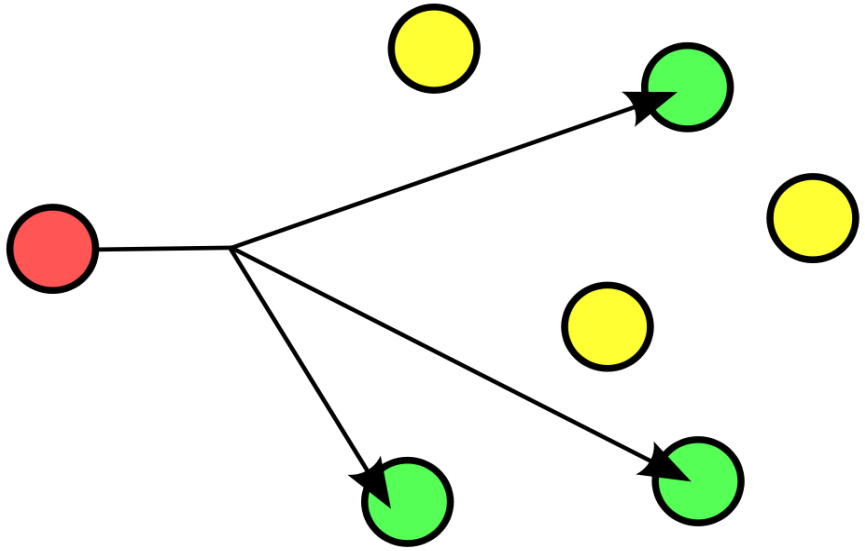
BIMODAL MULTICAST

KENNETH P. BIRMAN, MARK HAYDEN, OZNUR OZKASAP, ZHEN XIAO, MIHAI BUDIU and YARON MINSKY

Presented by: Anirudh Jayakumar

Multicast

Transmit a single message to a group of recipients



Reliable multicast: All non-faulty processes should receive the same set of multicasts

Reliable Multicast

- many protocols to make multicast reliable [virtual synchrony, SRM]
- Broadly split into two classes
  - Class 1: Strong reliability
    - Atomicity
    - Security properties
    - Real-time guarantees
  - Class 2: best effort reliability
    - Scalable
    - over come message loss or failure
    - Process join and exit asynchronously

