Analysis of Durability in Replicated Distributed Storage Systems

Sriram Ramabhadran and Joseph Pasquale

Dept. of Computer Science and Engineering
University of California, San Diego

IPDPS 2010

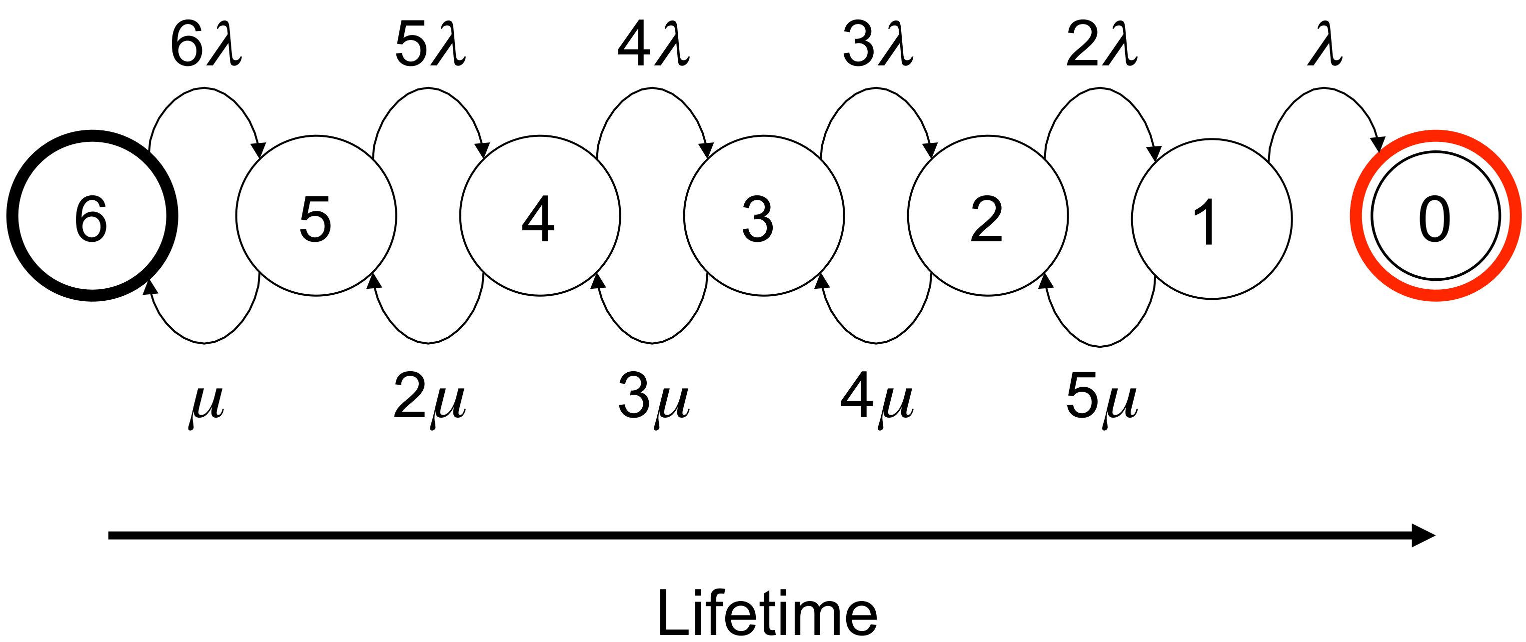
Replication

- Generate/regenerate enough replicas of object so it lives long enough
- Time to repair may be long
  - Must detect failure, and then regenerate
  - Repair process itself may fail
  - Have enough so we never run out
- *Uncertainty whether node is down or dead*

