

# Message Stability Detection for Reliable Multicast

Katherine Guo

Bell Laboratories

kguo@bell-labs.com

Injong Rhee

North Carolina State University

rhee@csc.ncsu.edu

# Introduction

- Reliable multicast protocols using the local repair scheme: SRM, RMTP, LBRM, LMS, Search Party.
- To detect when to delete a packet from the buffer.
  - Message stability detection
  - Membership failure detection
- Message is *stable* when it is received by all members in the group.

# Outline

- Assumptions
- Gossip protocol for stability detection and failure detection
- Analysis
- Protocol with unknown group membership
- Simulation
- Conclusion

# Assumptions

- Messages may get lost.
- Processes may crash.
- Group size is  $n$ .
- (sender id, sequence number): unique name for data.
- Do not reply on reliable multicast protocols.
- Membership *known or unknown*.

## Information at each member

at member **A**:

- |                               | <b>A</b> | <b>B</b> | <b>C</b> | <b>D</b> |
|-------------------------------|----------|----------|----------|----------|
| • <i>Sequence number</i>      | $R = [1$ | $2$      | $4$      | $2]$     |
| • <i>Whom-I've-heard-from</i> | $W = [1$ | $1$      | $1$      | $0]$     |
| • <i>Min-so-far</i>           | $M = [1$ | $1$      | $1$      | $2]$     |
| • <i>Stability</i>            | $S = M$  |          |          |          |
- when  $W$  is filled with all 1's

