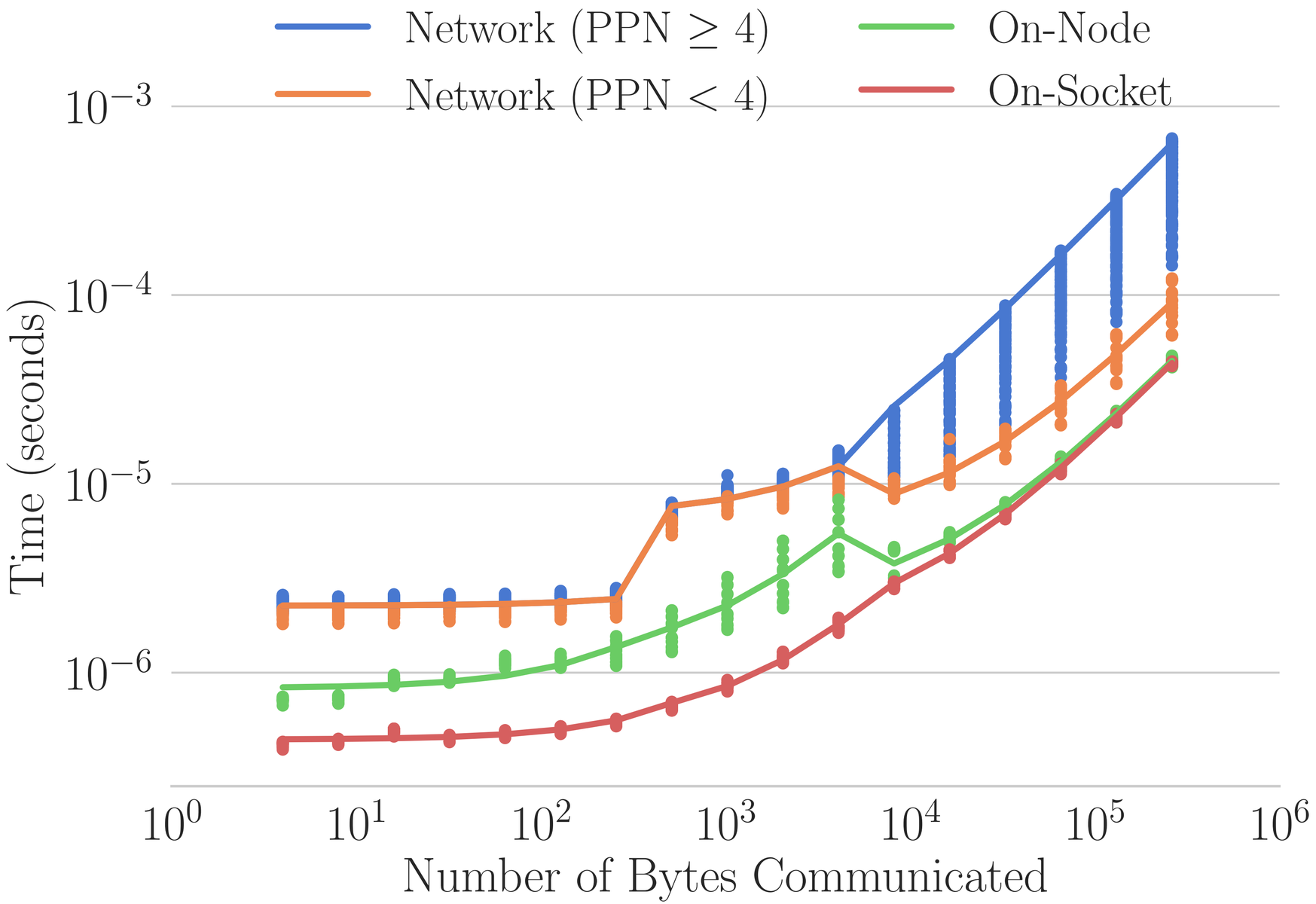
Inter-Node Communication



What about on-node communication?