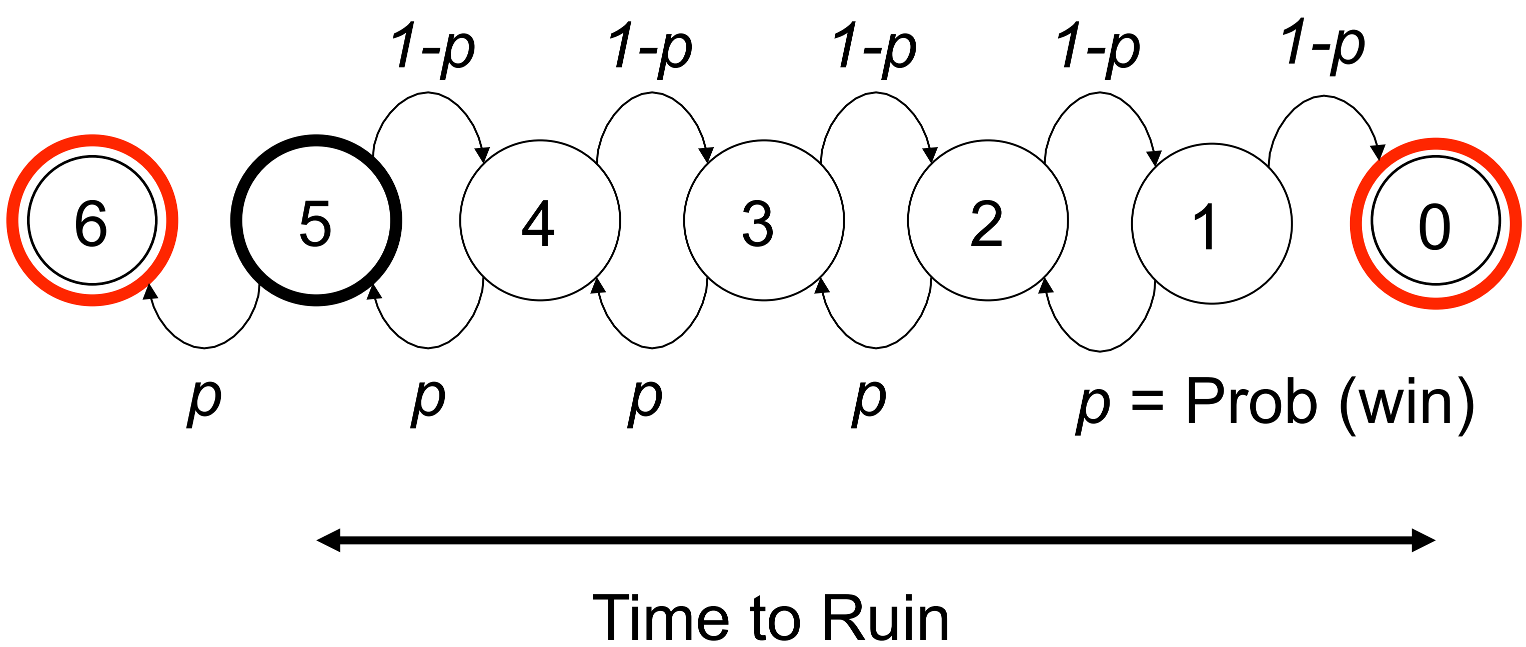
Formulation

- Given expected
  - Failure rate, λ
  - Repair rate, μ
  - Target time to live, L
- How many replicas N to achieve L?
- Note: *Repairs triggered by failures*

