SEBASTIAN AMENT

RELIABLE AND SCALABLE DISTRIBUTED SYSTEMS

ONE INCREASINGLY IMPORTANT APPLICATION

HIGH-ASSURANCE CLOUD COMPUTING

*Examples from Ken Birman's Presentation for CS6410 Fall 2015

AGENDA

- Presentation and Discussion of
  - The Process Group Approach to Reliable Distributed Computing, Birman. CACM, Dec 1993, 36(12):37-53.
  - Sinfonia: A new paradigm for building scalable distributed systems. Marcos K. Aguilera, Arif Merchant, Mehul Shah, Alistair Veitch, Christos Karamanolis. November 2009 Transactions on Computer Systems (TOCS), Volume 27 Issue 3
- Background
  - Eric Brewer's CAP Conjecture / Theorem
  - Bimodal Multicast

PROCESS GROUP TAKEAWAYS

- Reliable components do not make a reliable system
- Many applications are naturally structured into groups
- In distributed computing, Performance ~ Asynchrony
- Virtual Synchrony

COMPARISON TO STATE MACHINES

- "The *state machine approach* is a general method for implementing a fault-tolerant service by replicating servers and coordinating client interactions with server replicas" - Fred B. Schneider - Implementing Fault-Tolerant Services Using the State Machine Approach: A Tutorial
- Both approaches deal with reliable interprocess communication
- Members of process groups can collaborate in a variety ways, not just replicate

SINFONIA TAKEAWAYS

- Paradigm for distributed infrastructure applications
- Interprocess communication via atomic minitransactions
- Allows for implementation of process group approach
- Support for caching at application level
- Clever increase of asynchrony / performance through a pre-computation step

COMPARISON BETWEEN PROCESS GROUP APPROACH AND SINFONIA

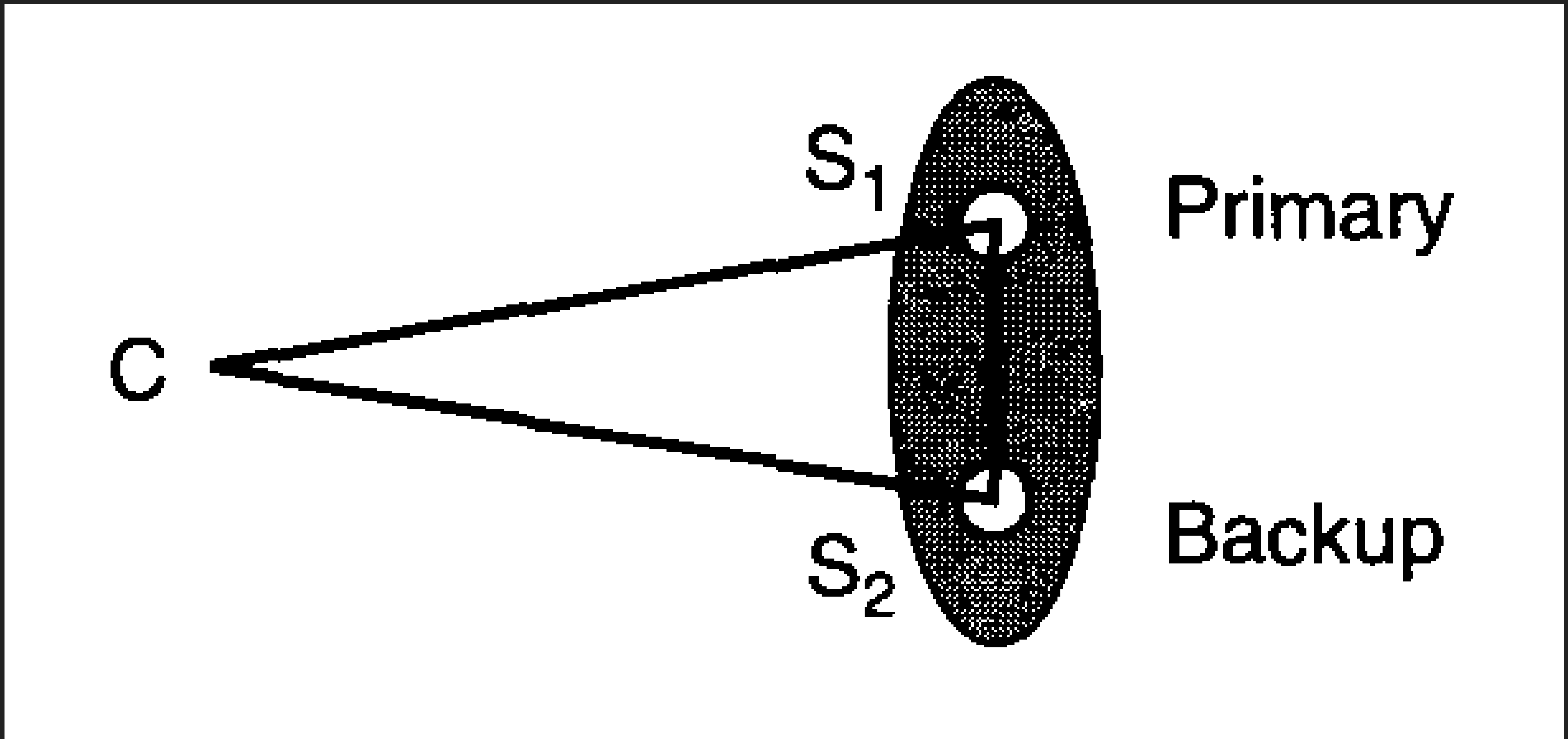
- Both are concerned with making distributed systems reliable and efficient
- Sinfonia possesses more generality
- Both identified the main performance factor of distributed systems to be the decoupling of processes
  - Process group: Virtual Synchrony
  - Sinfonia: Minitransactions, Pre-computation, Caching

ERIC BREWER'S CAP CONJECTURE

- CAP stands for*
  - Consistency: reads receive the most recent version of the data or an error
  - Availability: every request receives a response
  - Partition tolerance: the system is fault-tolerant with respect to network failures
- States that one cannot have all three simultaneously.
- Trade-offs are possible

*Definitions from wikipedia.com: CAP Theorem

A NAIVE APPROACH TO RELIABLE DATA STREAMS



- What happens if connection between C and S1 breaks?
- Not partition-tolerant

Figure from Birman 1993

MESSAGE-ORDERING PROBLEMS

- Concurrent messages
- Dependencies
- Note: This figure does not seem to fit the text