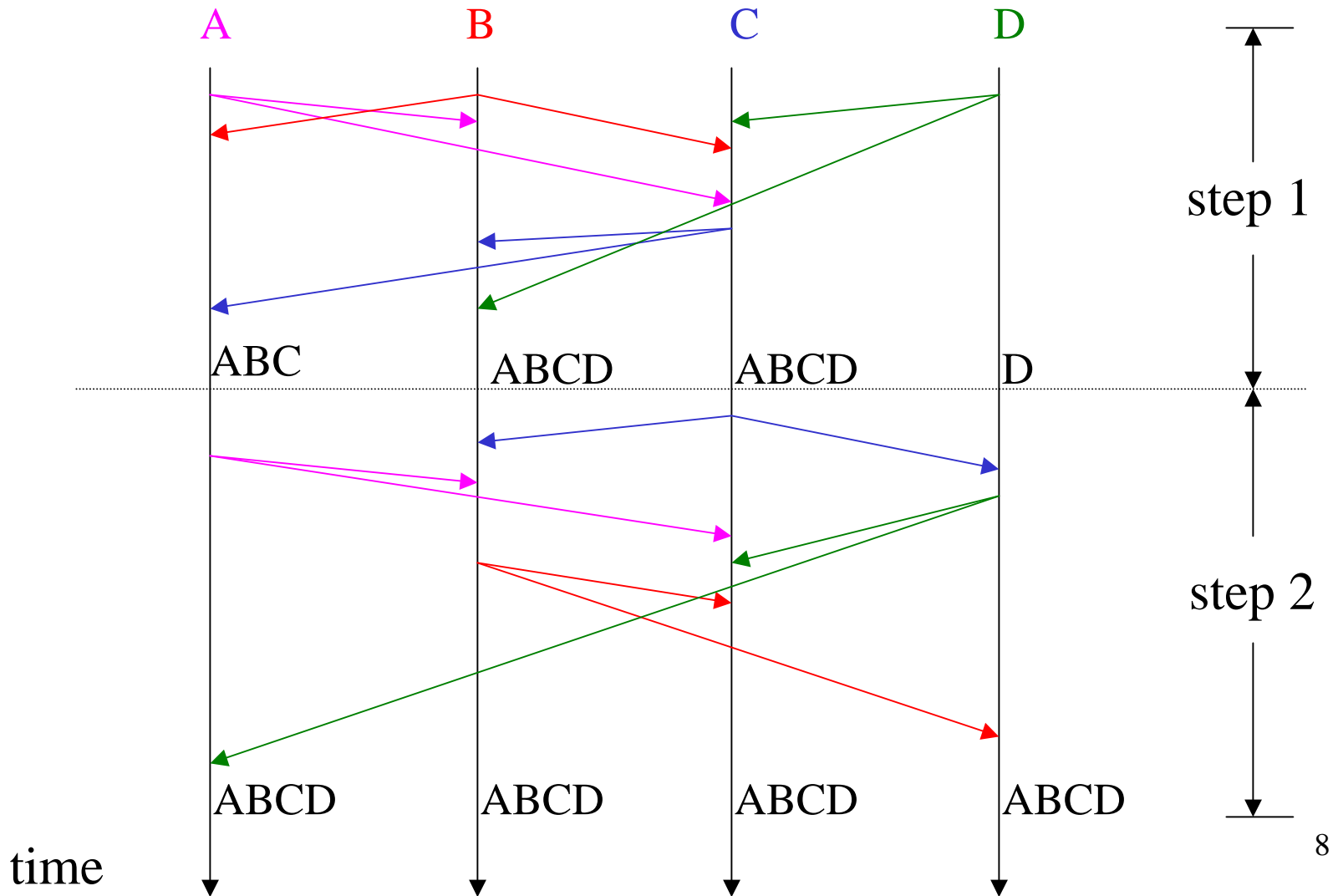
## Stability detection with gossiping

- Equally timed steps.
- Each member sends info to random gossip subset of size  $b$ .
- Starting points are scattered randomly.

## Stability detection with gossiping

- Arrays at each member:
- $R$ : Sequence number array
- $M$ : Min-so-far array
- $W$ : Whom-I've-heard-from bitmap array
- $S$ : Stability array
- Start:  $M := R$ ;
- Each time step: sends  $(M, W)$  to its gossip subset
- Upon receipt of a gossip message  $(M', W')$
- $M := \text{Ele\_Min}(M, M')$ ;  
 $W := \text{Ele\_Max}(W, W')$ ;
- $S := M$  When  $W$  has all 1's.  
Multicast  $S$  or piggyback  $S$  on future gossip messages.

# Sample run of stability detection



## Failure detection

- Information at each member (e.g. **A**)