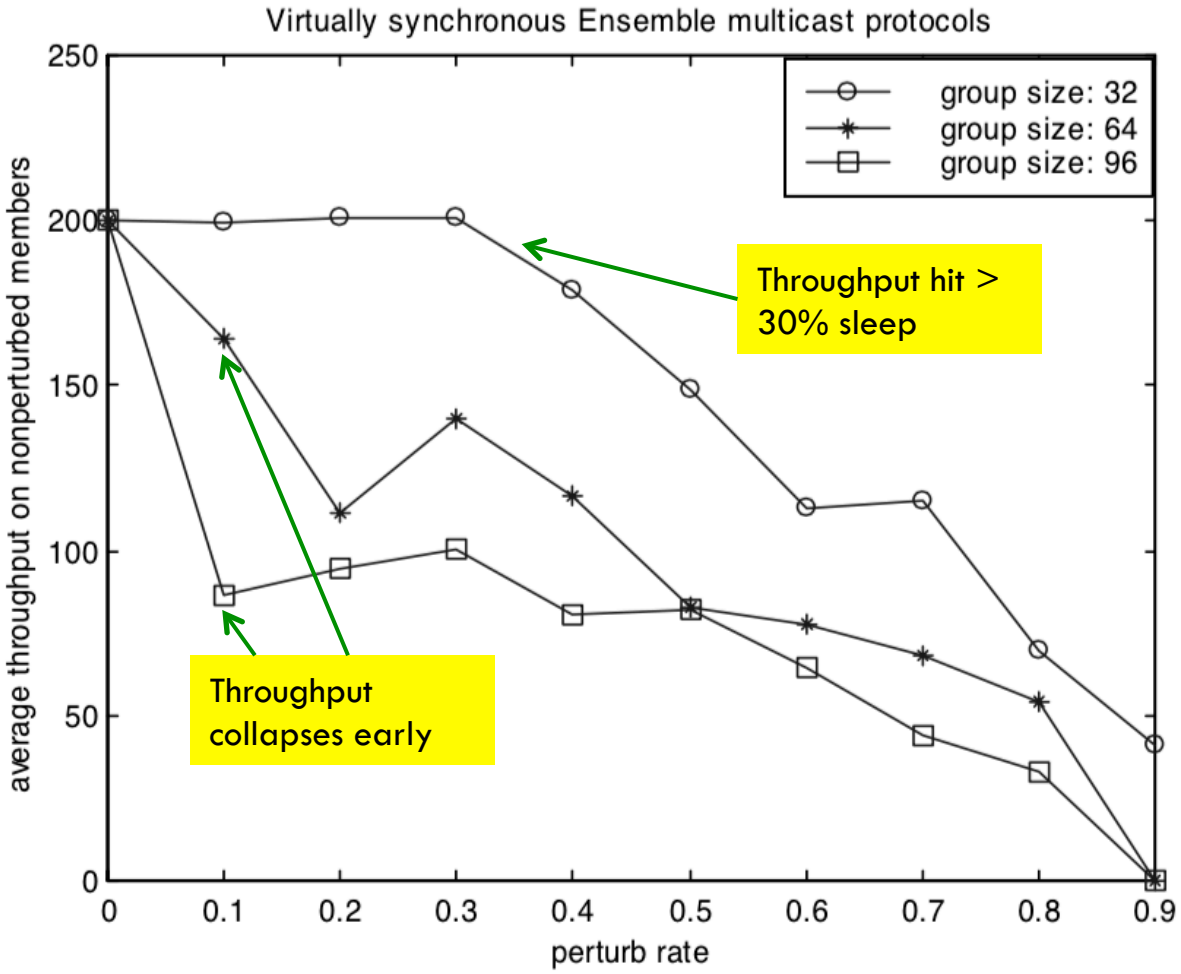
Issues

- Class 1: Strong reliability
  - Costly protocols
  - Unpredictable performance under stress
  - Limited scalability
- Class 2: best effort reliability
  - No end-to-end reliability guarantee
  - Gaps in message delivery may not be repaired
  - No core system to track membership

Goal of this paper

- For critical applications both classes of protocol are not acceptable
  - Class 1: impacts throughput
  - Class 2: becomes impossible to reason about the behavior of the system
- Bimodal multicast protocol (or) pbcast
  - Scalable
  - Predictable reliability even under perturbed conditions
  - Stable throughput