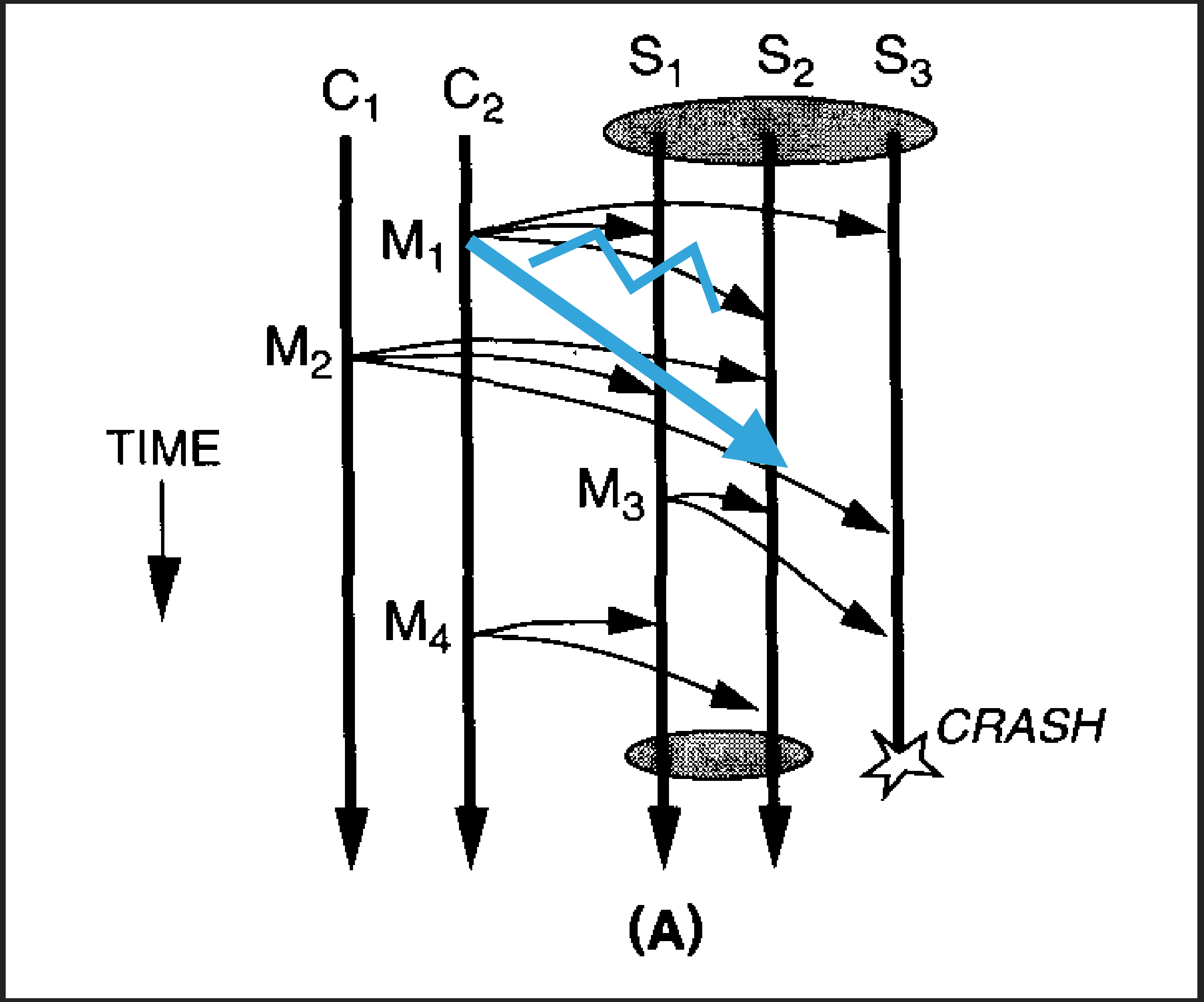
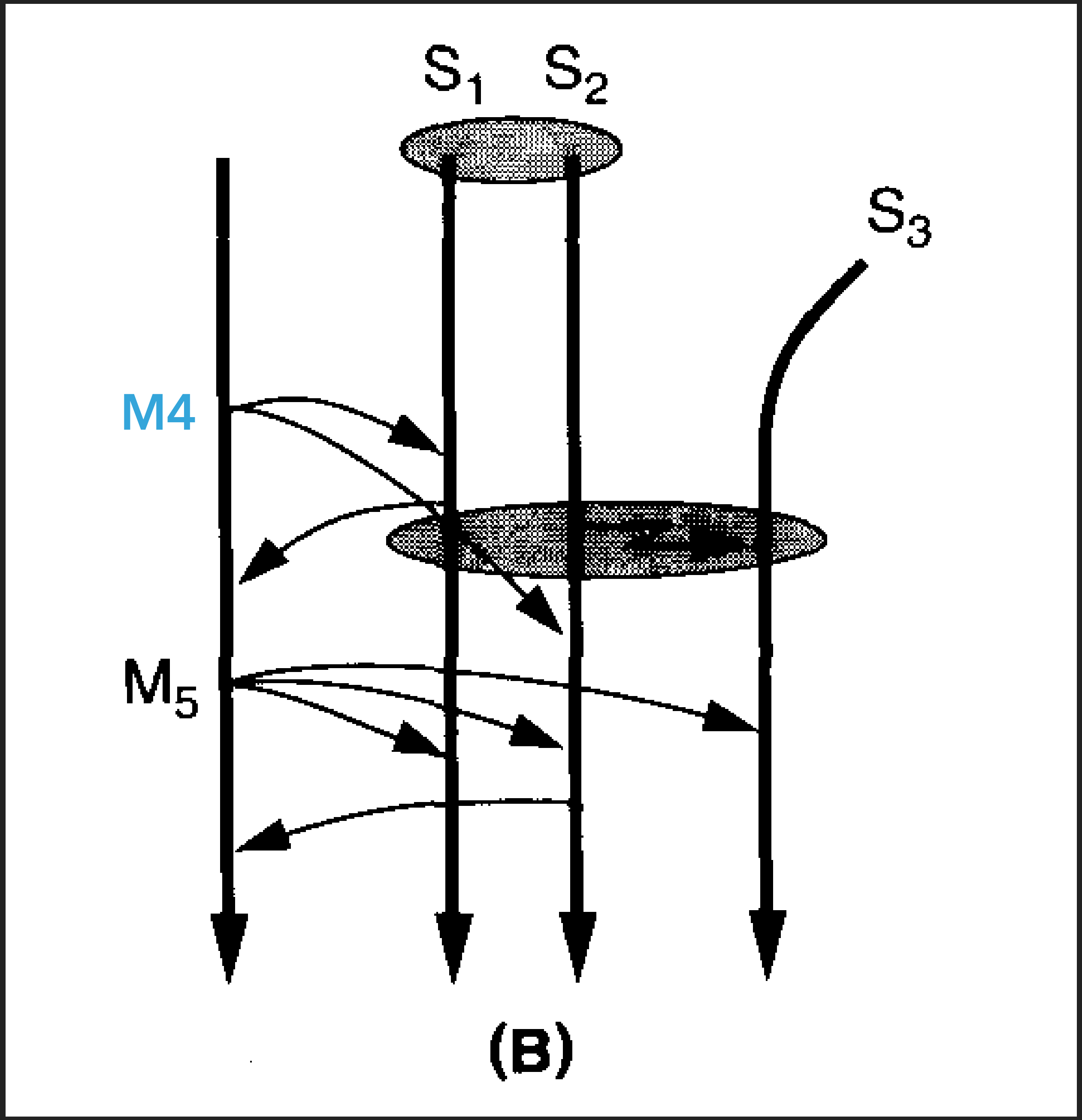


Figure from Birman 1993

STATE TRANSFER PROBLEMS



- S3 is not consistent

Figure from Birman 1993

SUMMARY OF CHALLENGES

- Reliable communication, especially in the presence of network failures
- Consistent group membership information
- Sensible group message ordering
- Consistent state transfer
- Fault tolerance

> ONE MIGHT EXPECT THE RELIABILITY OF A DISTRIBUTED SYSTEM TO CORRESPOND DIRECTLY TO THE RELIABILITY OF ITS CONSTITUENTS, BUT THIS IS NOT ALWAYS THE CASE.