**A B C D**

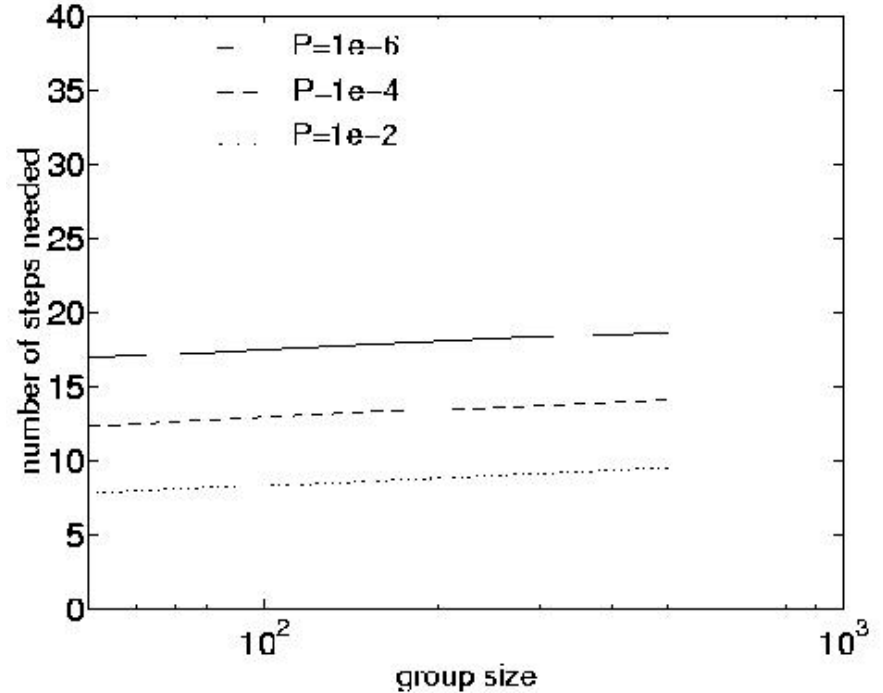
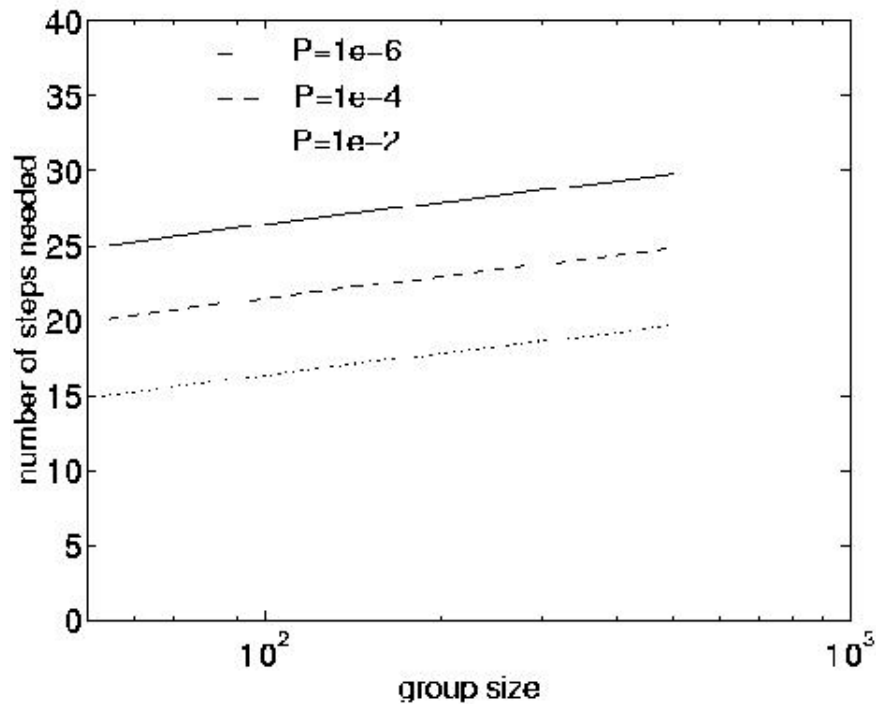
*Live*  $L = [ 1 \ 1 \ 0 \ 1 ]$

*Timer*  $T = [ a \ a \ 0 \ a ]$

- $T[i] = a$  initially, and when a message is received from  $i$ .
- $L[i] = 1$  if  $T[i] > 0$

