Fraglets

Stochastic Programming for Provable Program Dynamics and Self-Healing Programs

or

”Programming by Equilibria”

Thomas Meyer, Christian Tschudin

Computer Science Department, University of Basel, Switzerland

15. Kolloquium Programmiersprachen und Grundlagen der Programmierung (KPS’09)

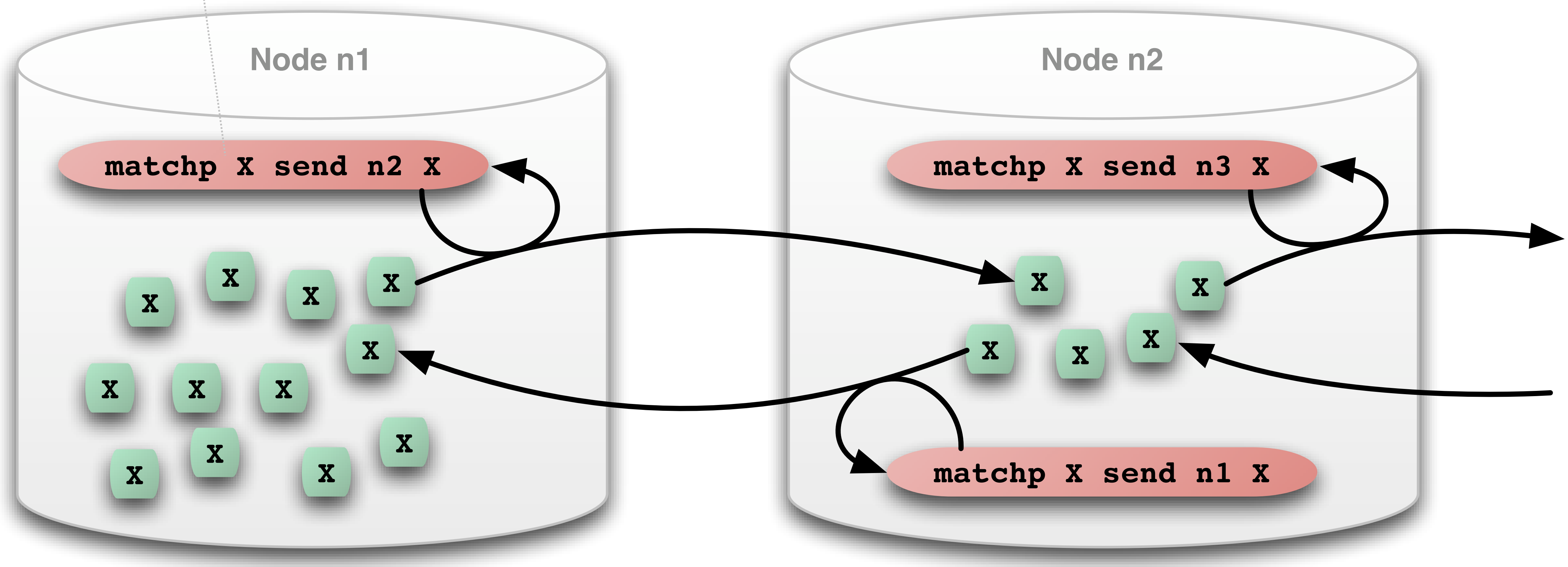
Maria Taferl, Austria, October 13th, 2009

Content

- Programming by Equilibrium
  Dynamical aspects of program execution in Fraglets
  - Stochastic reaction scheduler
  - Deterministic prediction of the program dynamics
- Self-Healing Programs
- Example: A Self-Healing Load Balancing Protocol

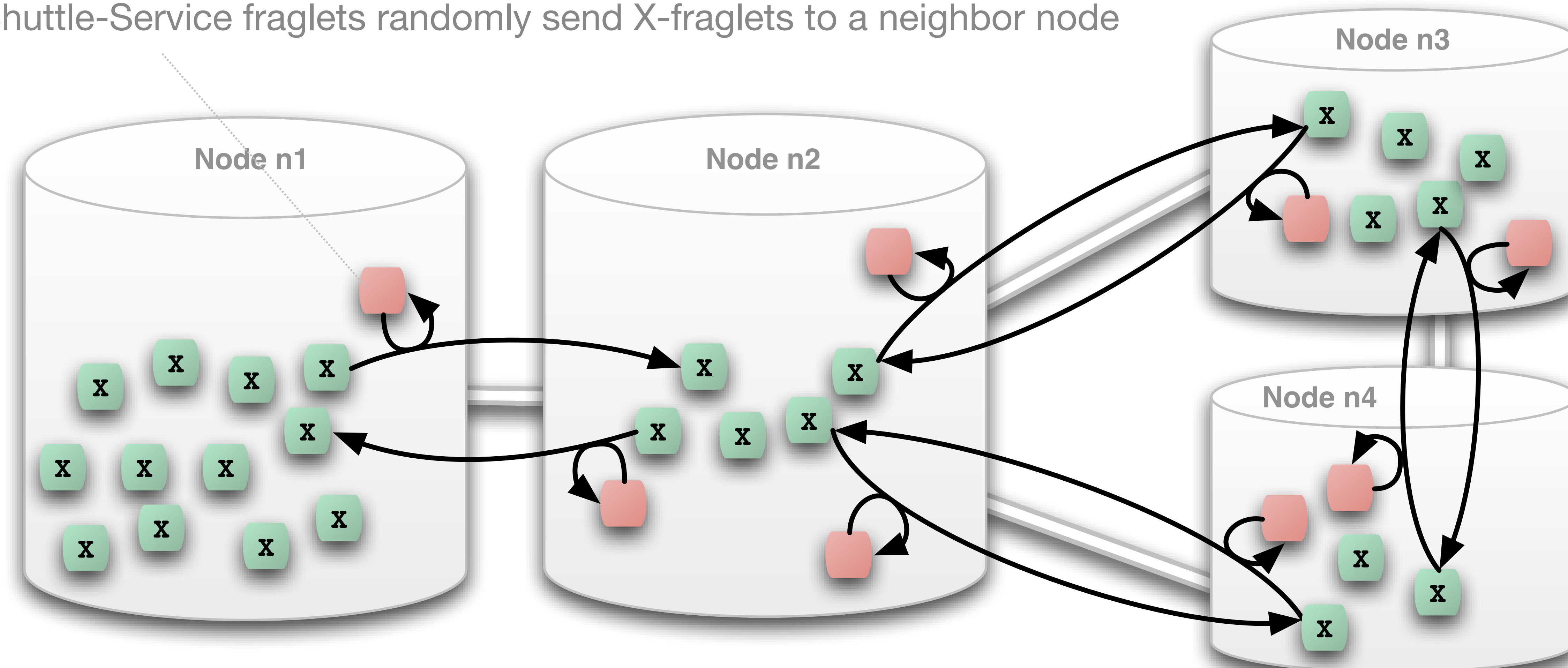
A Distributed Computation Example in Fraglets