Kenneth P. Birman – The Process Group Approach to Reliable Distributed Computing

THE PROCESS GROUP APPROACH

- Goals: scalability, reliability, and simplicity
- Structure processes in groups
- Focus on inter- and intra group communication
- Example: Isis
  - Group multicast
  - Virtual synchrony
  - Persistent storage

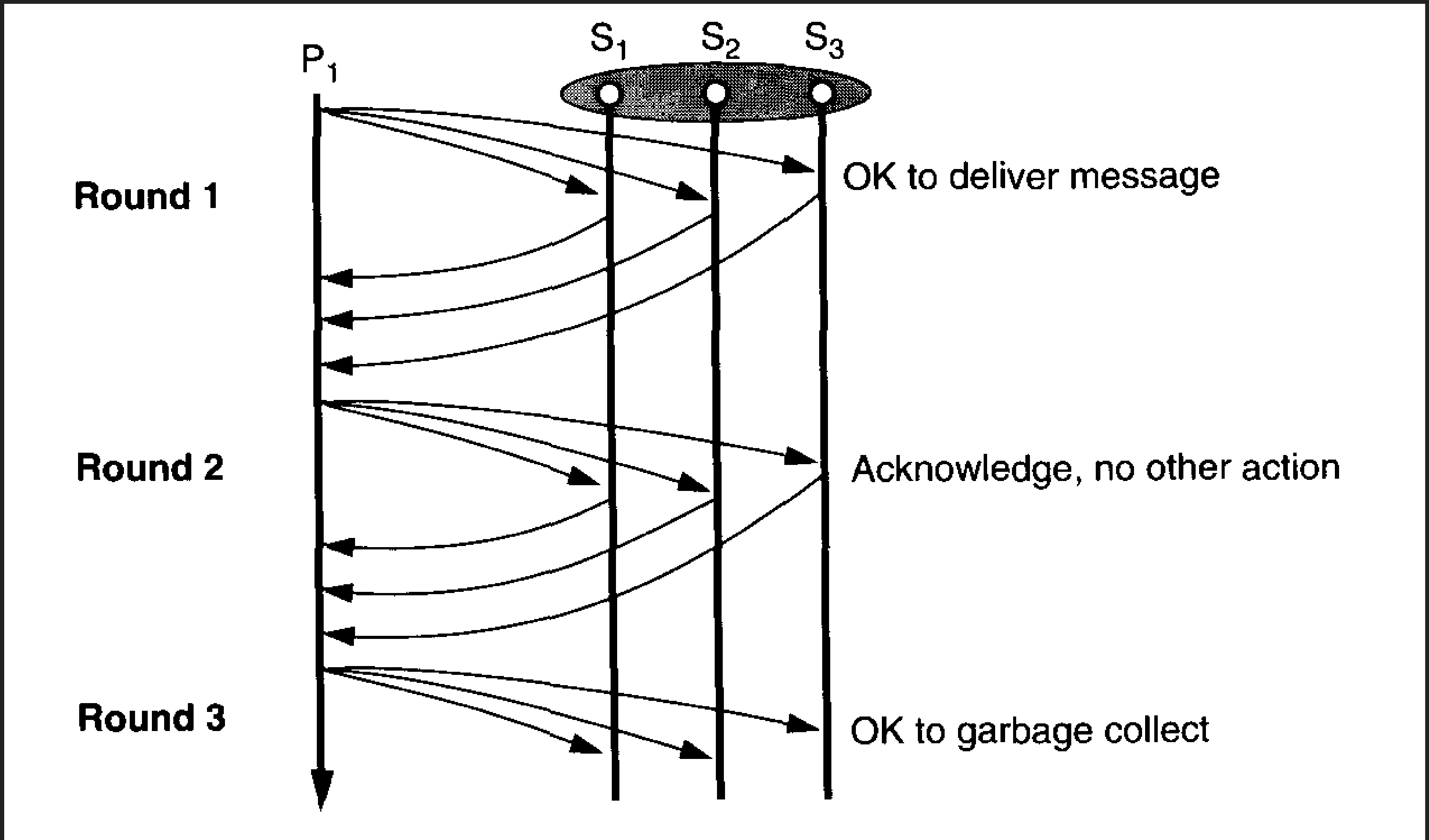
TWO BASIC TYPES OF GROUPS

- Anonymous groups
    - Arise with producer/consumer applications
    - Messages should be delivered once, in sensible order
- Explicit groups
    - Arise when applications are cooperating
    - Need consistent membership information to be effective

COMMUNICATION BETWEEN GROUP MEMBERS

- Multicast: message from one process to multiple destination processes
  - Ensure that a message is received once by every destination
  - Ensure that multicasts are received in the same order by each process
- Explicit groups can communicate in ways unique to the application