Throughput in class 1



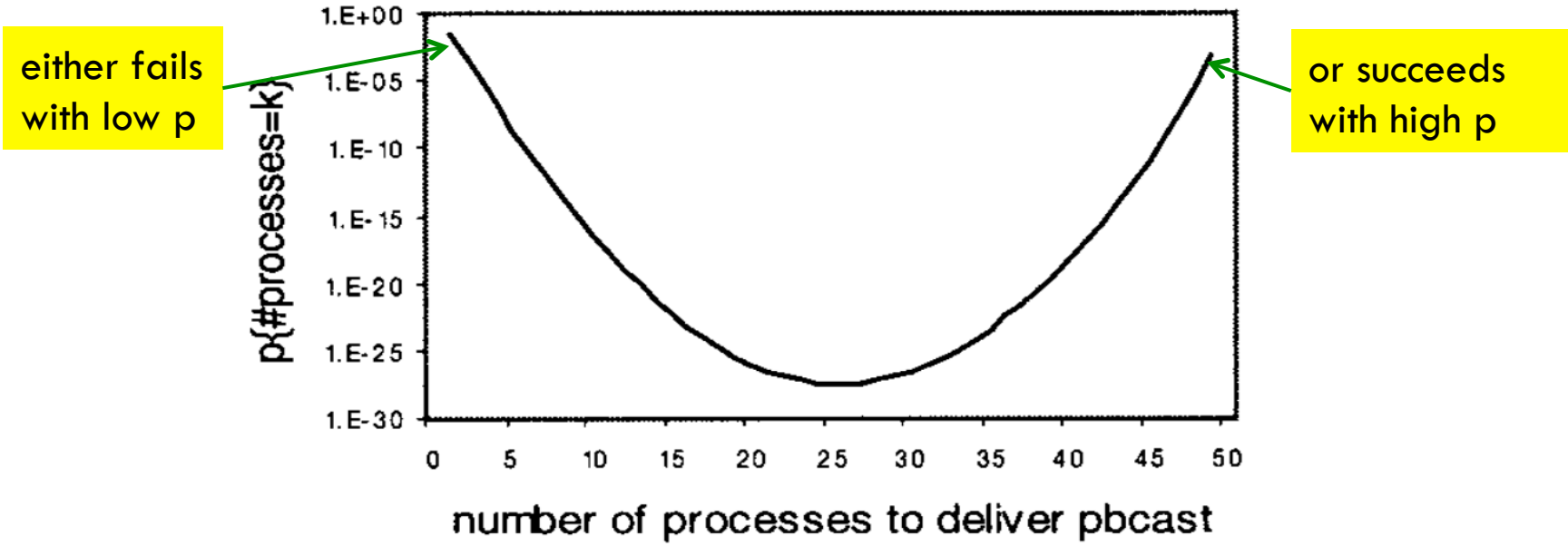
- 7KB message
- 200/sec
- SP2 cluster

Pbcast protocol

- 2 protocols in 1
- Step 1: Optimistic dissemination protocol
  - Unreliable, hierarchical broadcast
  - Best-effort attempt to delivery message
  - Choice of multicast protocol depends on the network and scalability requirements
- Step 2: Two-Phase Anti-Entropy Protocol
  - Phase 1: detect message loss
  - Phase 2: corrects such losses

Properties

- *Atomicity: redefined*
  - high probability - multicast reaching almost all processors
  - small probability - multicast reaching small set of processors
  - vanishingly small probability – multicast reaching intermediate number of processors