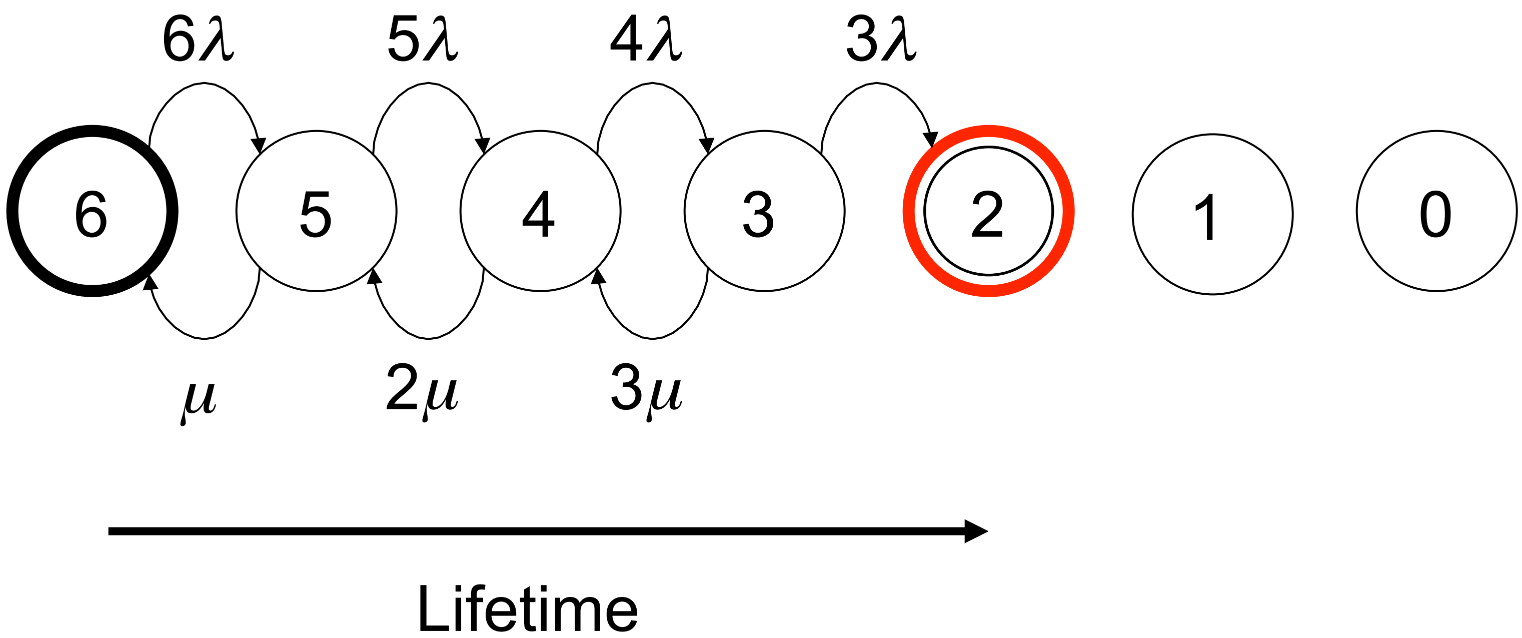
Model For 6 Replicas



Similar to Gambler's Ruin



Erasure Coding: 3 Fragments

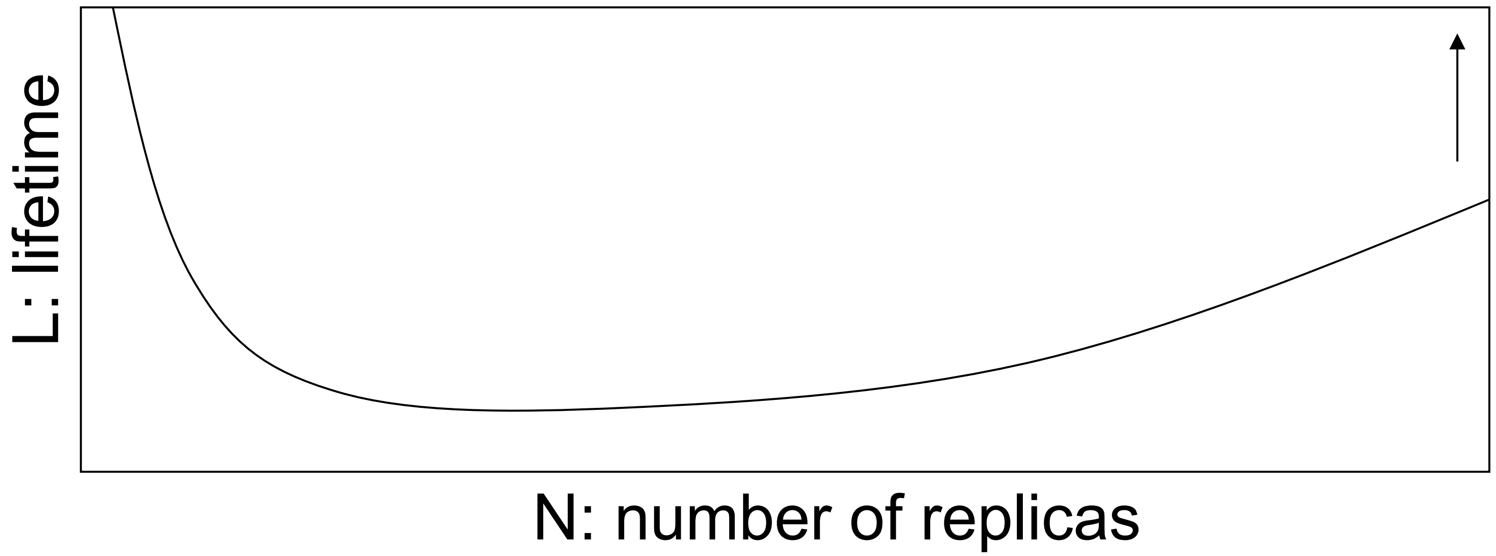


Result: Lifetime

$$L_N = \frac{1}{N\lambda} \sum_{i=0}^{N-1} \sum_{j=0}^{i} \frac{\binom{N}{j}}{\binom{N-1}{i}} \left(\frac{\mu}{\lambda}\right)^{i-j}$$

