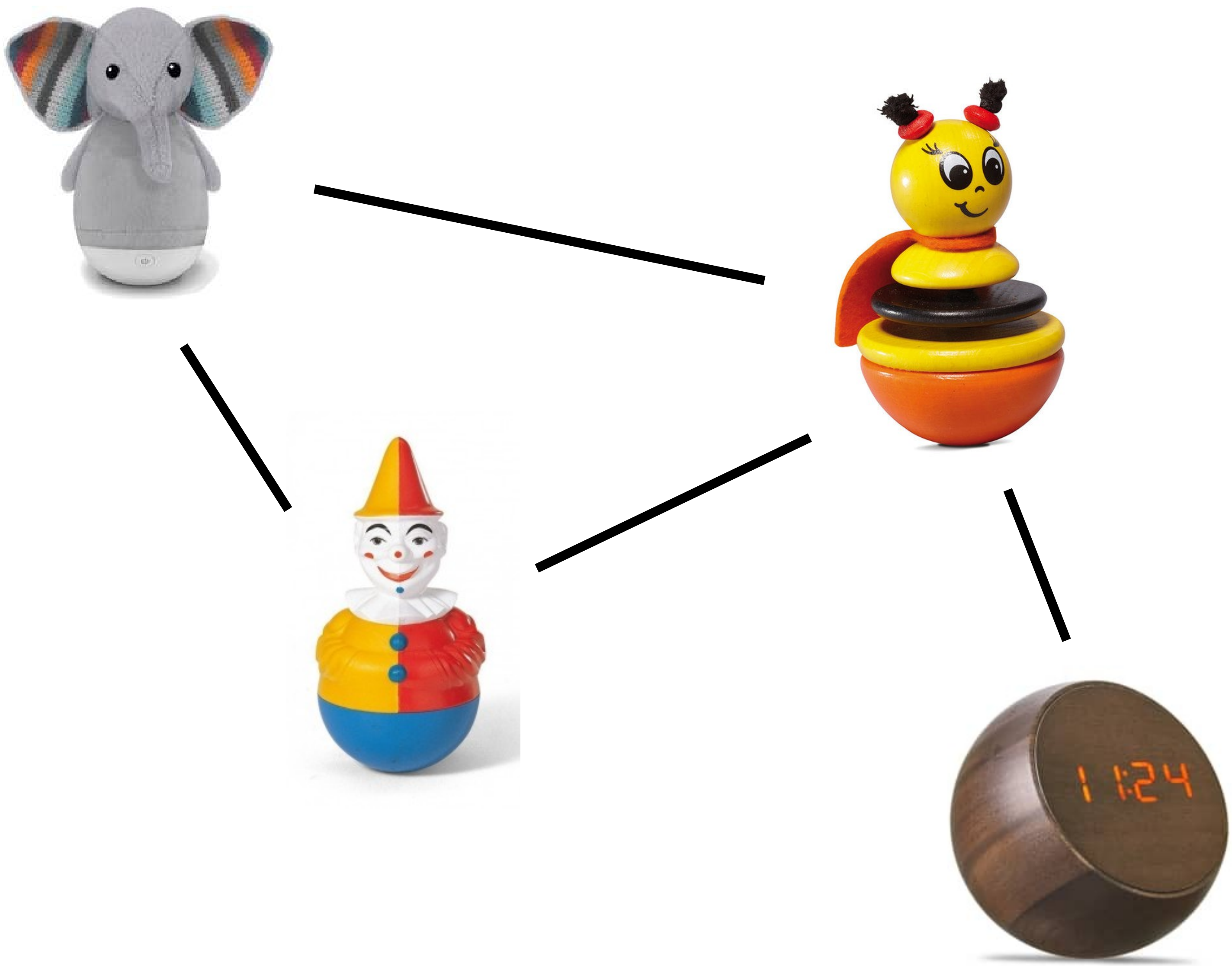
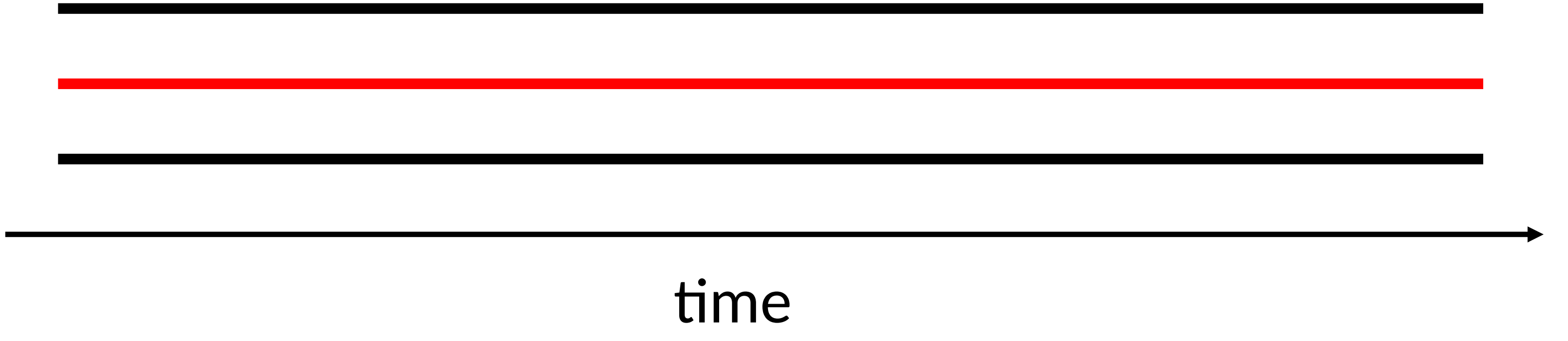
Self-Stabilization & Recovery