THREE-ROUND MULTICAST



- No pipelining, or asynchronous communication

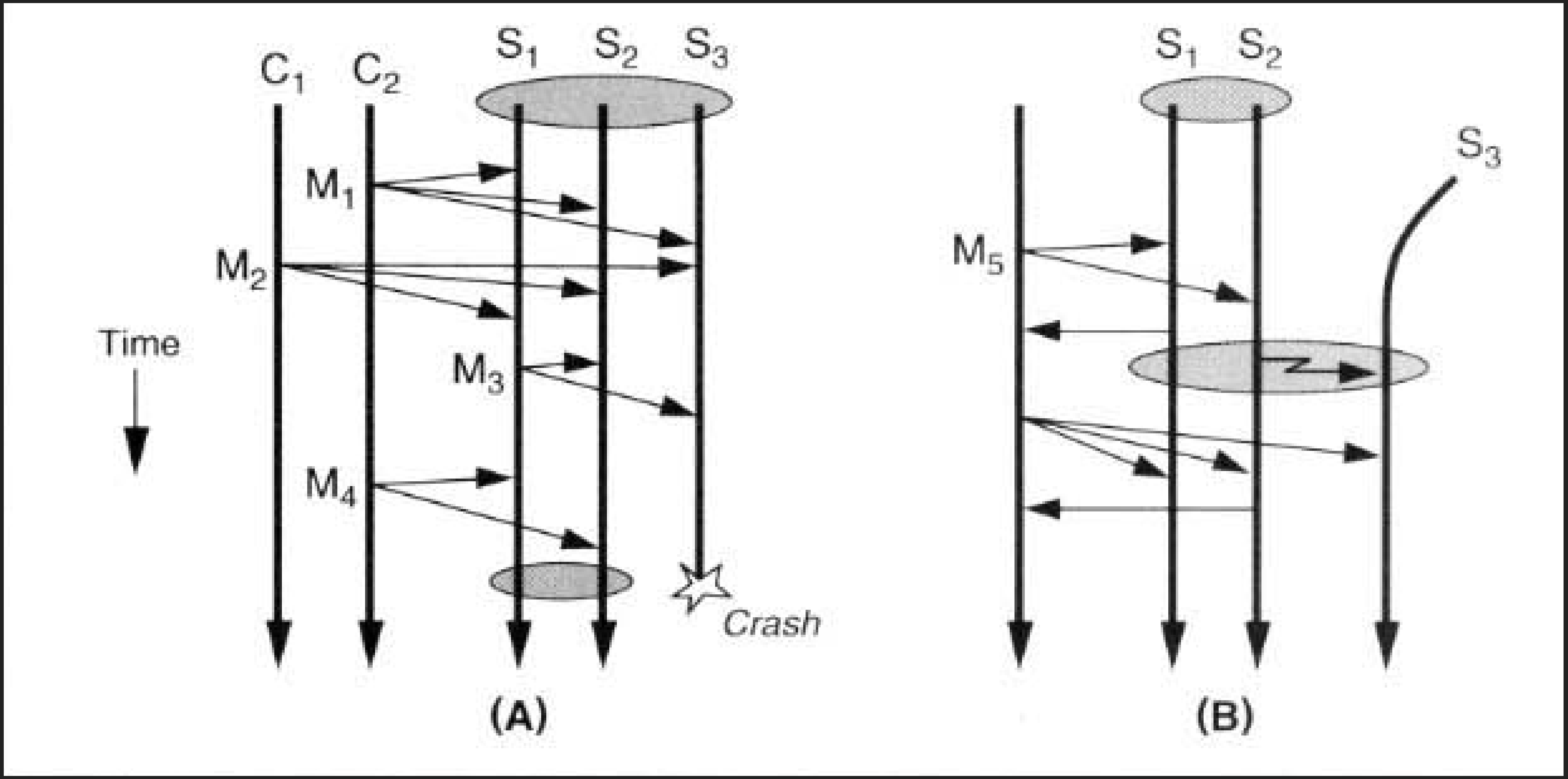
Figure from Birman 1993

CLOSELY SYNCHRONOUS EXECUTION

- Execution of each process consists of events
- A global execution consists of a set of process executions
- Every global event is seen in the same order by each process in a group
- A multicast to a group is delivered to all processes of that group

Definition from Birman 1993

CLOSELY SYNCHRONOUS EXECUTION



- Synchronous execution grants reliable, predictable behavior.
- What about efficiency?

Figure from Birman 1993

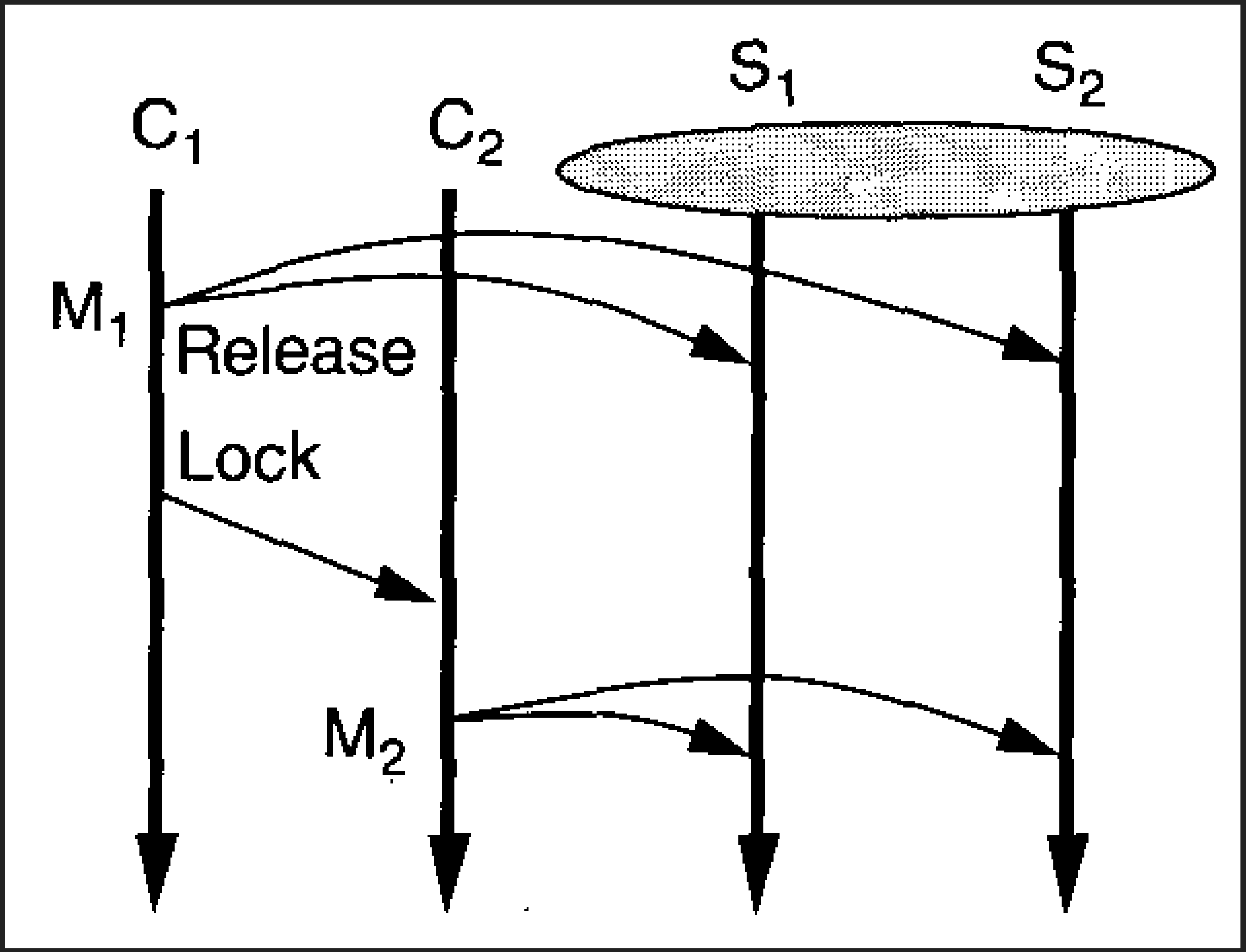
ACHIEVING CLOSE SYNCHRONY IS IMPOSSIBLE IN THE PRESENCE OF FAILURES.

Kenneth P. Birman – The Process Group Approach to Reliable Distributed Computing

VIRTUAL SYNCHRONY

- Allows for those asynchronous executions, which are indistinguishable from synchronous ones
- Prospect of dramatic performance improvement
- Sensible event ordering is crucial
- In the absence of synchronized clocks, this cannot be a temporal ordering
- See Lamport's "Time, Clocks, and the Ordering of Events in a Distributed System"