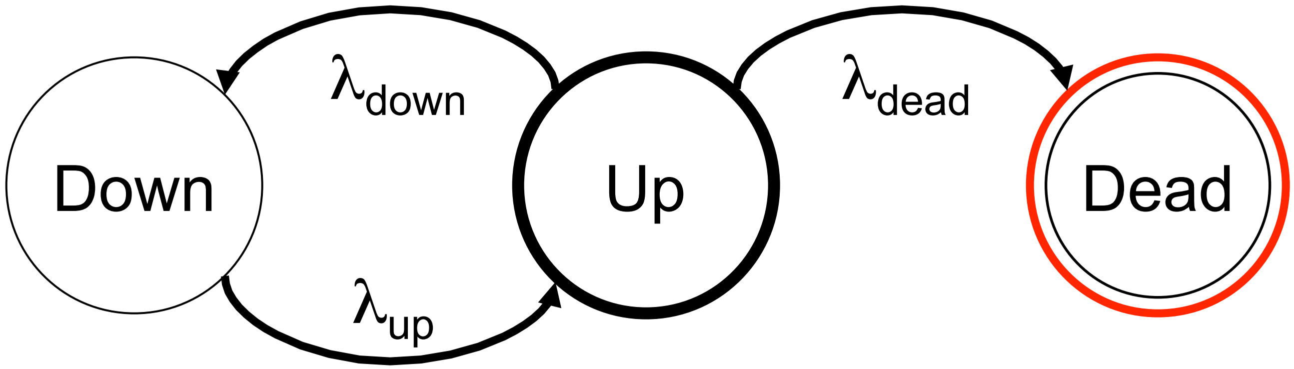
- Increases with N and μ (without bound)
- Question: Which is it better to increase?

Constrained Repair Bandwidth



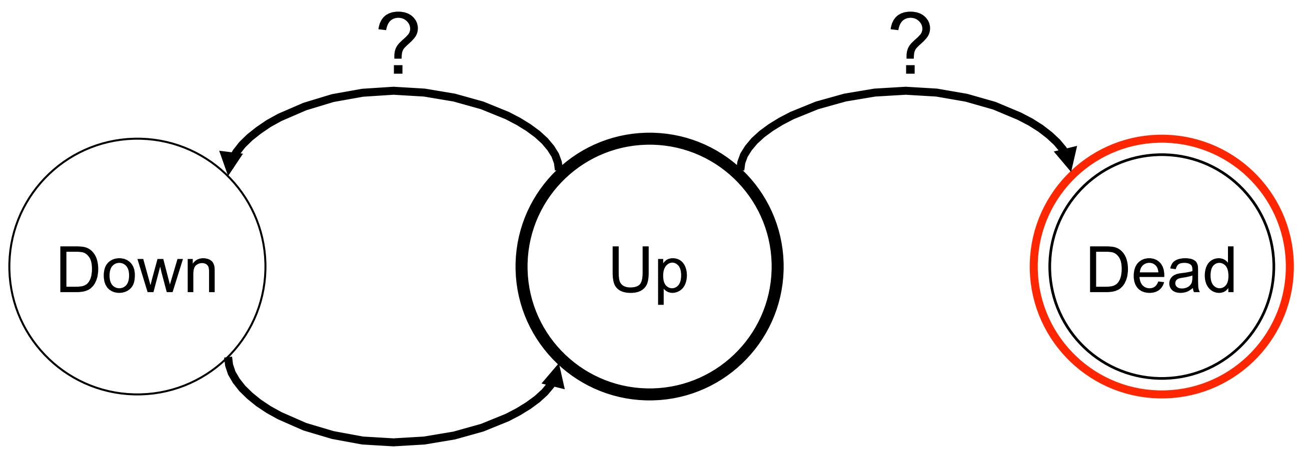
- Small N: aggressive repair
- Large N: minimal repair