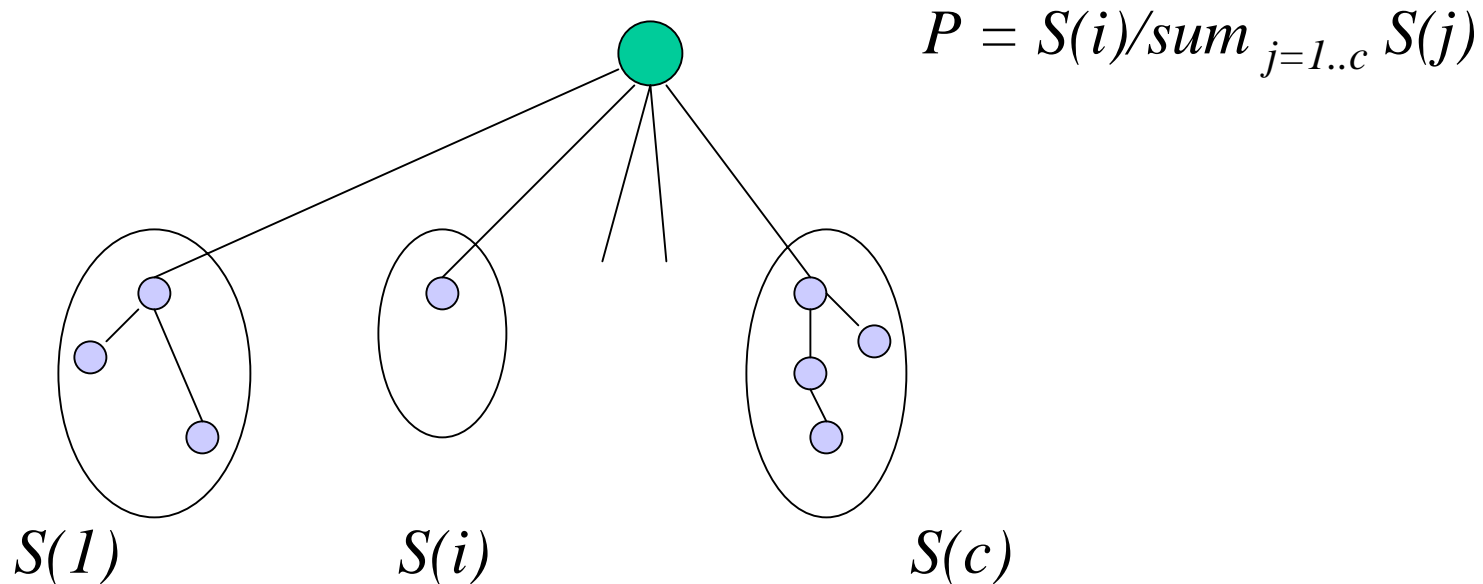
# Analysis for stability detection time

- Step interval & subset size  $b$
- Prob(*incomplete* stability detection)