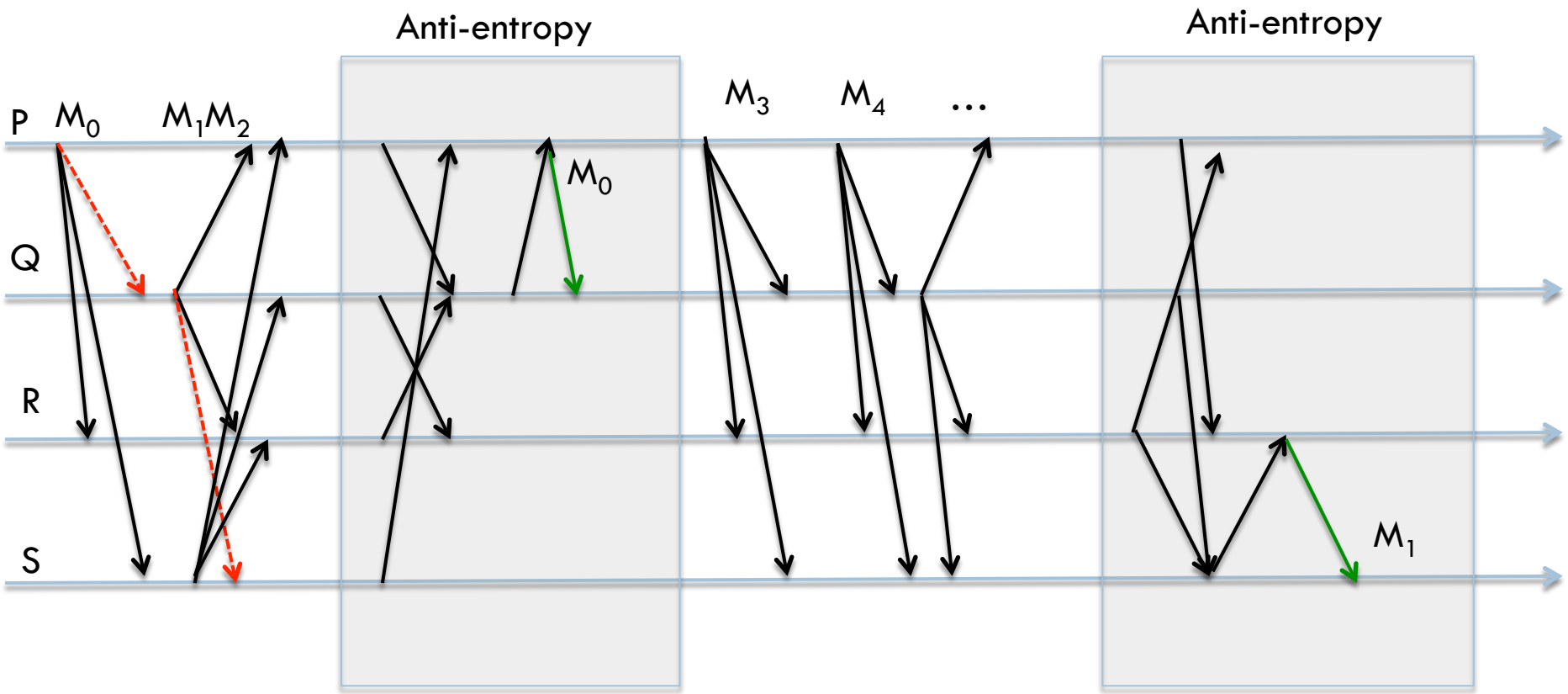
Properties

- *Throughput stability:*
  - Low variation in throughput
  - can be characterized for the settings of interest
- *Detection of lost messages:*
  - applications are informed about message loss
- *Scalability:*
  - Costs are constant or grow slowly as a function of network size

Two-Phase Anti-Entropy Protocol

- Members randomly choose a partner and sends summary [gossip message]
- The partner process will solicit any message that is missing in its buffer [solicitation message]
- Receiver of solicitation message retransmits some of the messages to the partner
- Message is garbage collected after fixed rounds of gossip
- fanout parameter: # rounds * # partners

Pbcast protocol



Some questions

- Will slow process catch up?
- What if a process is loaded with solicitation messages?
- Scalable over WAN?
- 7 Optimizations to address these issues

Optimizations

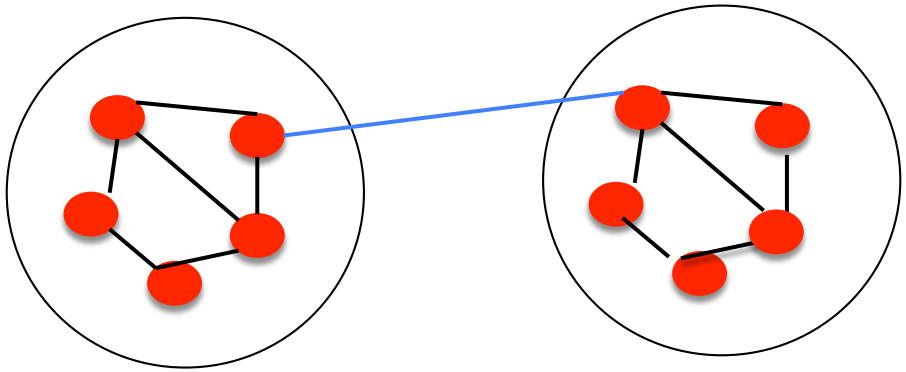
- *Soft-failure detection*
  - Re-transmission only if solicitation message is received in the same gossip round.
  - Indicates process or link failure. High chances of recovery using other healthy links
- *Round retransmission limit*
  - Retransmission limited to some maximum amount of data
  - Spreads the overhead spatially and temporally
- *Cyclic retransmissions*
  - No retransmission of message if the same message was transmitted in the previous round to the same partner
  - Avoid redundancy