Distinguishing Down / Dead



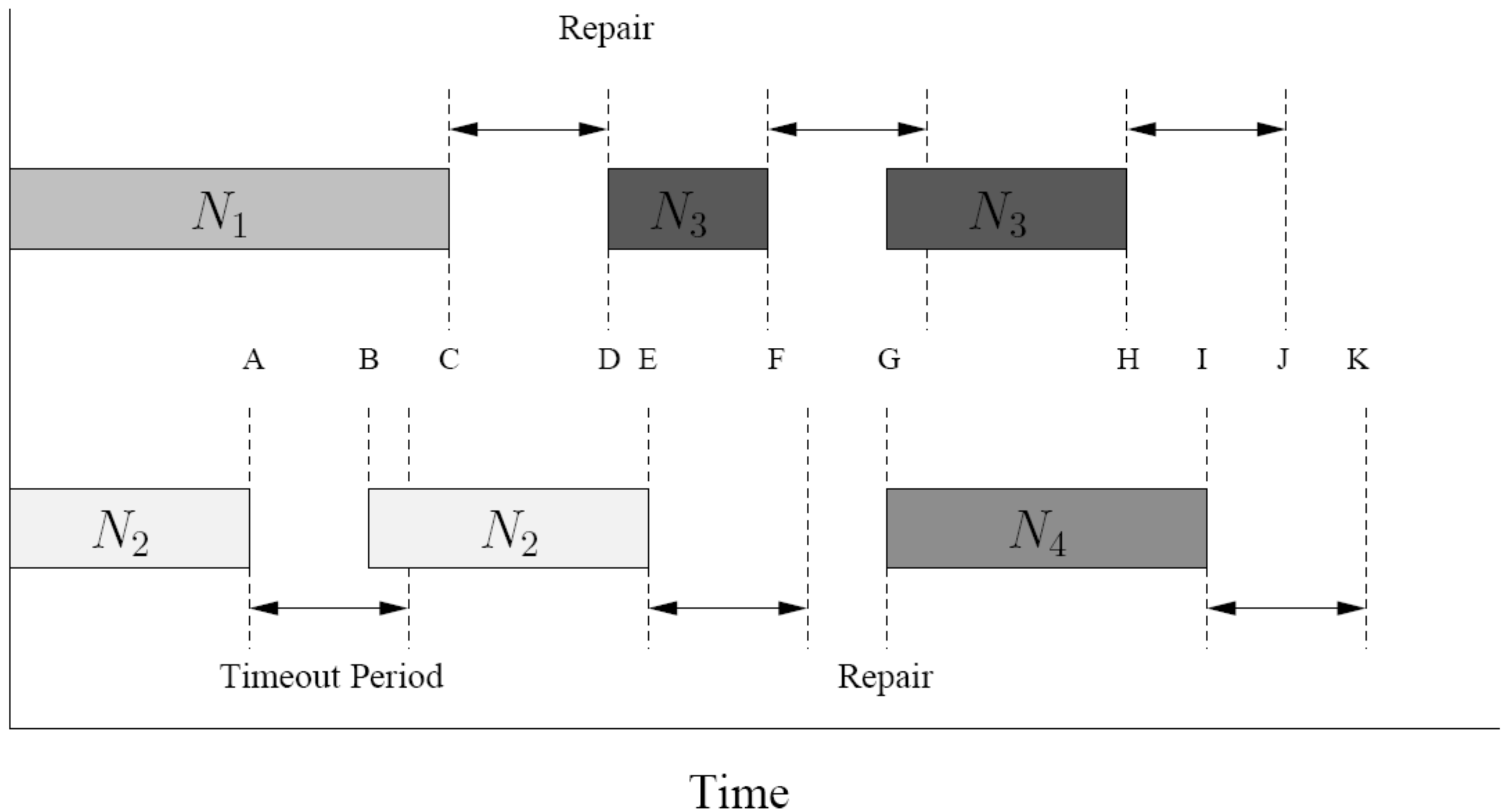
- Assumptions for single node, T>>d
  - lifetime ~ exp (T) $\lambda_{dead} = (u+d)/uT$
  - uptime ~ exp (u) $\lambda_{up} = 1/d$
  - downtime ~ exp (d) $\lambda_{down} = (T-u-d)/uT$