- Consider the following distributed Fraglets program:
  Shuttle-Service fraglets randomly send X-fraglets to a neighbor node



A Distributed Computation Example in Fraglets

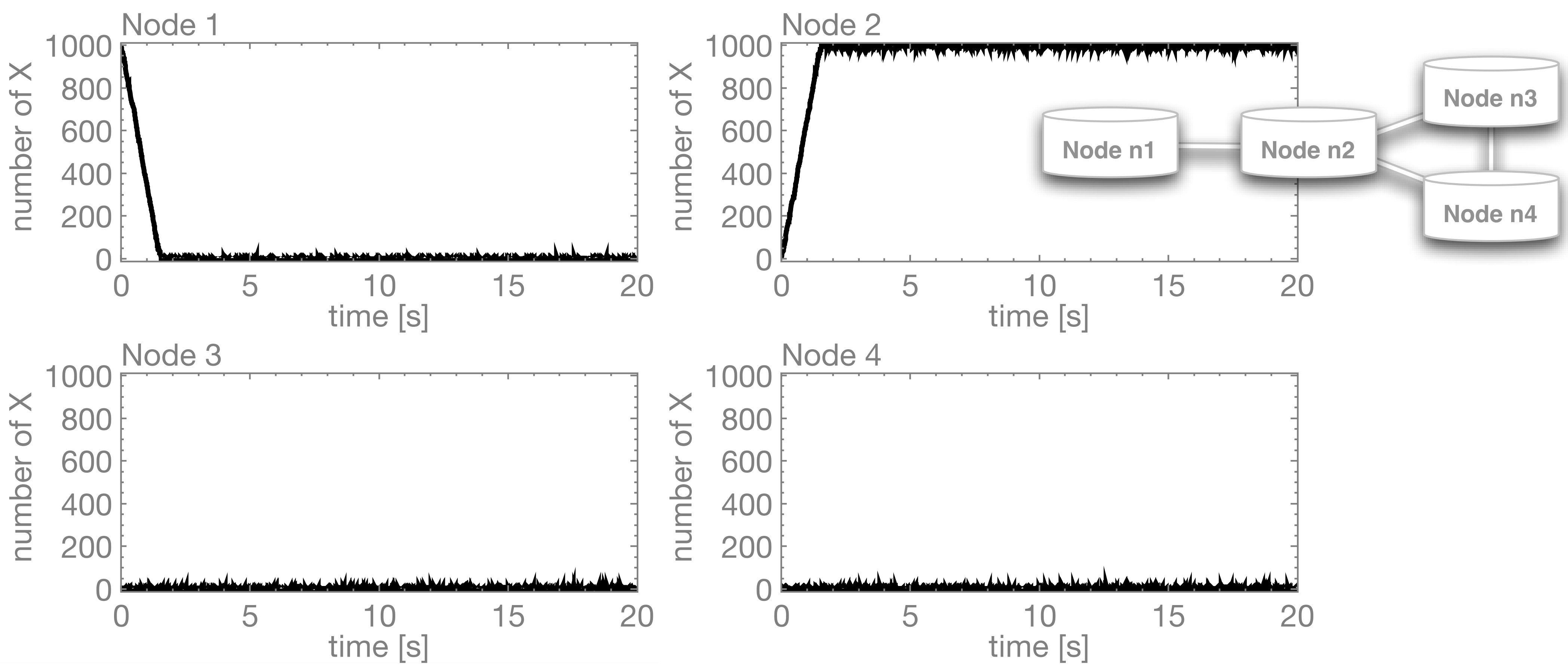
- Consider the following distributed Fraglets program:
  Shuttle-Service fraglets randomly send X-fraglets to a neighbor node



- What happens to the number of X-fraglets in each node?
  This depends on how the reactions are scheduled!

ASAP Scheduler

- ASAP Scheduling: Each node executes reactions as fast as possible.
- Result: X-molecules **drift to nodes with high degree**:



- How does nature “schedule” chemical reactions?

Chem. Scheduler ⇐ Stochastic Molecule Collisions

- Statistical Mechanics: **Prevalent molecules ⇒ more frequent collisions**.

  In a reaction vessel of constant volume, the collision frequency increases with a higher mol. density.