# Modification with unknown membership

- Multicast router support:
- Use estimation of number of steps

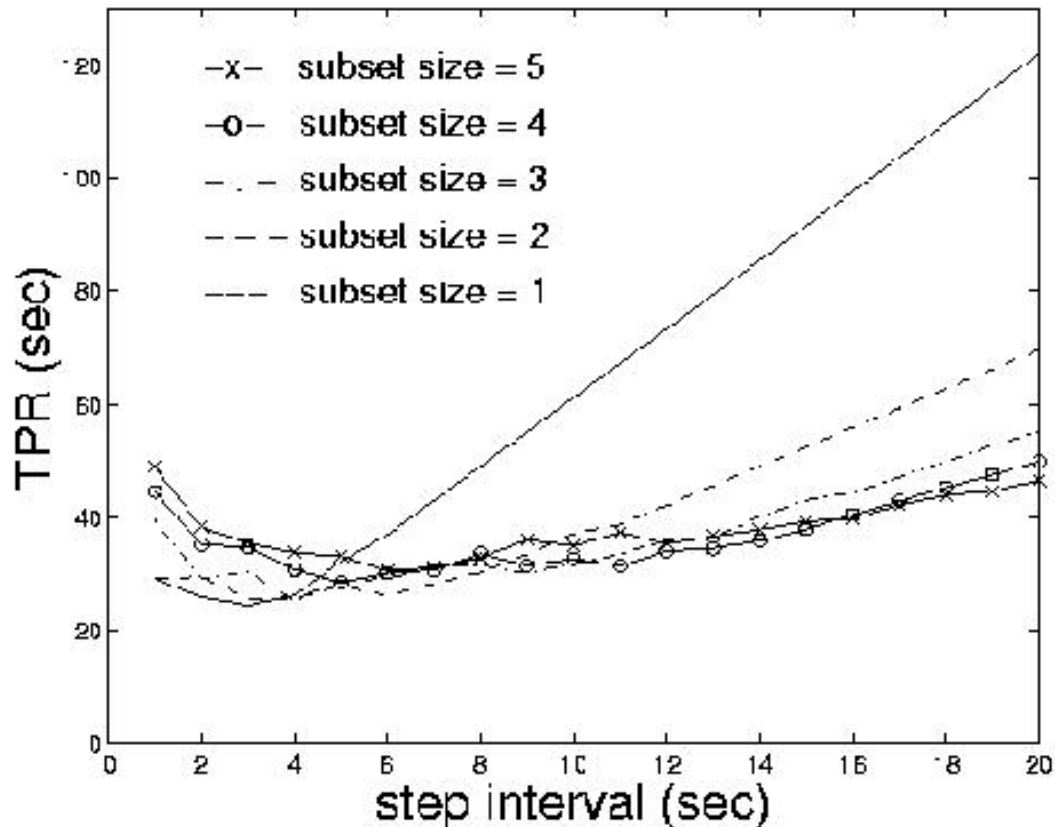


# Simulation setup

- NS simulator, bounded degree tree of size 1000.
- Performance metric
  - Time Per Round ( $TPR$ )
  - Time To Stable ( $TTS$ )
    - $TPR$
    - Frequency to trigger protocol:  $F$  seconds
    - Reliable Multicast protocol:  $D$  seconds
    - Maximum  $TTS = F + D + TPR$

# Simulation results

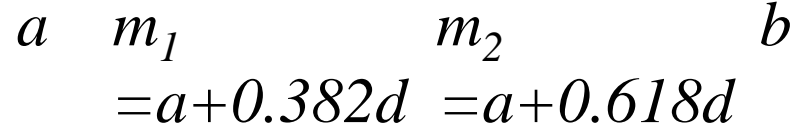
- Time Per Round for fixed group size 200



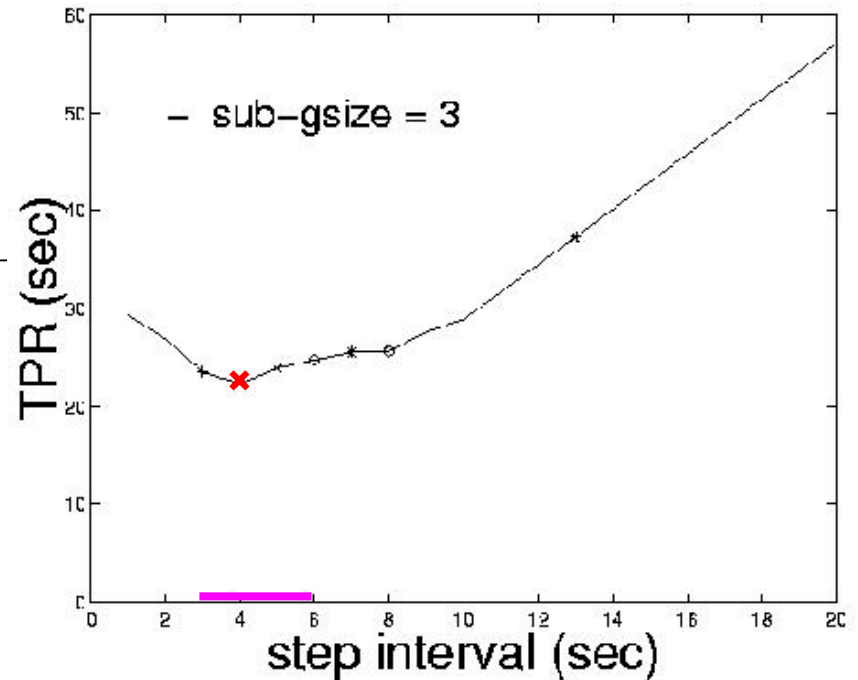
# Adaptive method

To find the window of optimal step intervals

- find a near-min  $TPR$   
(golden-ratio method)