Uncertainty: Down or Dead

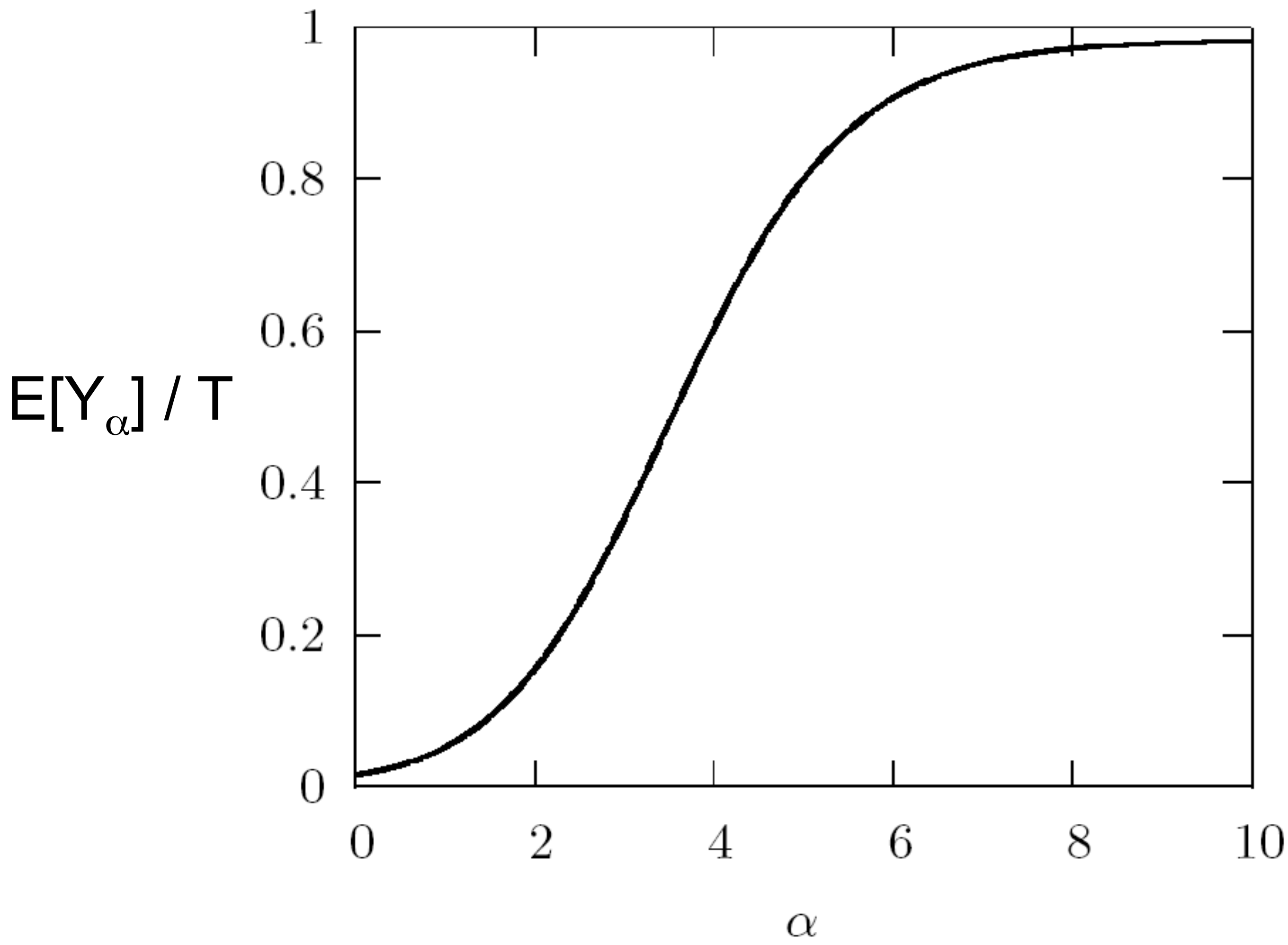


- Timeout α
  - If node not up after α, declare dead, repair
  - What value of α maximizes efficiency?