- Stochastic scheduling algorithm: [Gibson&Bruck, 2000]
  - For each reaction A + B → C, calculate the reaction interval

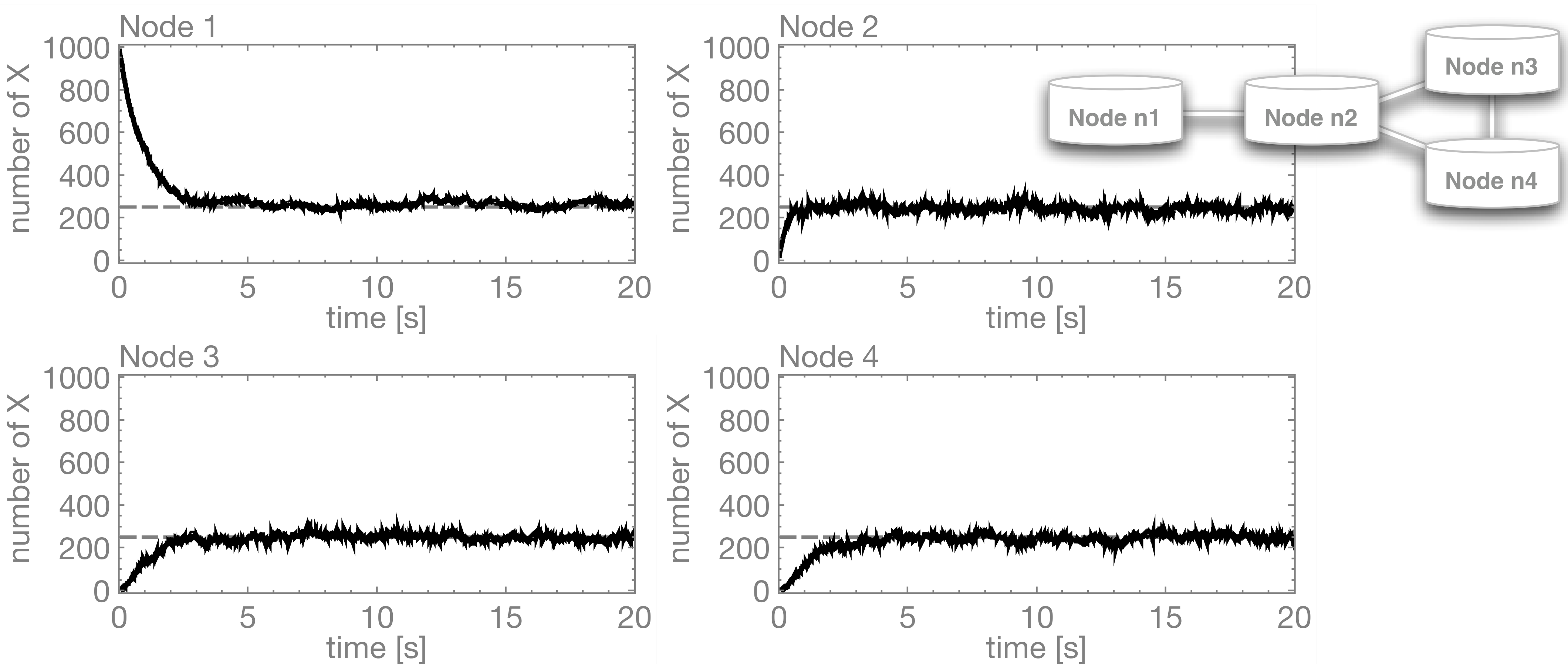
$$\tau \sim \mathrm{Exp}\left(\frac{1}{x_A x_B}\right)$$

x_A, x_B : number of A, B molecules; randomness due to Brownian motion.

  - Sort the next reaction time of all reactions into a priority queue.
  - **Wait** for the next reaction time.
  - Execute that reaction and calculate its next reaction interval.

Chem. Scheduler ⇒ Equilibrium

- Chemical Scheduler: Each node **sleeps** for a well-defined (but inherently stochastic) time between two reactions.
- Result: The distributed reaction network strives for an **equilibrium** in which a numerical result is present:
  Each node contains the same averaged number of initial X-molecules.

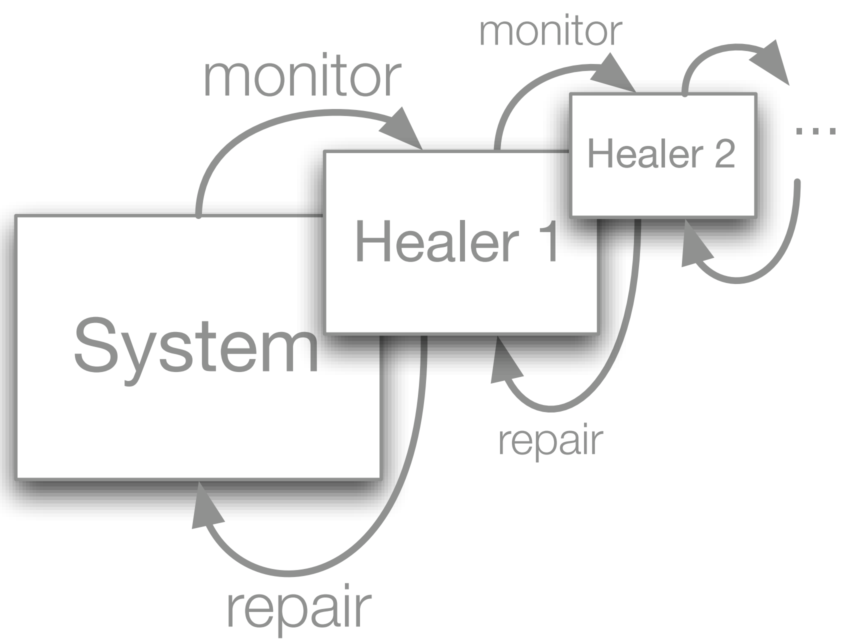


Stochastic Execution → Deterministic Prediction

- External observer: Due to the random reaction intervals, the observer cannot predict **which** reaction is next and **when** it will occur.
- But we can approximate the macroscopic dynamic behavior:
- Macroscopically, all reactions obey the **Law of Mass Action:**
  The reaction rate is proportional to the reactant concentrations;
  e.g. the rate of a reaction A + B → C is $r = x_A x_B$
- This allows us to use **ordinary differential equations** to describe the reaction dynamics.
- **Convergence proof** for our distributed averaging algorithm:
  - Create the ODEs from the reaction network: $\dot{x} = \mathbf{N}r$
  - Find the fixpoint: $\dot{\vec{x}} \equiv 0$
  - Determine whether the fixpoint is stable (perturbation analysis)