Optimizations

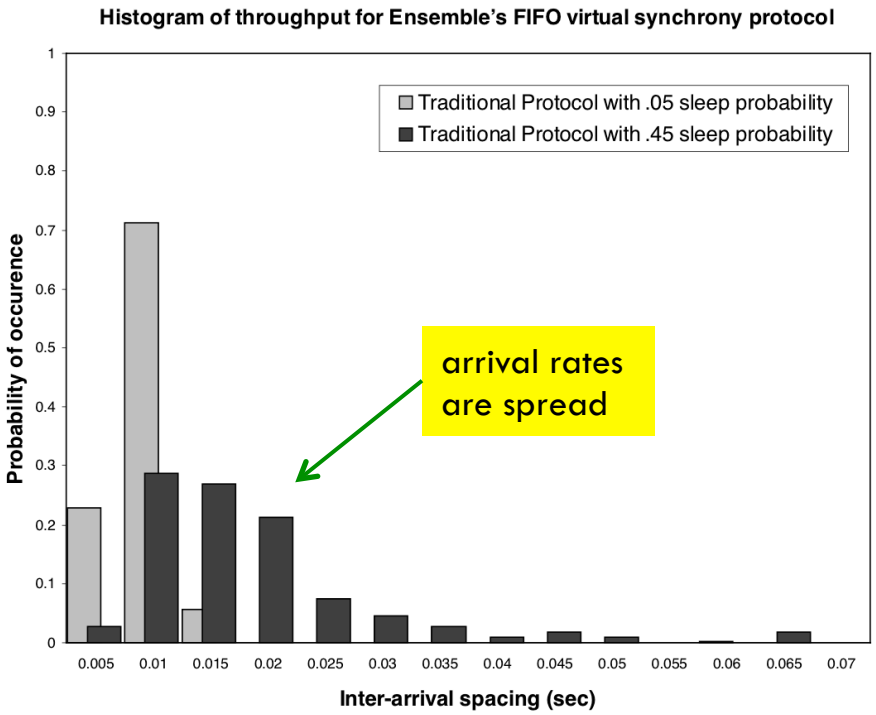
- *Independent numbering of rounds*
  - Each process manages its own round numbers
  - round number used to take delivery or garbage collection decisions, which are local
- *Multicast for some retransmissions*
  - If a message is requested twice, the process multicasts.
- *Most-recent-first retransmission*
  - Solicitation message is send for the most recent message
  - Avoids scenarios in which faulty process is unable to catch up and hence lags behind the group

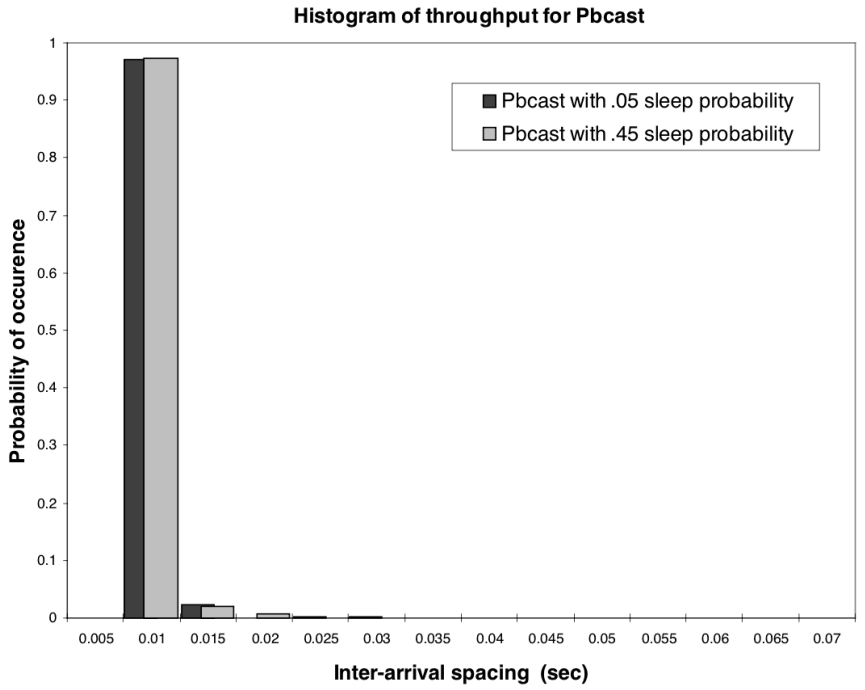
Optimizations

- *Hierarchical gossip for scalability*
  - full membership information needed – scalability issue for large-scale groups
  - Communication over WAN