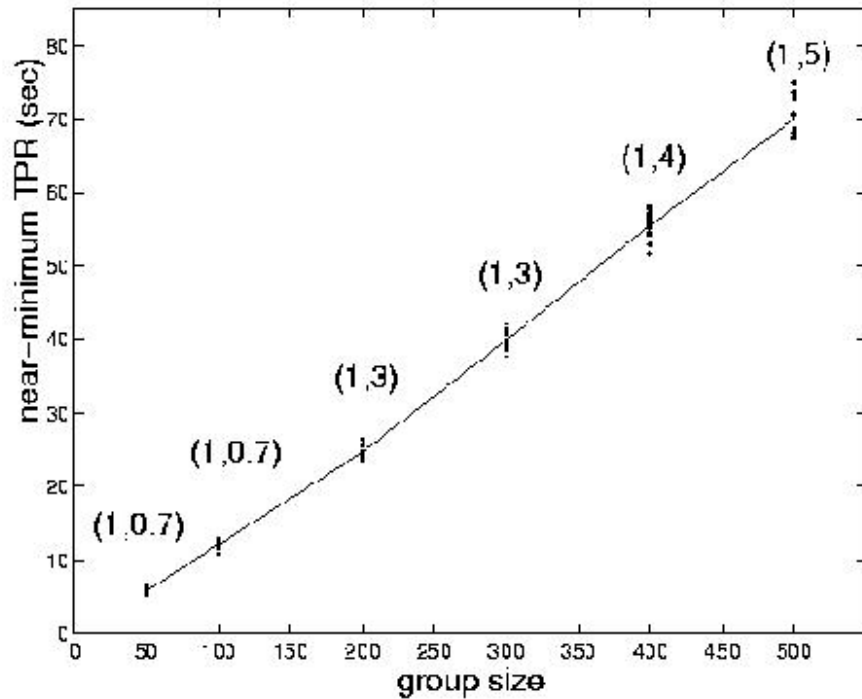


- find the optimal step interval window  
 $s, s+k, s+(1+2)k,$   
 $s+(1+2+4)k, s+(1+2+2)k, \dots$