Self-Healing Programs

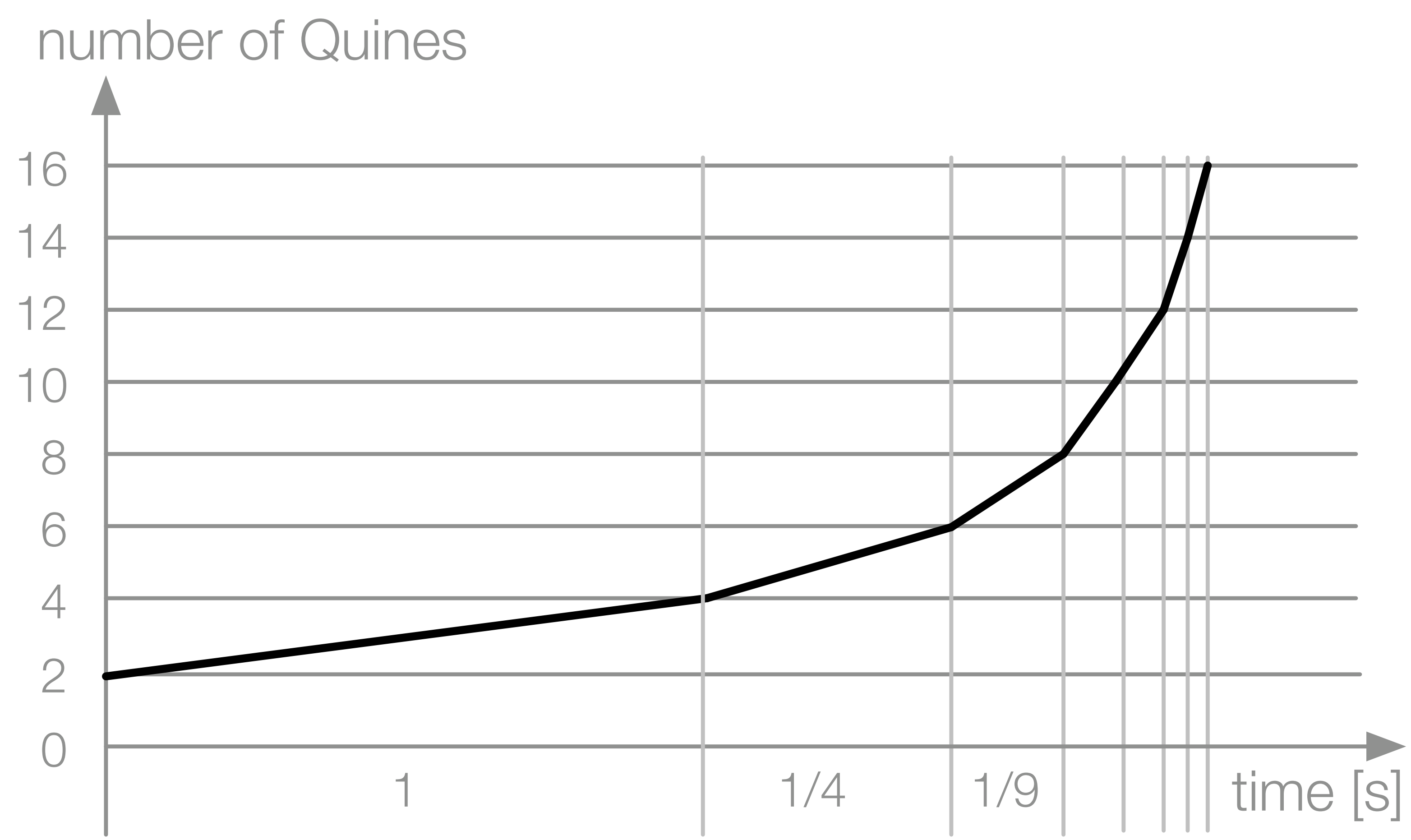
- The traditional approach: An infinite hierarchy of healers



- Our “chemical” approach avoids infinite regression:
  - The system shall monitor and **repair itself**.
  - Goal: **Code Homeostasis**: The system continuously regulates its internal composition to maintain a stable state.