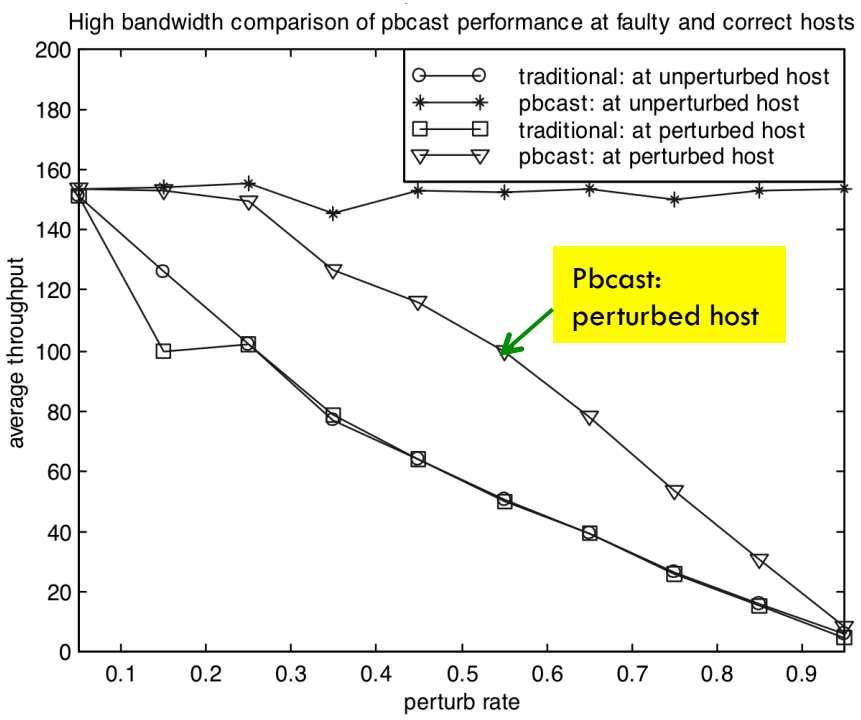
Experimental Results

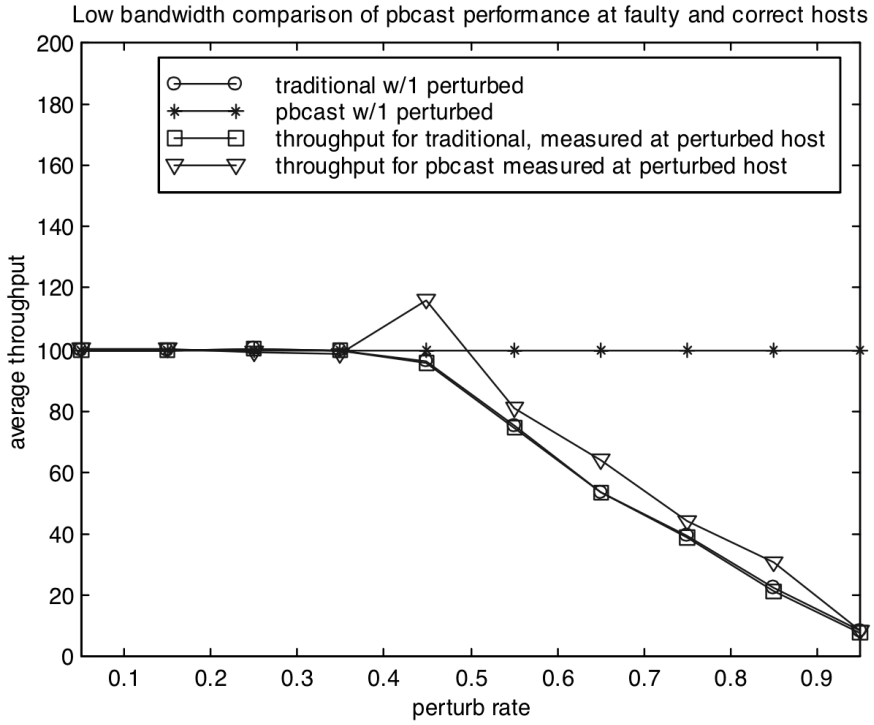




Group count: 8 processes; 75 7KB multicast per second;
1 process put to sleep with some probability. Perturb rate not specified