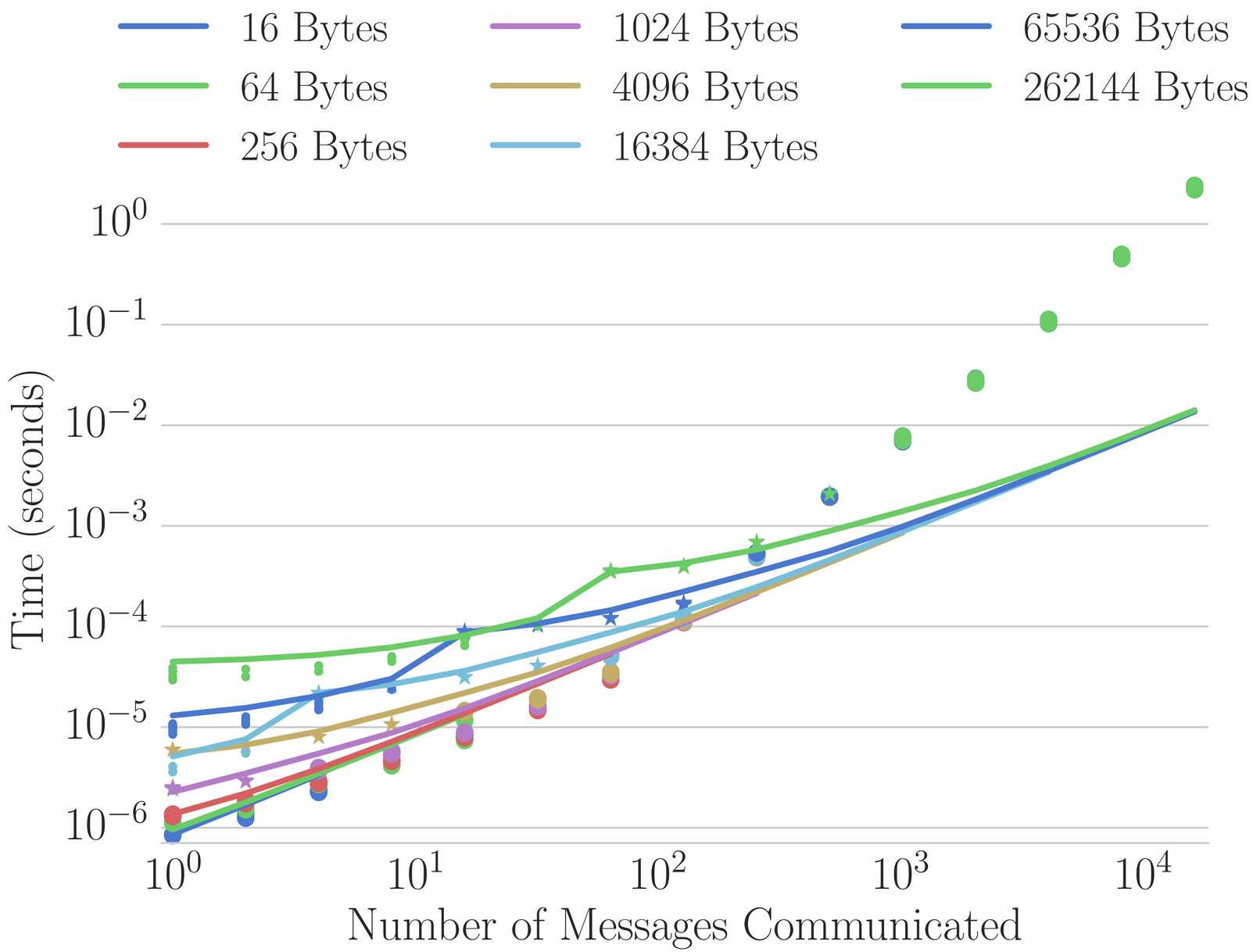
- Cost of communication varies greatly based on relative locations of sending and receiving processes




Network (PPN ≥ 4)
Network (PPN < 4)
On-Node
On-Socket
Time (seconds)
10^−3
10^−4
10^−5
10^−6
10^0
10^1
10^2
10^3
10^4
10^5
10^6
Number of Bytes Communicated