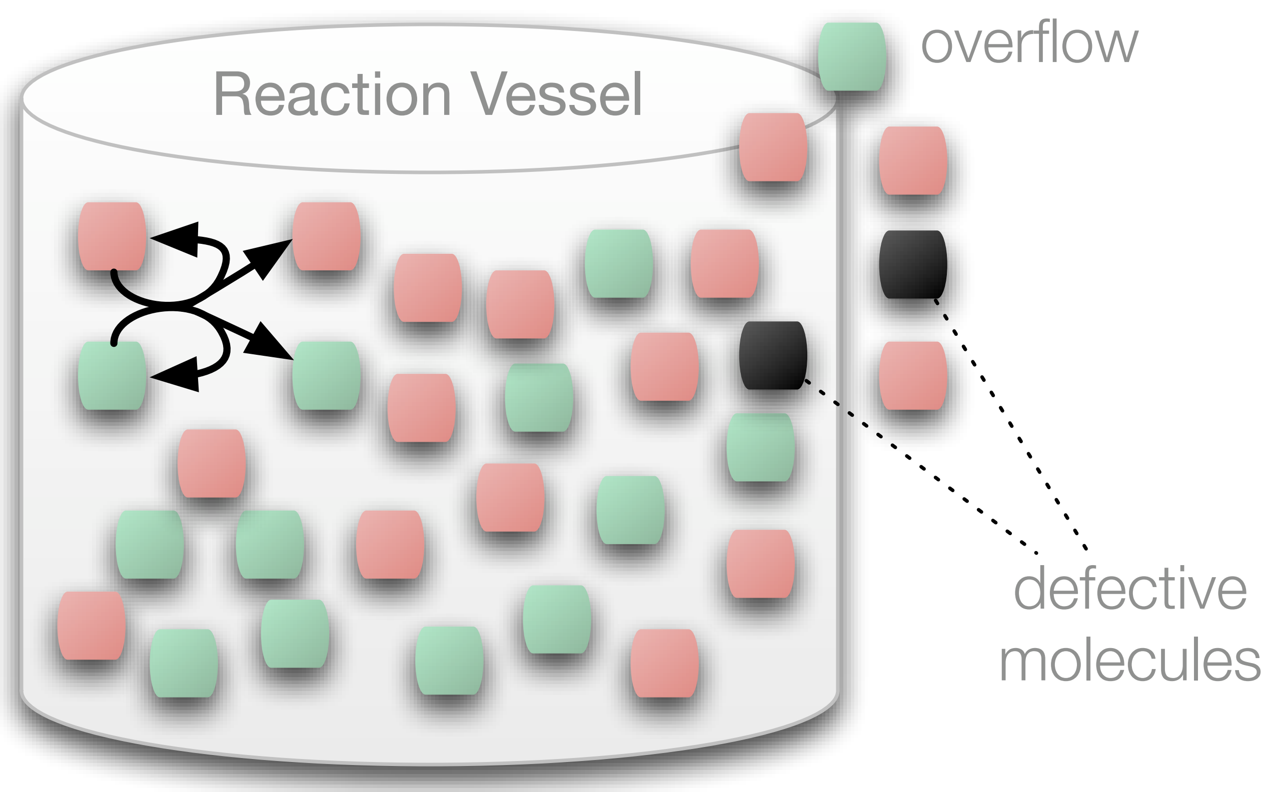
Self-Healing Programs - Growth

- In the first talk we presented the **duplicating Quine**.
- When scheduled by the Chemical Scheduler, the population of Quines exhibits **exponential growth**:



Self-Healing Programs - Finite Reactor

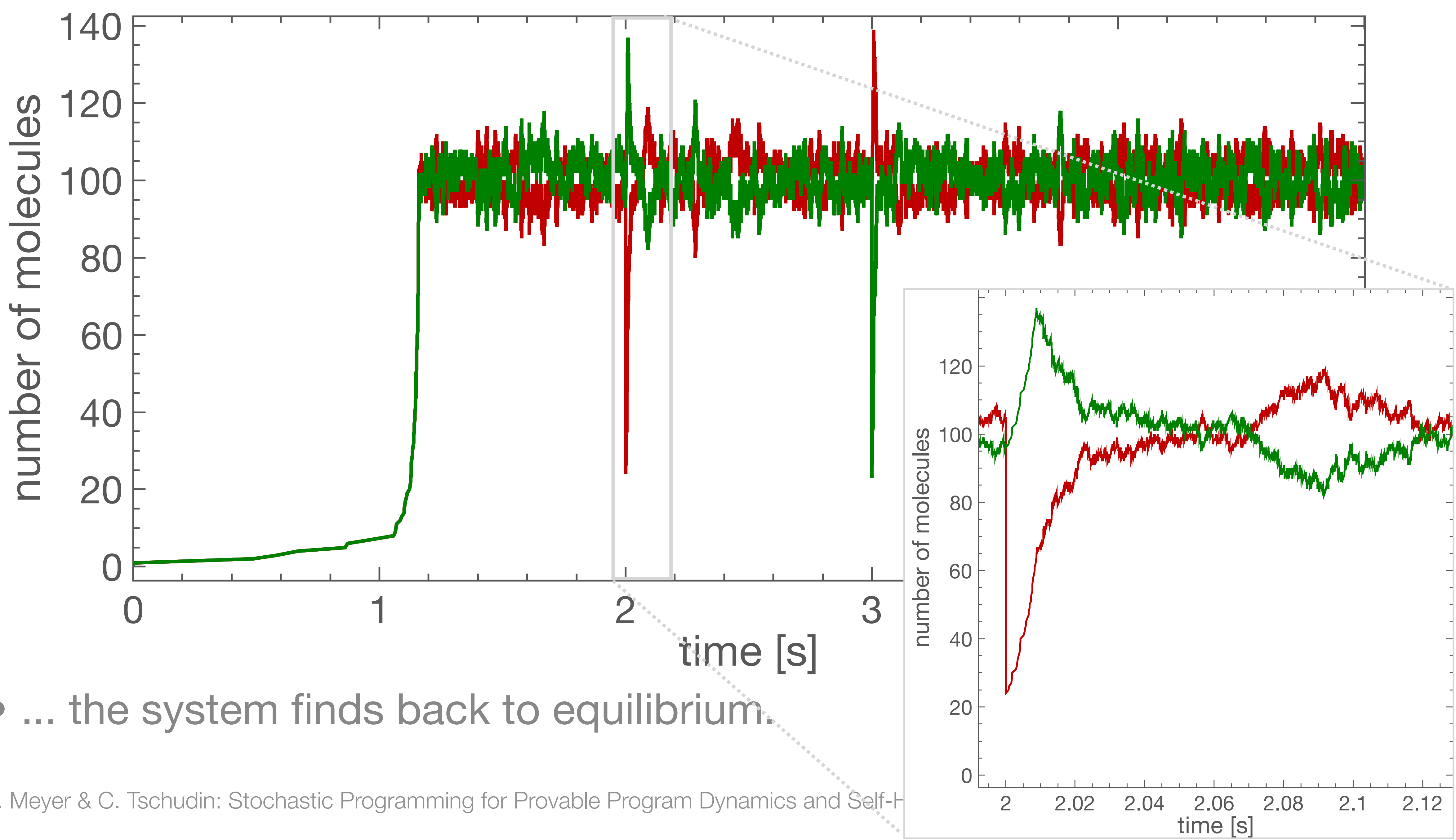
- We limit the capacity (number of molecules) of the reaction vessel...
  Due to limited resources, we cannot let the population of Quines grow infinitely.
- ...by randomly destroying molecules when this vessel capacity is exceeded.



- This adds selective pressure to the population of molecules:
  Only molecules that continuously replicate themselves are able to “survive”.