Model

TMP, but faults are not eternal any more!

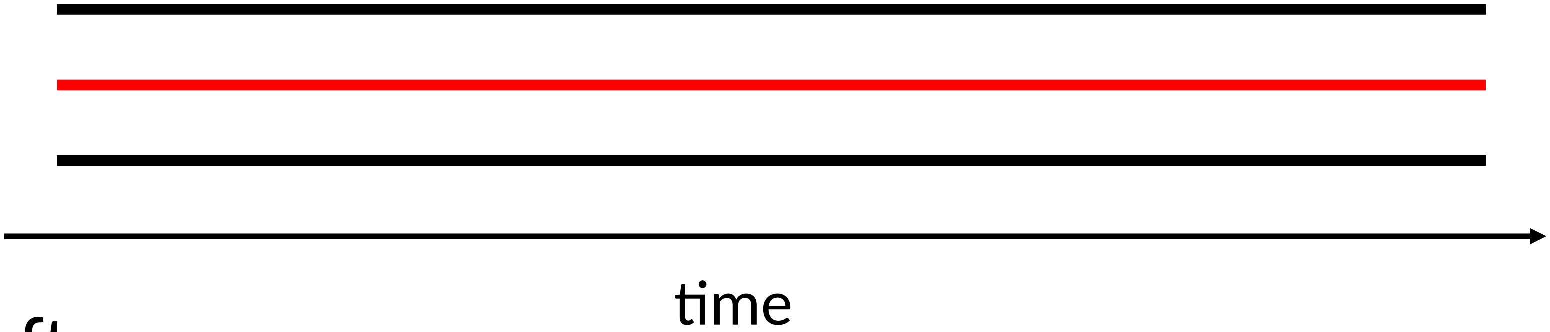
before:



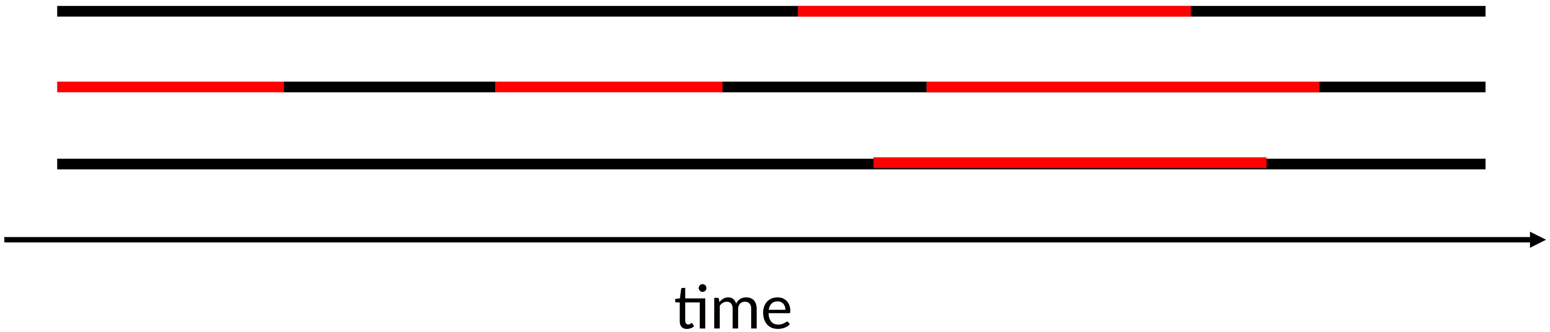
Model

TMP, but faults are not eternal any more!

before:

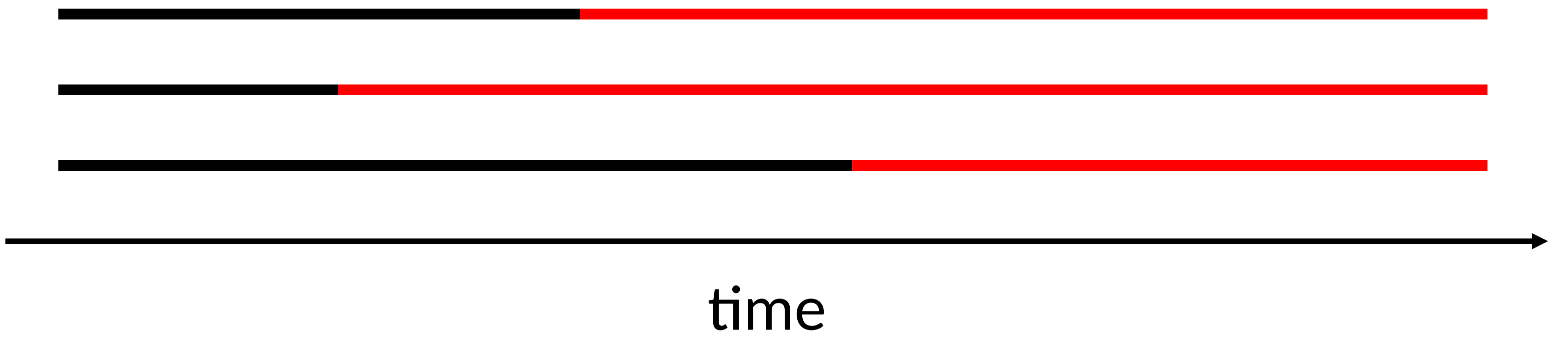


after:



What if Nodes Fail Randomly (in Time)?

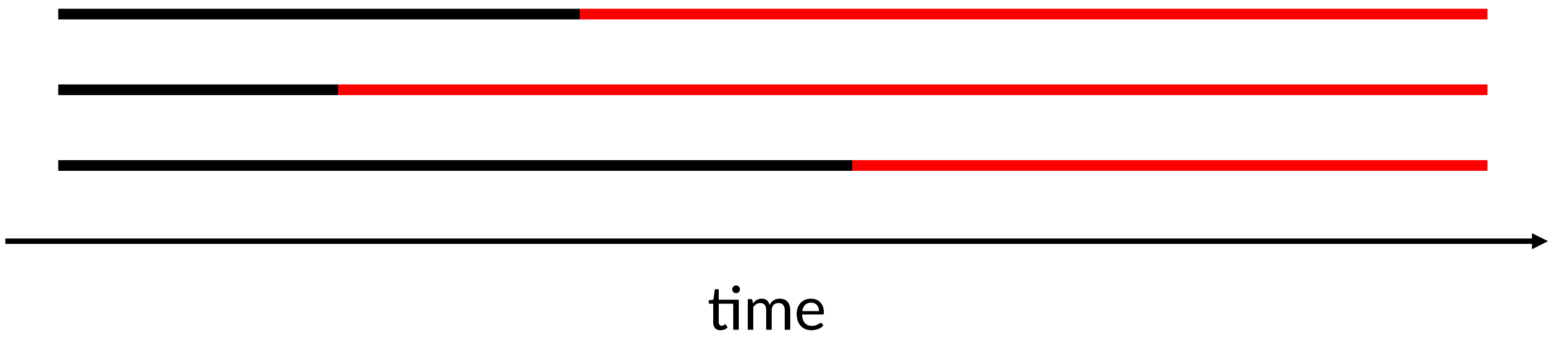
If there is no recovery mechanism:



System fails as soon as too many nodes failed!