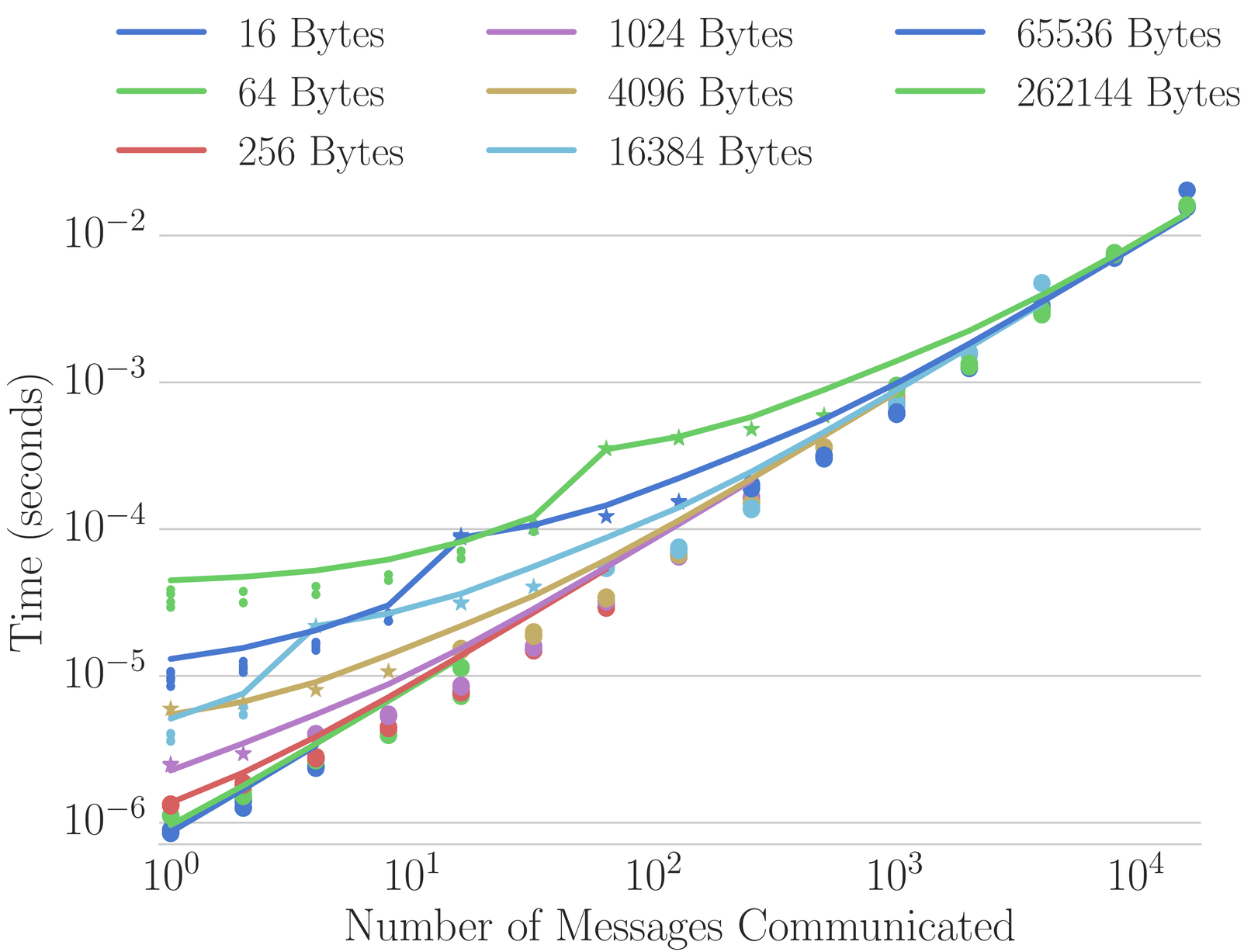
Large Numbers of Messages

- Queue Search :
- Each process posts MPI_Irecvs
- Each process sends their messages
- Received data gets lined up in a queue, and MPI steps through both the queue of received messages and the posted MPI_Irecvs to match them up
- Standard implementation steps through the entire queue and the entire list of incomplete MPI_Irecvs at each step (n^2 operation!)


16 Bytes
64 Bytes
256 Bytes
1024 Bytes
4096 Bytes
16384 Bytes
65536 Bytes
262144 Bytes
Time (seconds)
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
Number of Messages Communicated

Models don’t match timings

- When sending large number of messages, models and measured timings don’t match, because queue search cost is not part of model
- Can add this to max-rate model (or postal model) by adding the following:
  - $T += \theta n^2$


16 Bytes
64 Bytes
256 Bytes
1024 Bytes
4096 Bytes
16384 Bytes
65536 Bytes
262144 Bytes
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
Time (seconds)
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
Number of Messages Communicated