Programs Recover From Code Deletion Attacks

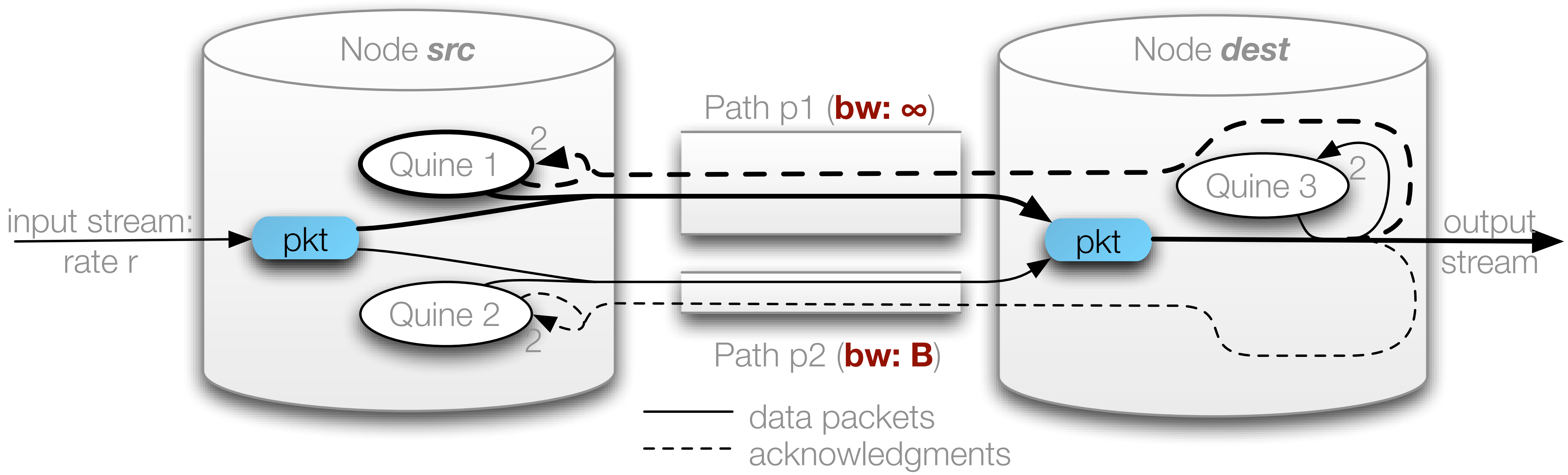
- Even when removing 80% of the `[matchp x ...]` or the `[x ...]` molecules...



- ... the system finds back to equilibrium.

A Self-Healing Load-Balancing Protocol

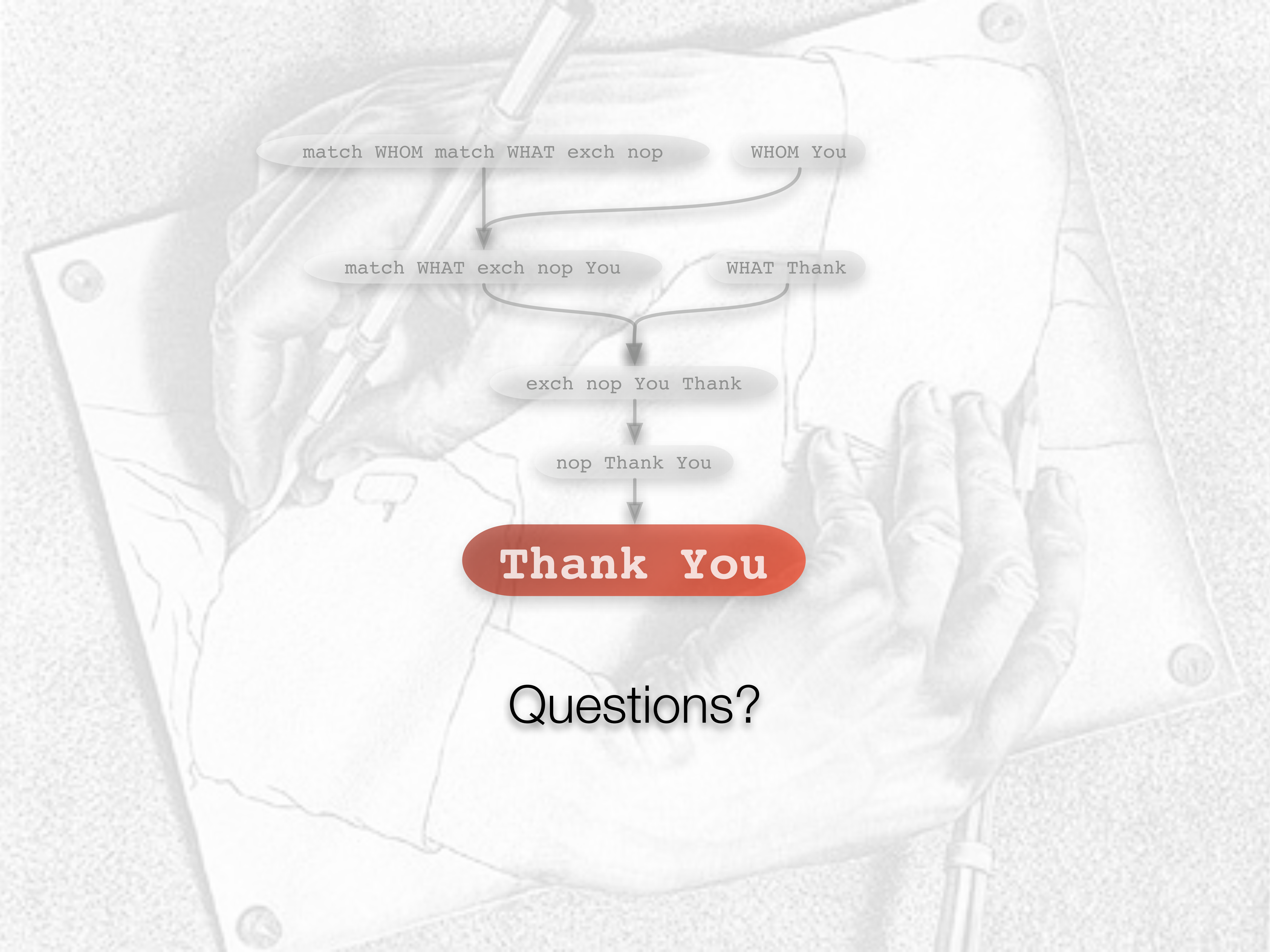
- Goal: Balance packet stream from node ***src*** to node ***dest*** over two links.
- Node ***src*** uses of two **competing Quines** to send data packets to node ***dest***.