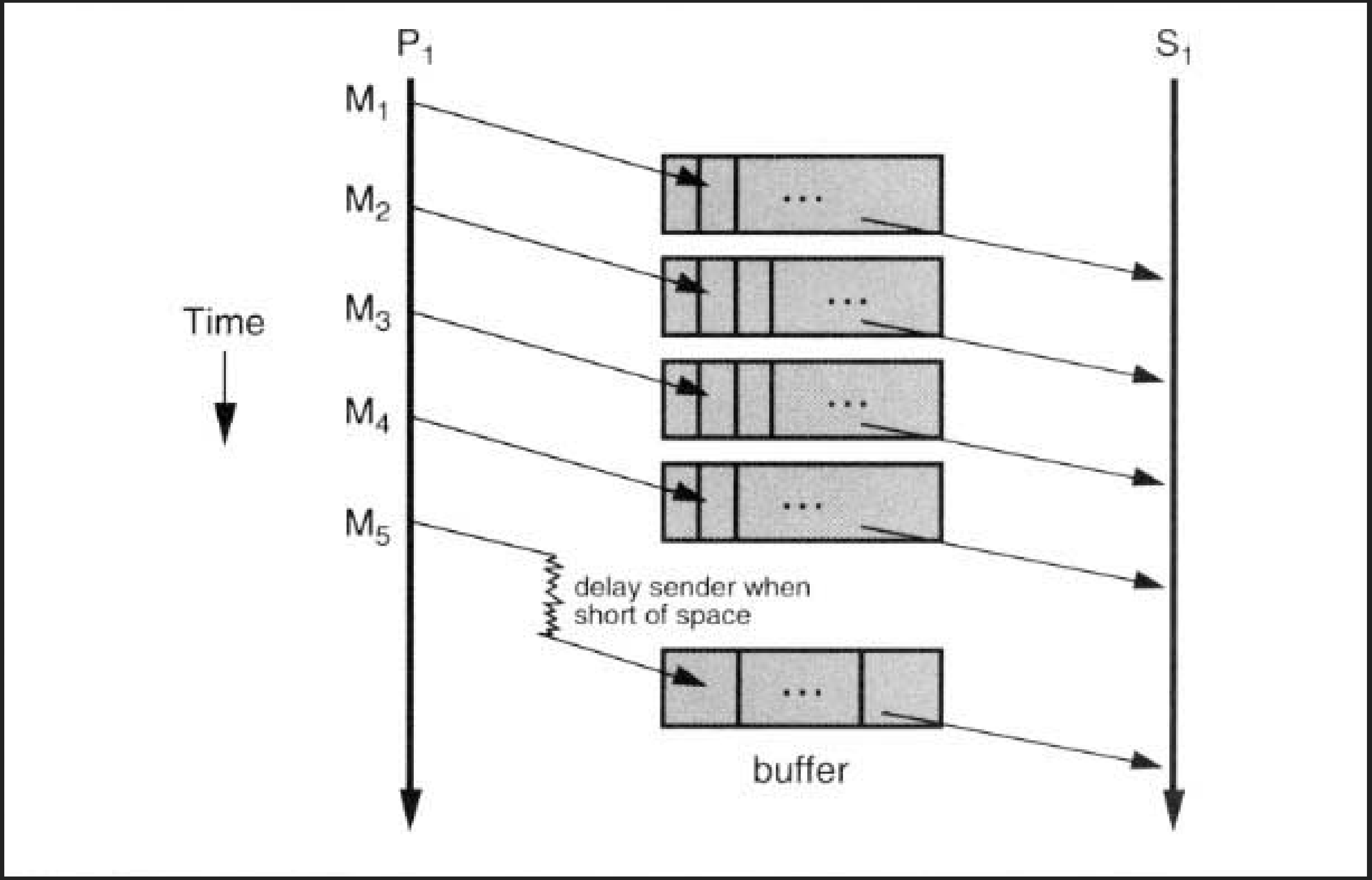
CAUSAL EVENT ORDERING



- FIFO message delivery between a client-pair won't work
- M1 has to be delivered to <u>both</u> S1 and S2 before M2

Figure from Birman 1993

ASYNCHRONOUS PIPELINING



- Allows P1 to make progress while S1 has not issued a read
- This works in the presence of multiple S's

Figure from Birman 1993

BENEFITS OF VIRTUAL SYNCHRONY

- Efficient, asynchronous communication
- Allows developers to assume close synchrony
- The notion of group state is sensible
- Dynamically changing group membership can be handled easily
- Any disadvantages? Let's look at the assumptions.

WE ALSO ASSUME THAT LAN COMMUNICATION PARTITIONS ARE RARE.

Kenneth P. Birman – The Process Group Approach to Reliable Distributed Computing

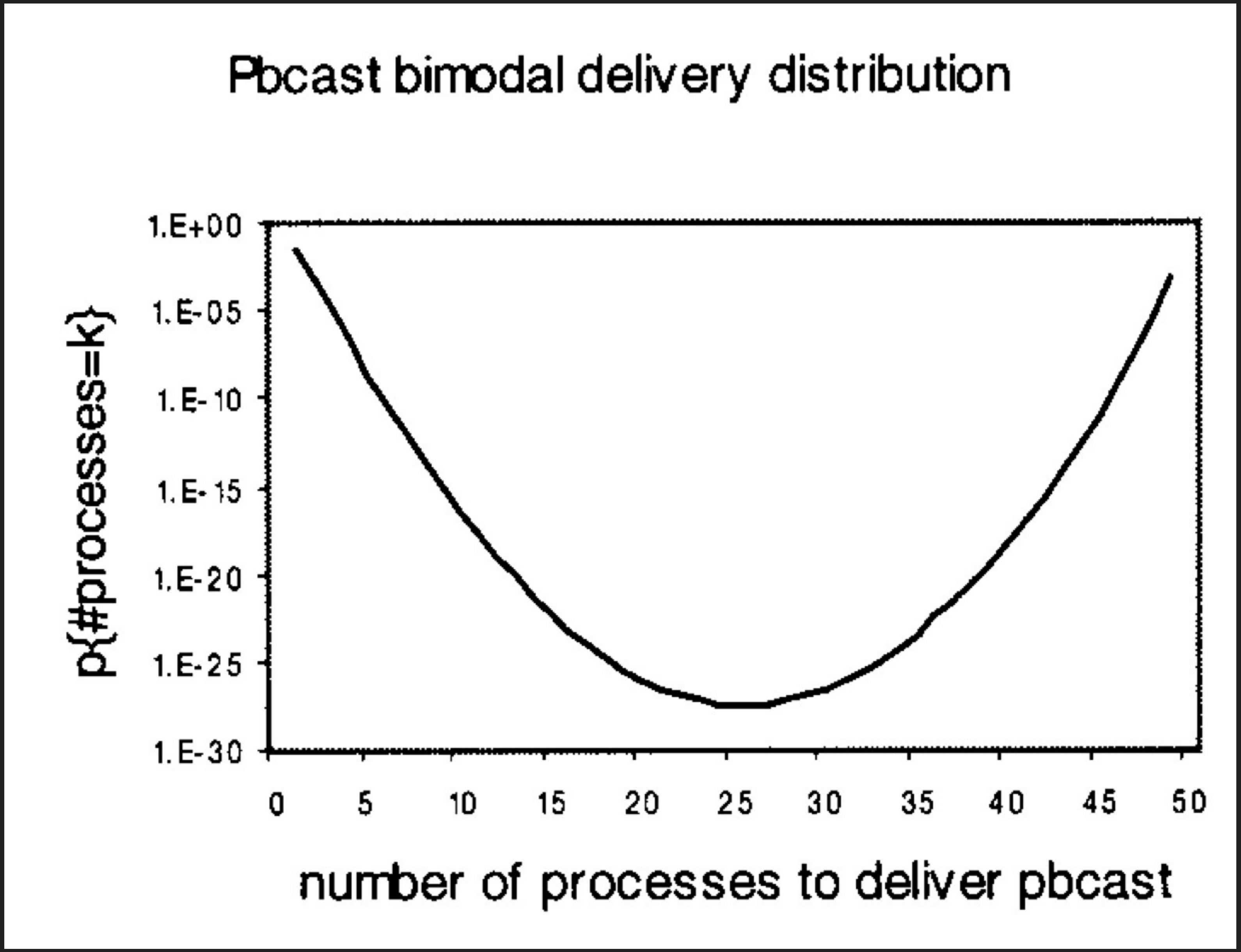
[…] PROCESSES AND PROCESSORS ARE ASSUMED TO FAIL BY HALTING WITHOUT INITIATING ERRONEOUS ACTIONS OR SENDING INCORRECT MESSAGES.