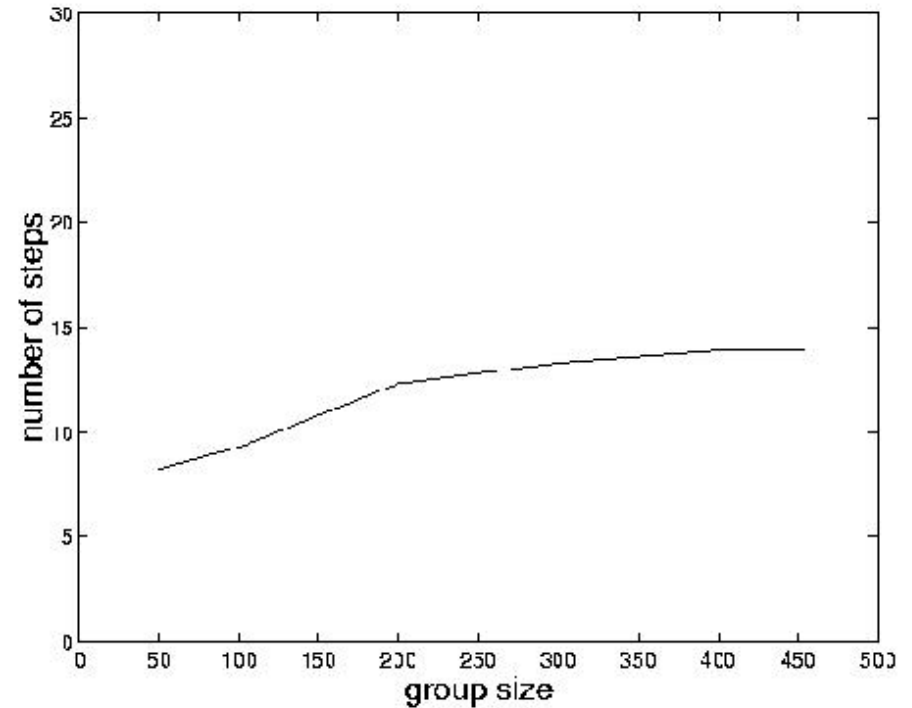


# Different group sizes

TPR:  $O(n \log n)$



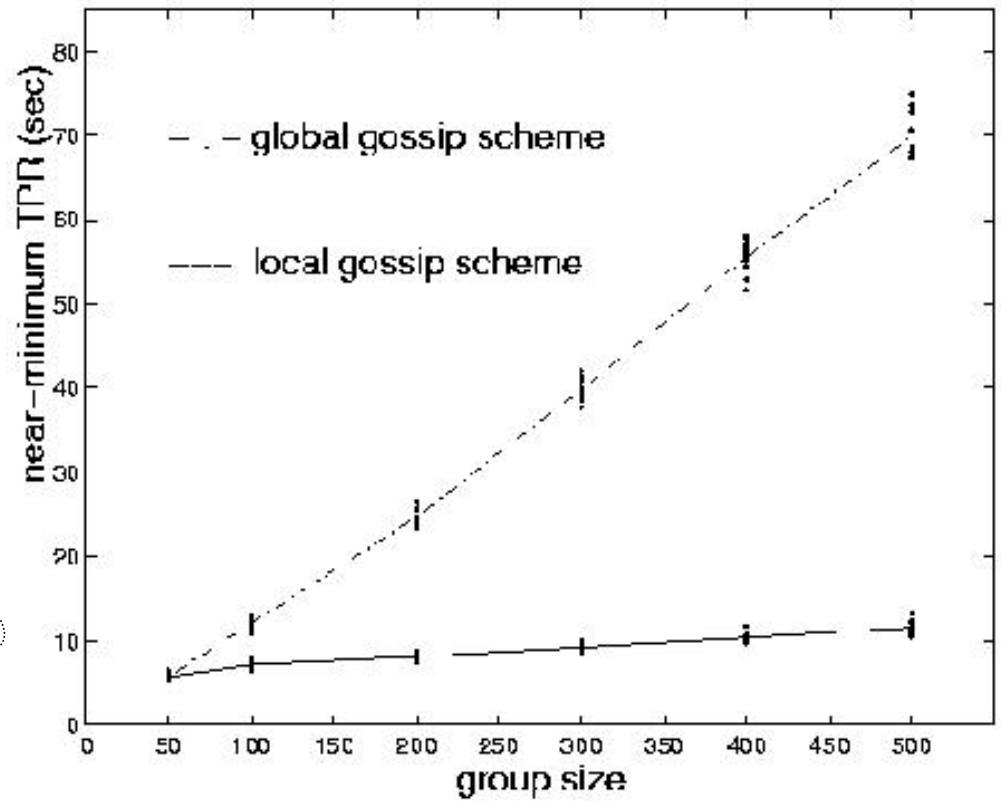
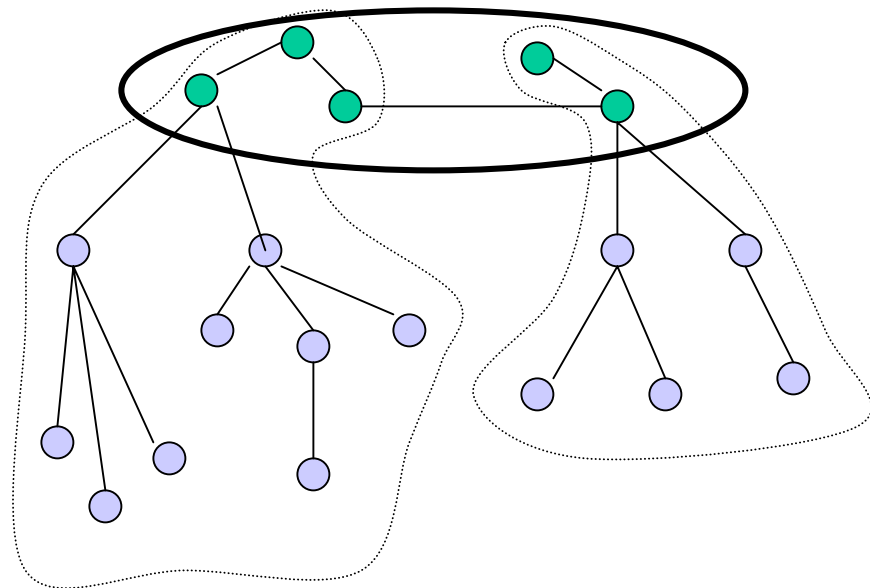
number of steps:  $O(\log n)$



# Local gossip

● stability controller

● local member



# Conclusion

- Fault Tolerant
  - message losses
  - membership changes
- Scalable
  - each member sends a fixed number of messages, total  $O(n)$
  - message size is less than  $O(n)$
  - no implosion
  - membership change does not affect operation