What if Nodes Fail Randomly (in Time)?

If there is no recovery mechanism:



System fails as soon as too many nodes failed!

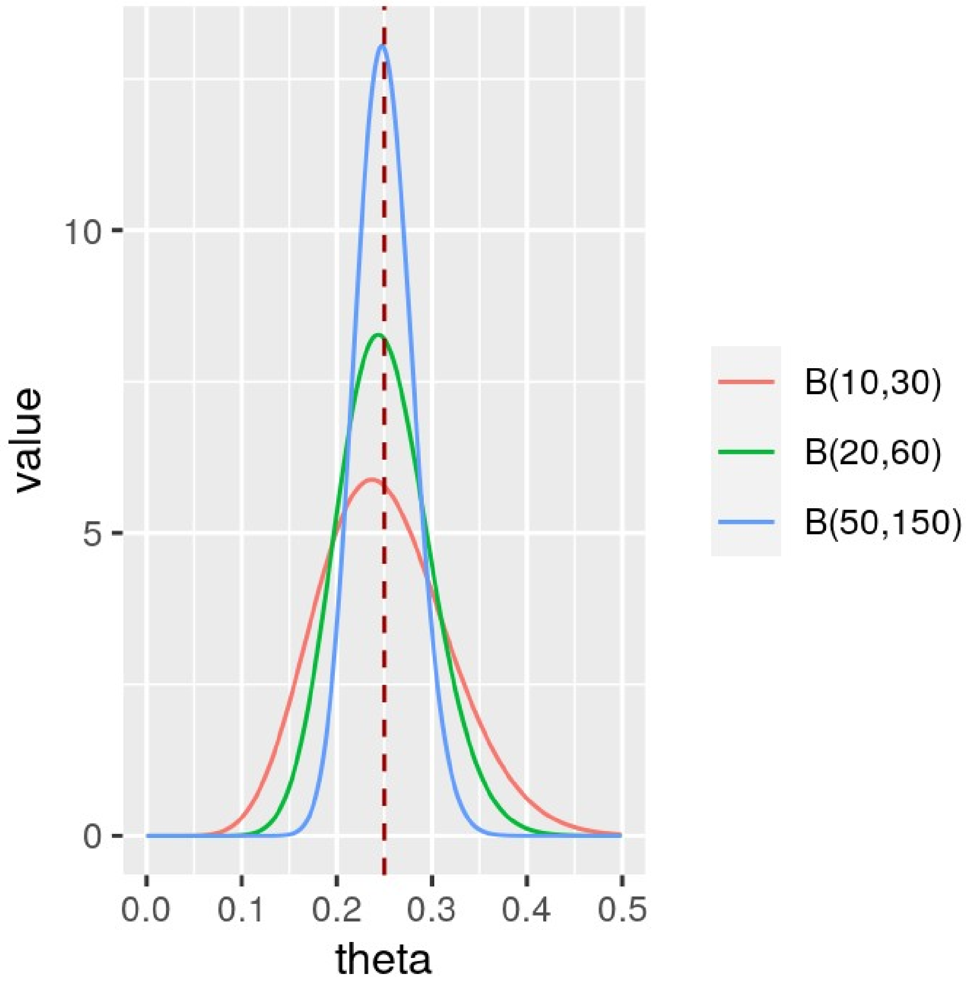
MBTF for error rate λ (single node):

$\int e^{-\lambda t}\, dt = -e^{-\lambda\infty}/\lambda - (-e^{-\lambda 0}/\lambda) = 1/\lambda$

What if Nodes Fail Randomly (in Time)?