Kenneth P. Birman – The Process Group Approach to Reliable Distributed Computing

BIMODAL MULTICAST - BIRMAN ET AL 1999

- Uses two stages:
  - 1. Unreliable “best effort” delivery attempt
  - 2. Conditionally executed protocol which corrects message losses
- Stage 1 is efficiently implemented using a spanning-tree over the application nodes

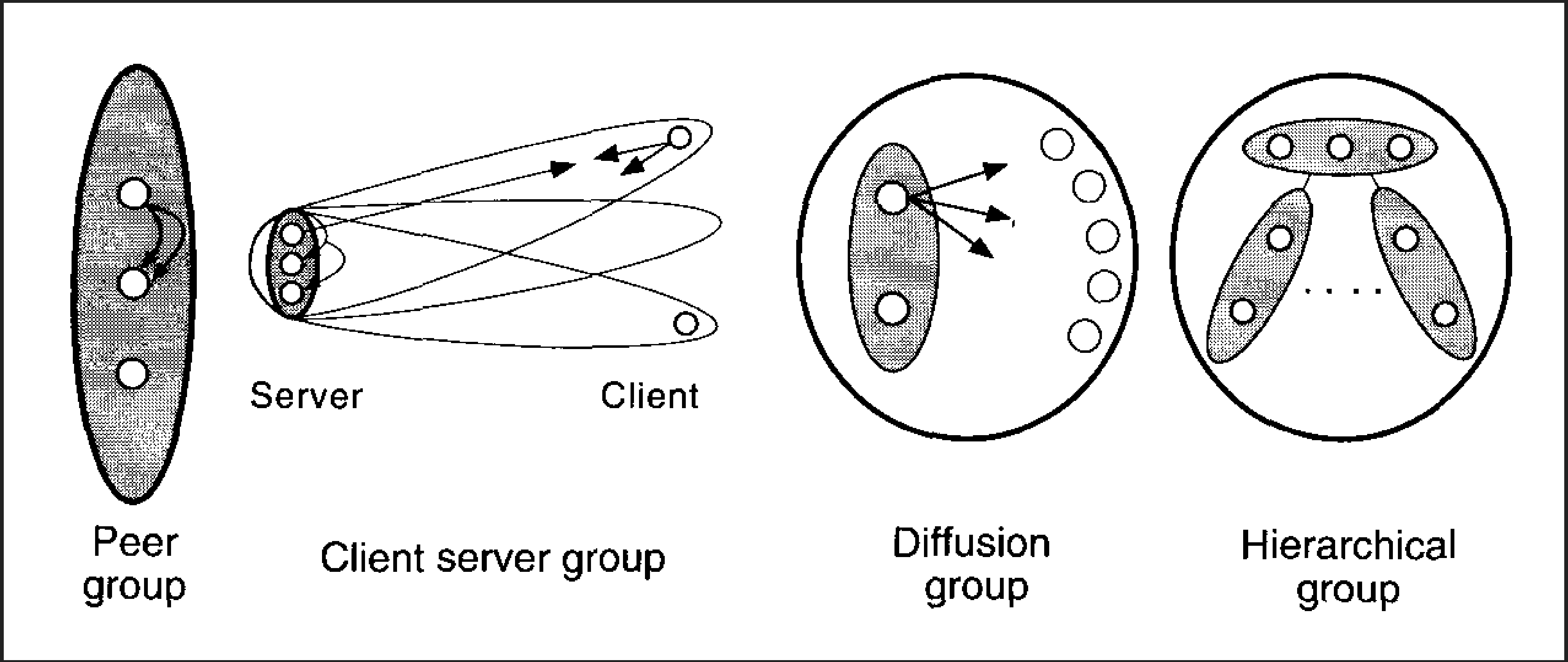
BIMODAL MULTICAST



- Graph assumes unreliable stage 1 failed

Figure from Birman et al 1999

STYLES OF GROUPS IN ISIS



- The efficacy of groups relies on membership information

Figure from Birman 1993

SINFONIA

- Similar system to Isis, but more recent
- Targets data center infrastructure applications
- Goals: reliability, consistency, scalability
- Core idea: atomic minitransactions
- Message-passing protocols are not exposed to developer

THE CORE TO ACHIEVING SCALABILITY IS TO DECOUPLE OPERATIONS EXECUTED BY DIFFERENT HOSTS AS MUCH AS POSSIBLE

Aguilera et al – Sinfonia: A New Paradigm for Building Scalable Distributed Systems

MINITRANSACTIONS

- Reduced complexity as compared to database transactions
- Consist of three steps:
  - Pre-computation
  - Validation step
  - Action step

*Slide from Ken Birman's Presentation for CS6410 Fall 2015

MINITRANSACTIONS: PRE-COMPUTATION

- Gives Sinfonia remarkable performance and scalability
- Computations can be executed asynchronously on cached copies of the system state without updating disk image
- System states include uniquely identifiable version numbers
- If version numbers of state replicas match, an update to the core state can be made
- This can hide compute and communication costs