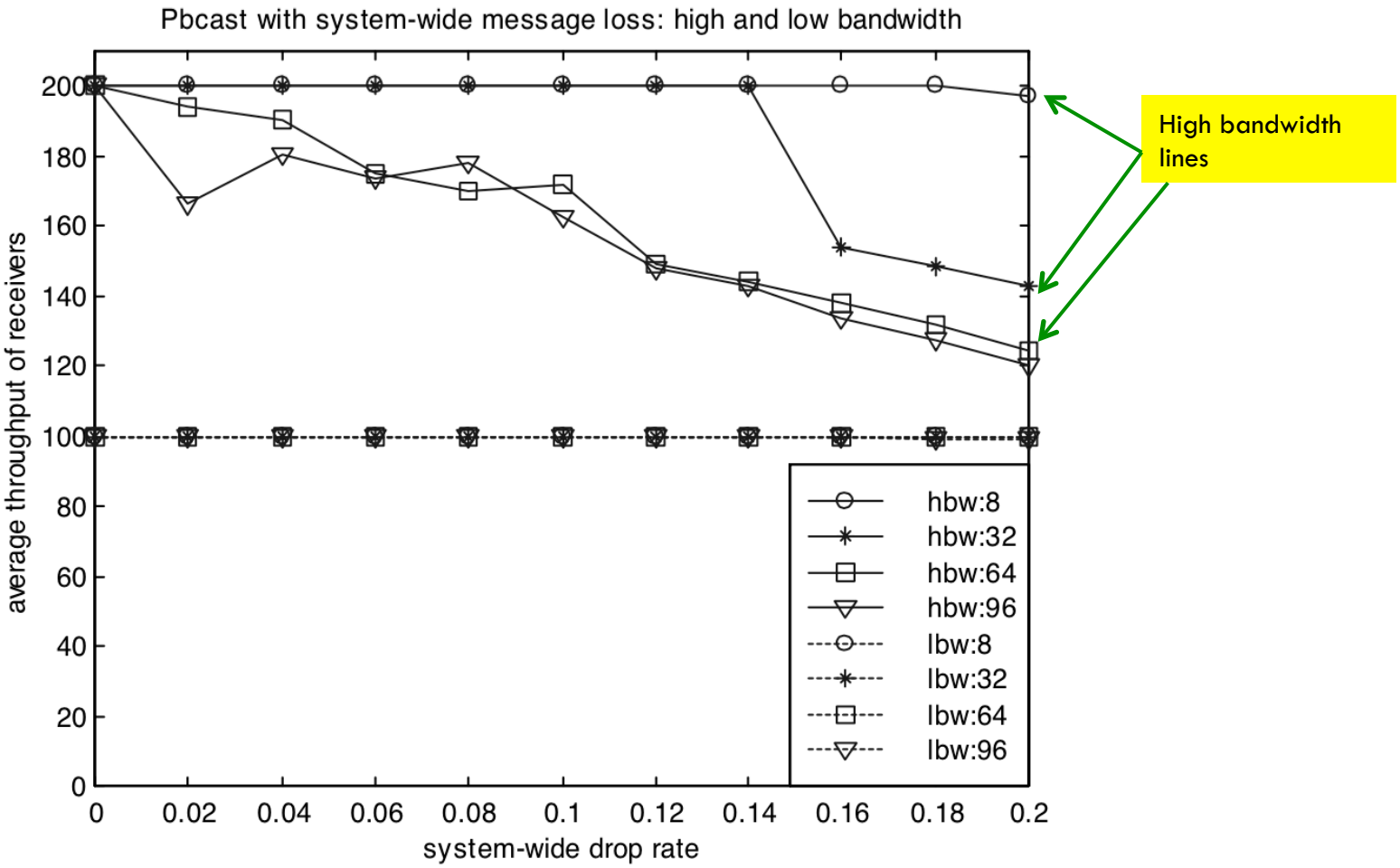
Experimental Results



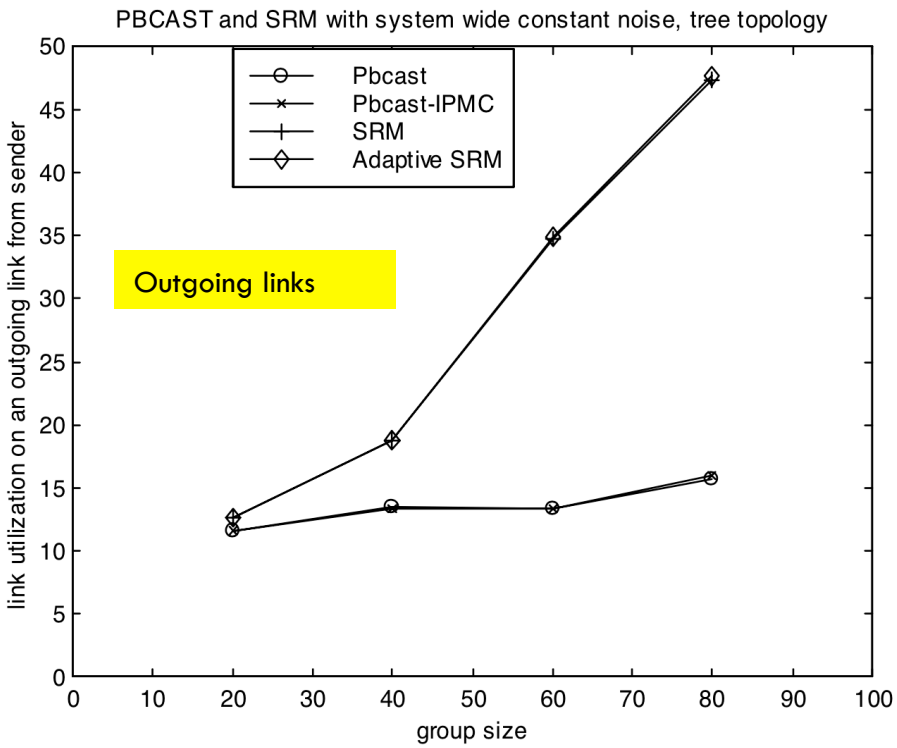


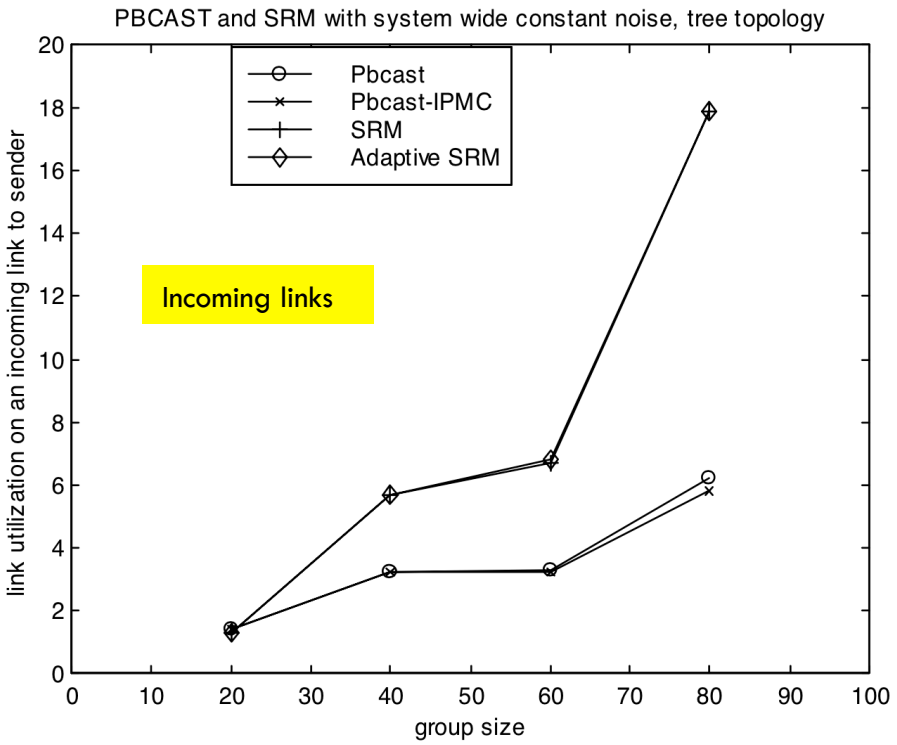
Group count: 8 processes; 150 & 100 7KB multicast per second;

Experimental Results



Experimental Results





NS2 simulations; 100 210-byte messages

Conclusion

- Pbcast provides bimodal delivery guarantee in realistic network environments
- Pbcast is scalable and gives stable throughput
- Ideal for applications that can tolerate some degree of message loss
  - Stock market updates
  - Air traffic control
  - Medical telemetry
  - Streaming multimedia

Discussion Points

- Is the protocol really scalable?
- Support for identifying Byzantine failures
- Most recent first transmission – too conservative?
- Dynamic adjustment of control parameters