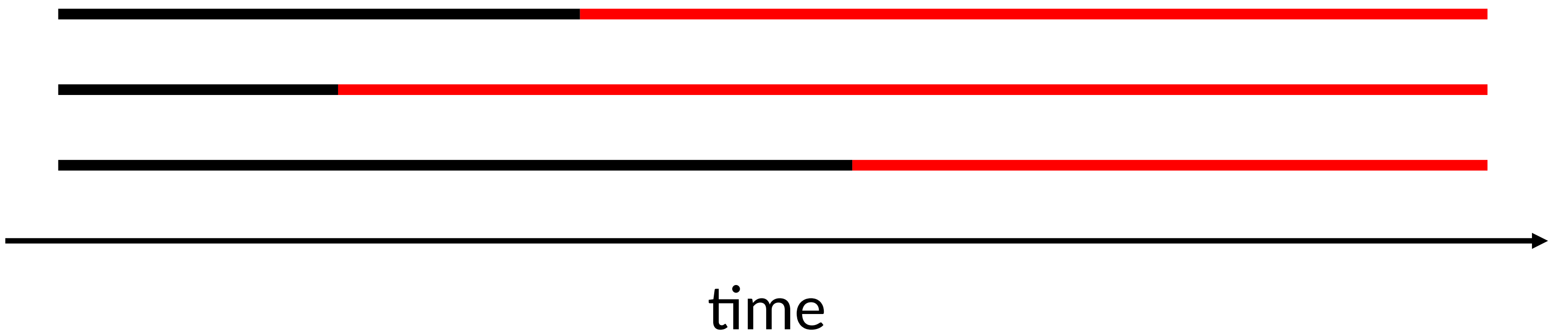
P[fail by time t]
= P[≤ 2/3 of nodes survive until time t]
= $F(2n/3;n,e^{-\lambda t})$,

where
F(k;n,p) =
P[n independent probability-p coins show ≤ k heads]



What if Nodes Fail Randomly (in Time)?

If there is no recovery mechanism:



System fails once too many nodes failed!

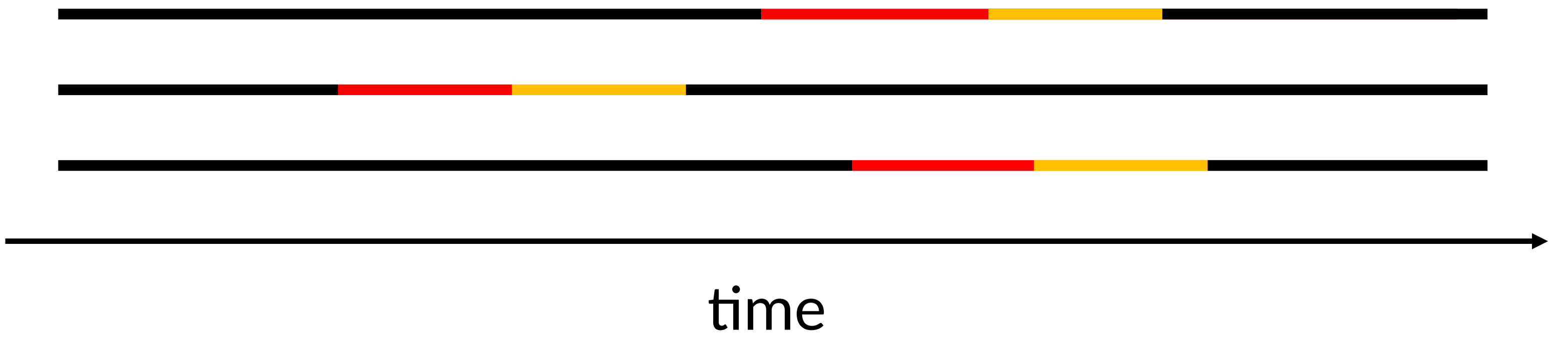
MBTF for error rate λ (n nodes, <n/3 faults):

$\int 1\text{-}F(2n/3;n,e^{-\lambda t})\,dt \rightarrow \ln(3/2)/\lambda < 1/\lambda$,