*From Ken Birman's Presentation for CS6410 Fall 2015

PERFORMANCE BENCHMARKS

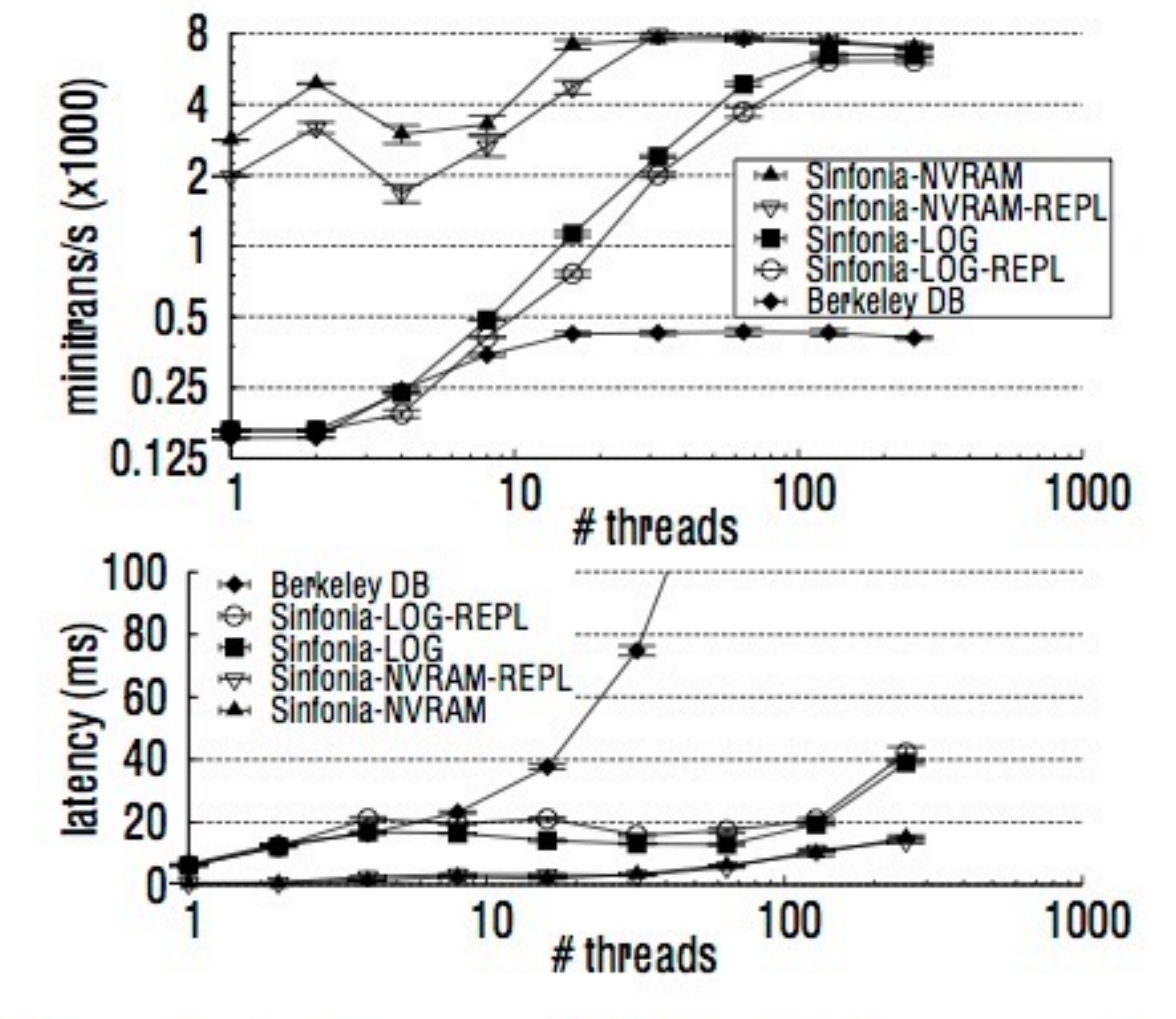


Figure 12: Performance of Sinfonia with 1 memory node.

Figure from Aguilera et al 2009

SCALABILITY BENCHMARKS

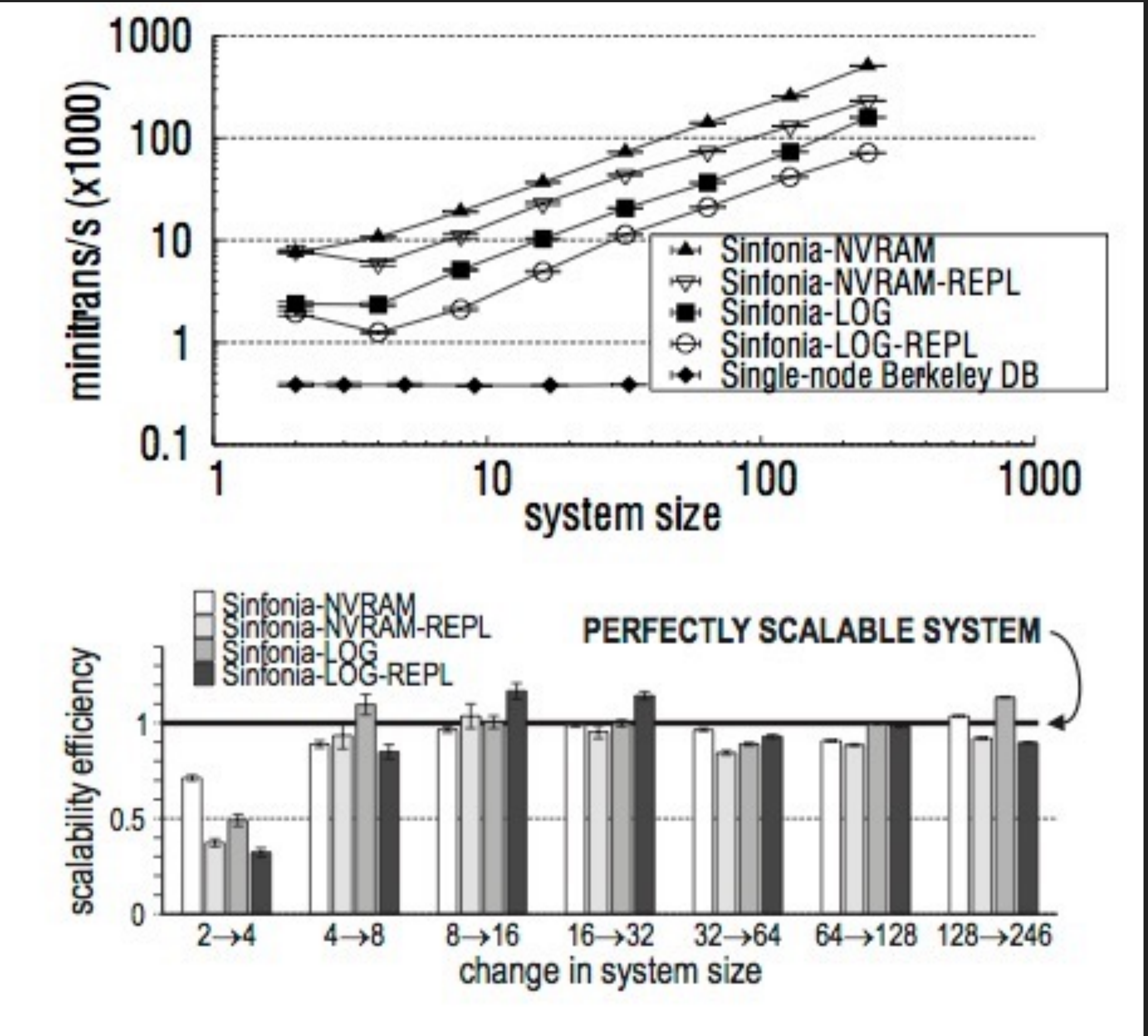


Figure 14: Sinfonia scalability.