Ex: Number of replicas r=2

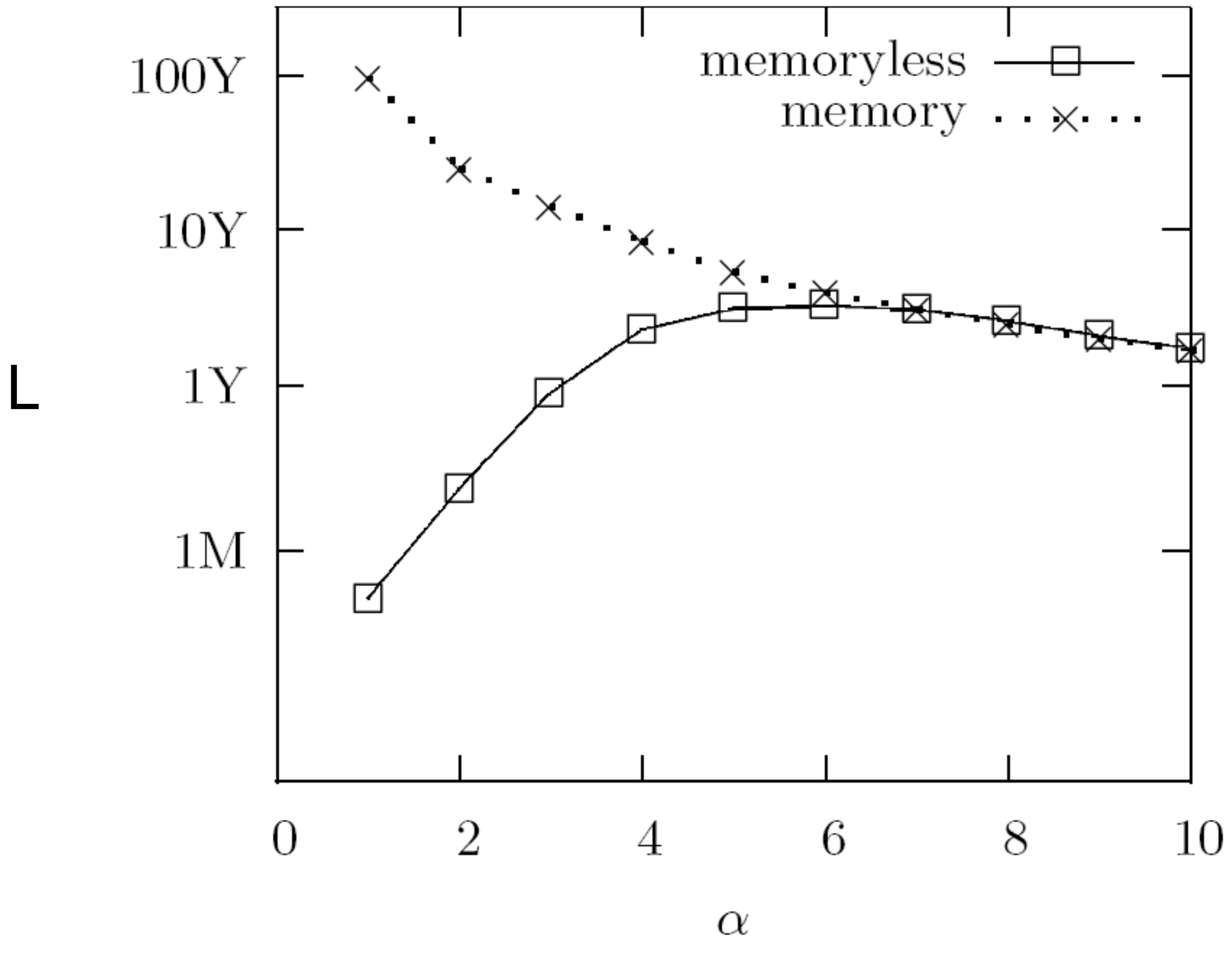


$E[Y_\alpha]$ = Replica Mean Timeout



u = 12 hours
d = 12 hours
T = 1 month
α in units of d

Lifetime L vs. α, r = 4



Conclusions

- There is an optimal timeout
- Can be determined by observing a single node
- Without-memory performs close to with-memory