because $e^{-\lambda t} = 2/3 \Rightarrow t = \ln(3/2)/\lambda$

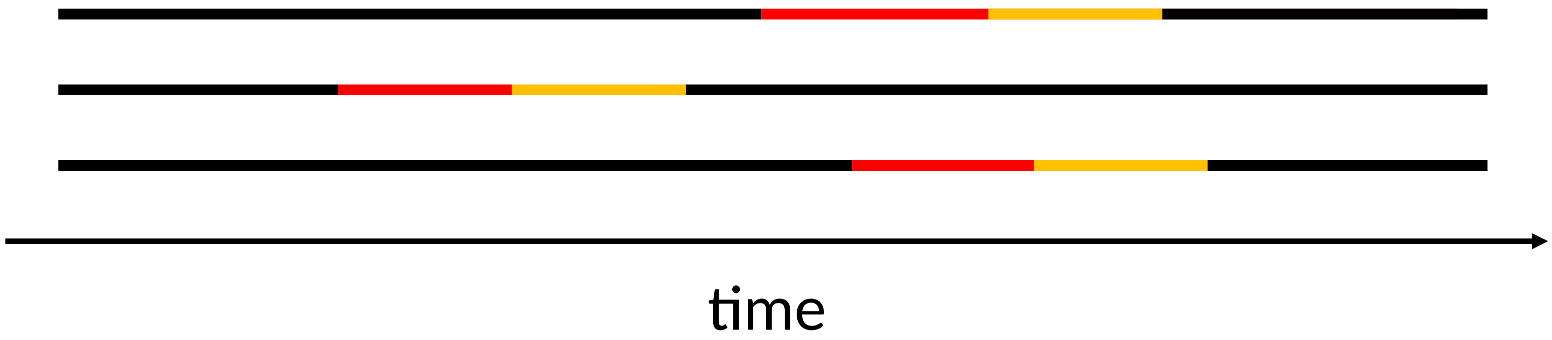
What if Nodes Fail Randomly (in Time)?

If we can ensure recovery within in time T:



What if Nodes Fail Randomly (in Time)?

If we can ensure recovery within in time T:



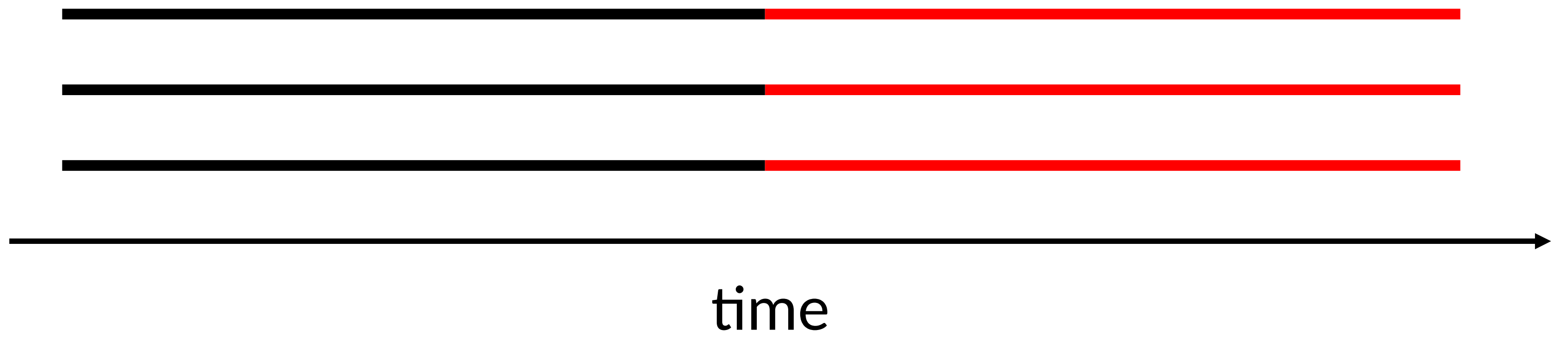
Balance tips at $1/\lambda = \theta(T)$ (T includes time for transient faults to end)

If $1/\lambda \ll T$: As $n \to \infty$, probability density for failing at any given time t tends to 0