Figure from Aguilera et al 2009

SINFONIA GROUP COMMUNICATION SYSTEM

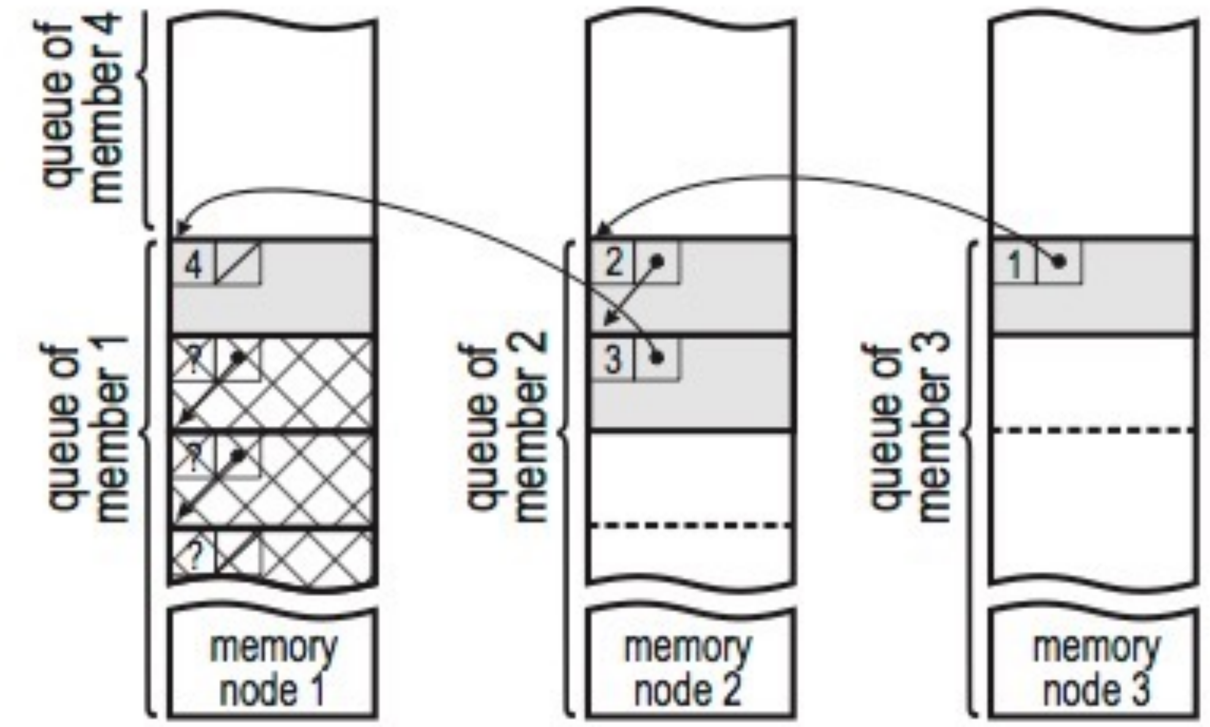


Figure 10: Basic design of SinfoniaGCS. Gray messages were successfully broadcast: they are threaded into the global list. Cross-hatched messages are waiting to be threaded into the global list, but they are threaded locally.

- Messages are stored in local circular queue
- Also, a reference to them is put in a global list

Figure from Aguilera et al 2009

SINFONIA GROUP COMMUNICATION SYSTEM

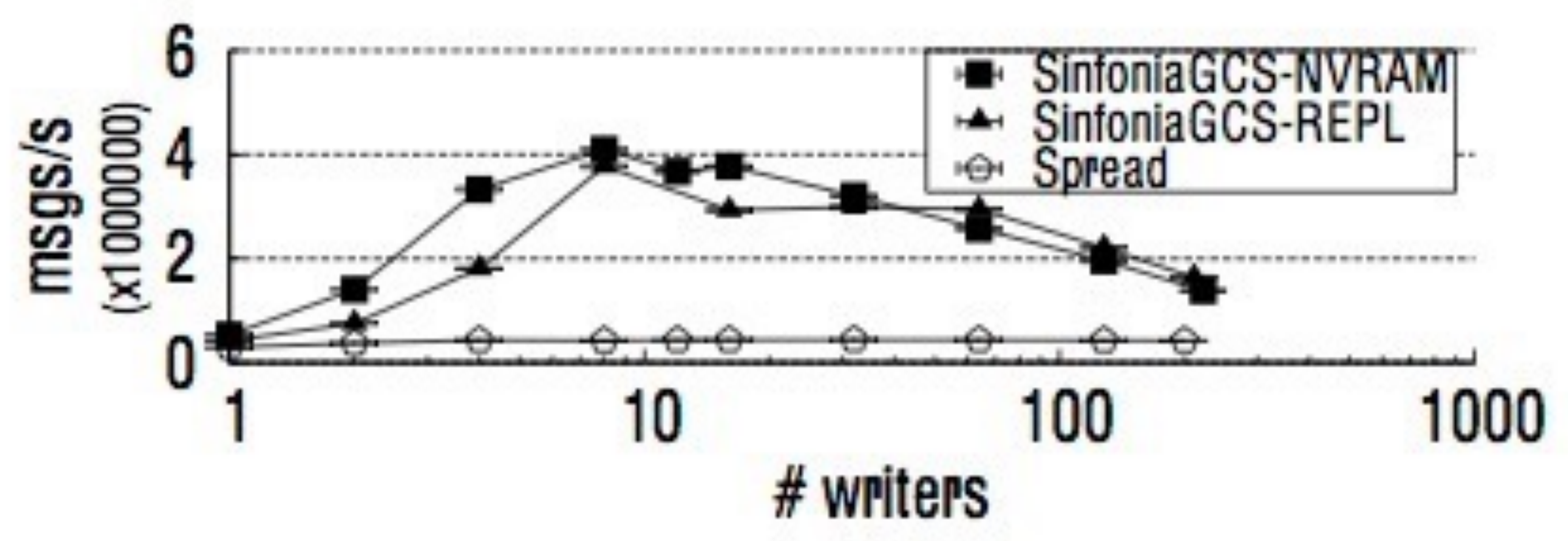


Figure 23: SinfoniaGCS base performance as we vary the number of writers. There are 16 readers, and 8 memory nodes or Spread daemons.

Figure from Aguilera et al 2009

COMPARISON BETWEEN PROCESS GROUP APPROACH AND SINFONIA

- Both are concerned with making distributed systems reliable and efficient
- Sinfonia possesses more generality
- Both identified the main performance factor of distributed systems to be the decoupling of processes
  - Process group: Virtual Synchrony
  - Sinfonia: Minitransactions, Pre-computation, Caching

DISCUSSION

- What trade-offs are Sinfonia and ISIS making with respect to CAP? Are the trade-offs different?
- How does Sinfonia GCS's multicast strategy compare to ISIS?
- What did you find interesting about the papers?
- Feel free to ask your own question!

THANK YOU FOR LISTENING

SOURCES

- The Process Group Approach to Reliable Distributed Computing, Birman. CACM, Dec 1993, 36(12):37-53.
- Bimodal multicast. Kenneth P. Birman, Mark Hayden, Oznur Ozkasap, Zhen Xiao, Mihai Budiu, and Yaron Minsky. 1999. ACM Trans. Comput. Syst. 17, 2 (May 1999), 41-88. DOI=http://dx.doi.org/10.1145/312203.312207
- Sinfonia: A new paradigm for building scalable distributed systems. Marcos K. Aguilera, Arif Merchant, Mehul Shah, Alistair Veitch, Christos Karamanolis. November 2009 Transactions on Computer Systems (TOCS), Volume 27 Issue 3
- Implementing fault-tolerant services using the state machine approach: A tutorial, Fred Schneider. ACM Computing Surveys Volume 22, Issue 4 (December 1990), 299--319.
- Can Cloud Computing Systems Offer High Assurance Without Losing Key Cloud Computing Properties?, Ken Birman, http://www.cs.cornell.edu/courses/CS6410/2015fa/slides/HAatLargeScale.pdf
- wikipedia.com: CAP theorem