If $1/\lambda \gg T$: prob. of fail state at time $t > 1/\lambda \to 1$

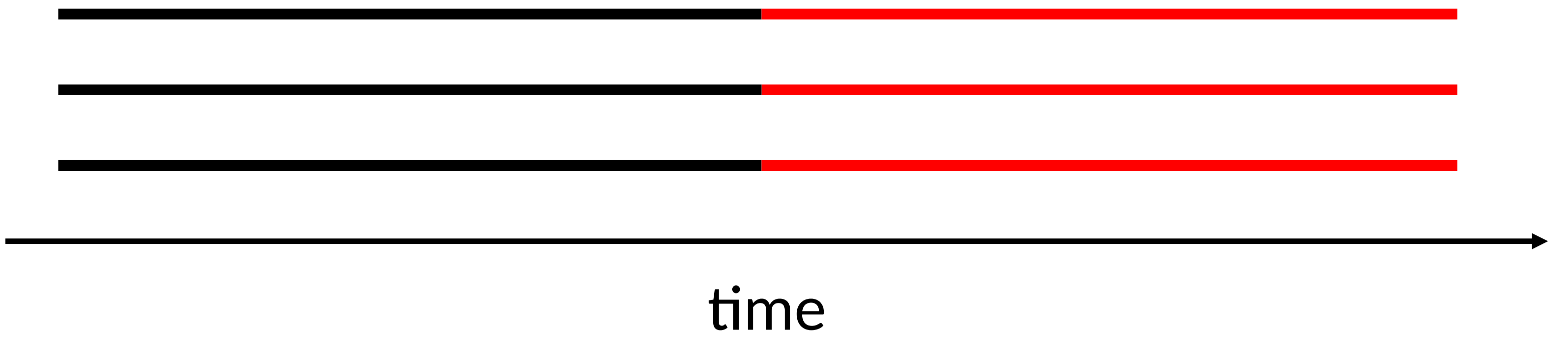
What if Nodes Fail Randomly (in Time)?

…but correlation still breaks things:



What if Nodes Fail Randomly (in Time)?

…but correlation still breaks things:

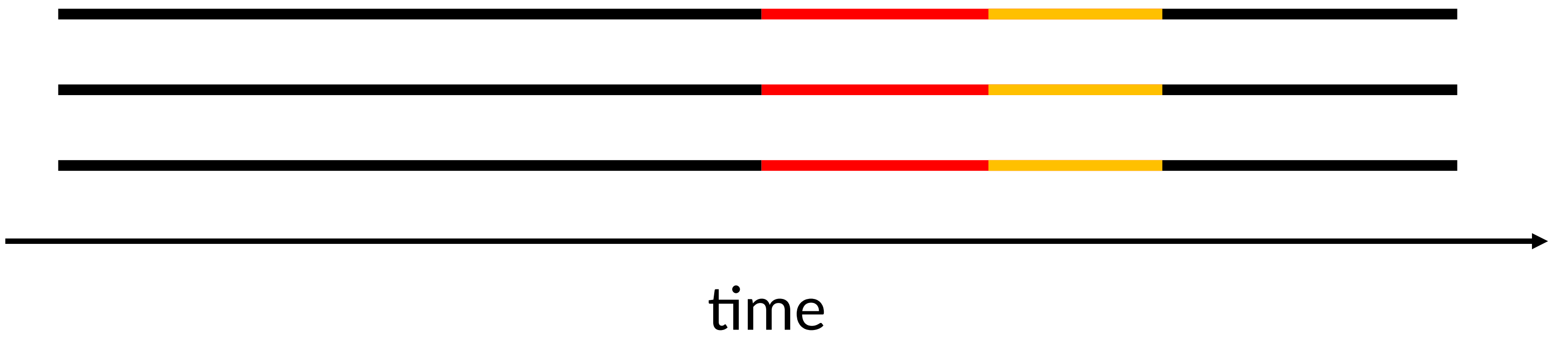


Failure rate λ per node, but all nodes fail together:

Equivalent to 1-node system!

What if Nodes Fail Randomly (in Time)?

…but correlation still breaks things:



Failure rate λ per node, but all nodes fail together:

Equivalent to 1-node system!

=> want recovery from arbitrary states!

This Chapter

today:

breakout session on selfstabilizing BFS tree construction

(self-stabilization: recovery from arbitrary transient faults ㄹㅌ getting a correct result from arbitrary initial state)

other sessions:

Up to you!

This Chapter

menu options for the other two sessions:

- Gradient Clock Synchronization (GCS):
  + overview of algorithm
  + proof of key lemmas & local skew
  + showing stabilization (unbounded time)
  + how unbounded skew breaks implementation
  + showing stabilization (bounded time)
- Lynch-Welch with recovery:
  + overview of algorithm
  + proof sketch
  + why it‘s not self-stabilizing

This Chapter

today:

breakout session on selfstabilizing BFS tree construction

(self-stabilization: recovery from arbitrary transient faults ㄹㅌ getting a correct result from arbitrary initial state)

other sessions:

Up to you!