- The Quines only replicate when receiving an acknowledgment.
- If path ***p2*** drops packets, the replication rate of Quine 2 decreases:
- The relative concentration of Quine 1 increases; it forwards more packets.

Conclusions

- A chemical, instead of ASAP, scheduling enables **emerging equilibria**.
- The solution to a computation is represented as an equilibrium of a dynamic system.
- Since there is no distinction between code and data:
  Code can be brought to equilibrium, i.e. software becomes **self-healing**.
- Application fields: continuously running (distributed) processes where an equilibrium represents the ideal situation/solution: e.g.
    - Routing protocols,
    - Flow control of data traffic,
    - Robotics,
    - Topological self-organization of sensor/actor networks, and reactive systems in general

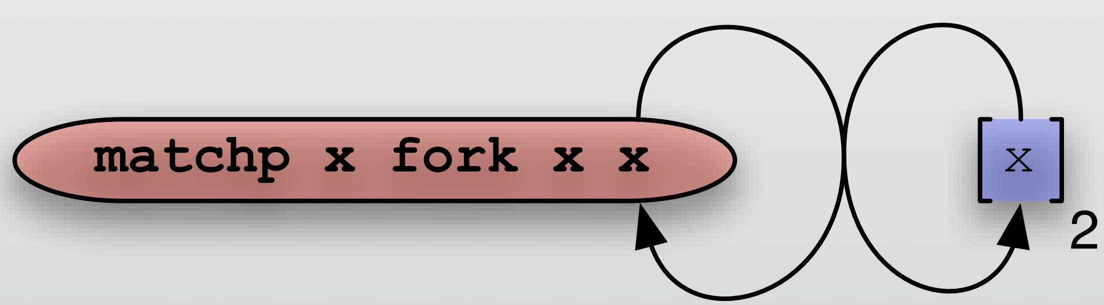

match WHOM match WHAT exch nop
WHOM You
match WHAT exch nop You
WHAT Thank
exch nop You Thank
nop Thank You
Thank You
Questions?

Backup Slides

Stochastic Scheduling and Deterministic Prediction

- Reactions are **NOT** executed as fast as possible
- Reactions are rather scheduled for a later time according to the **Law of Mass Action**:
  reaction rate = product of the reactant concentrations

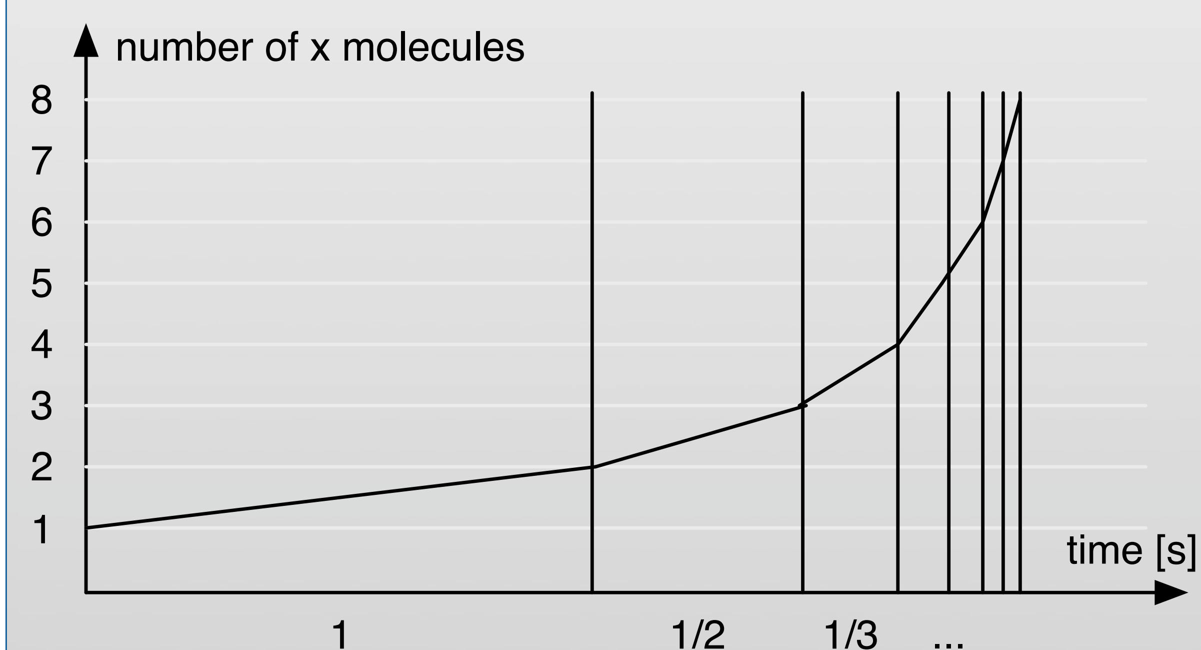
Reaction Network:



Differential Equation Approximation:

$$\frac{dx}{dt} = x$$

Exponential Growth of x w.r.t. Simulation Time:



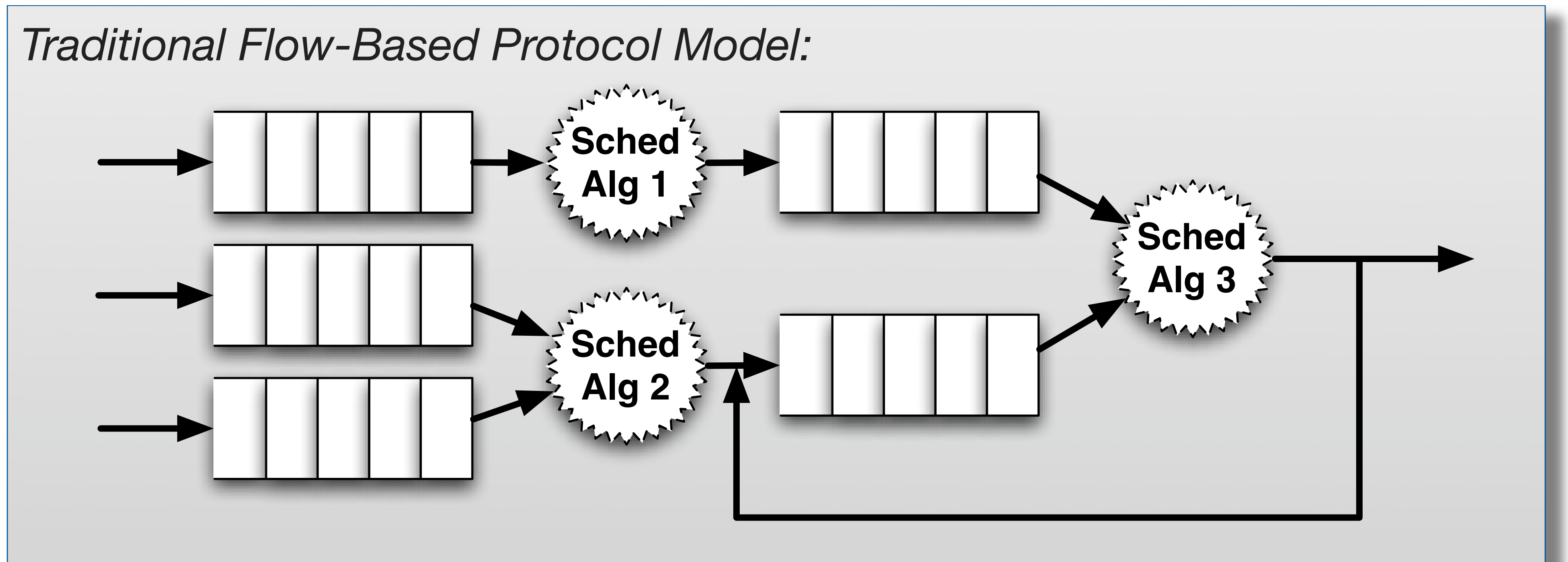
Representation-Free Communication

- Comparison of how information is encoded in traditional and chemical networking protocols:

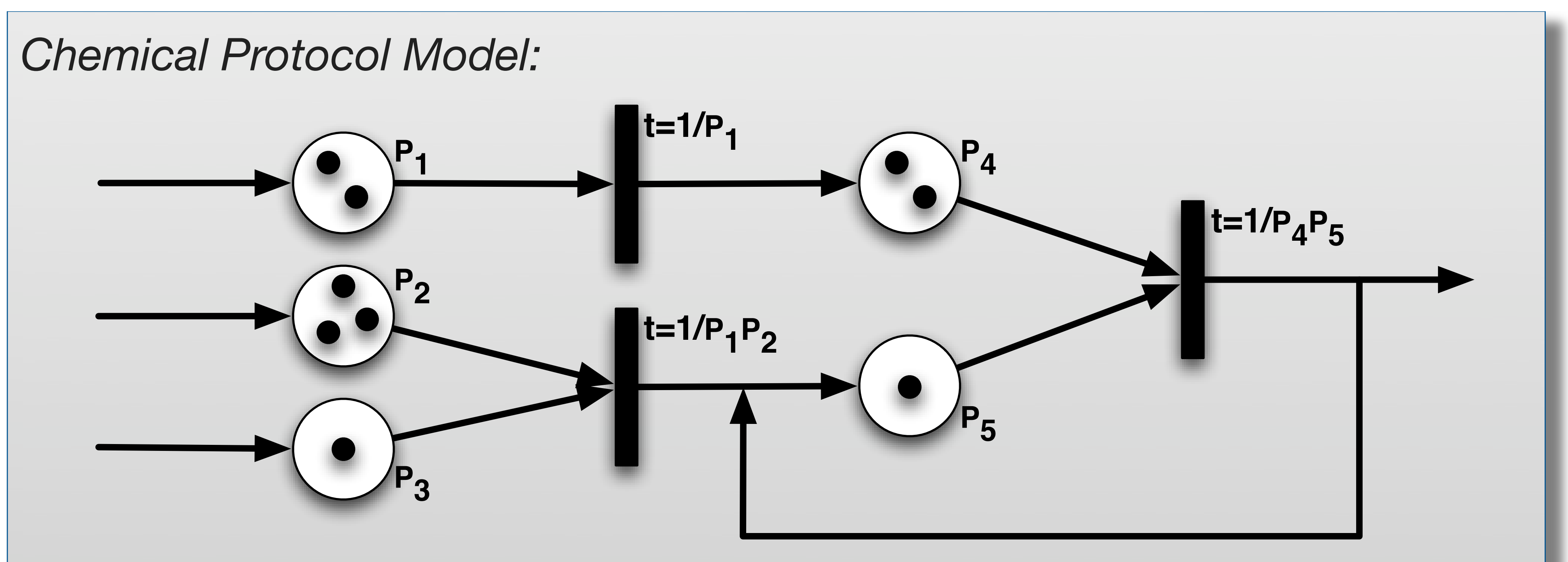
	Traditional Protocols	Chemical Protocols
Representation of local state information	**symbolically**: e.g. integers, flags	**abundance**: number of molecules
Representation of exchanged information	**symbolically**: encoded as symbols inside packets	**rate-based**: the packet rate represents inform.

Petri-Net Representation

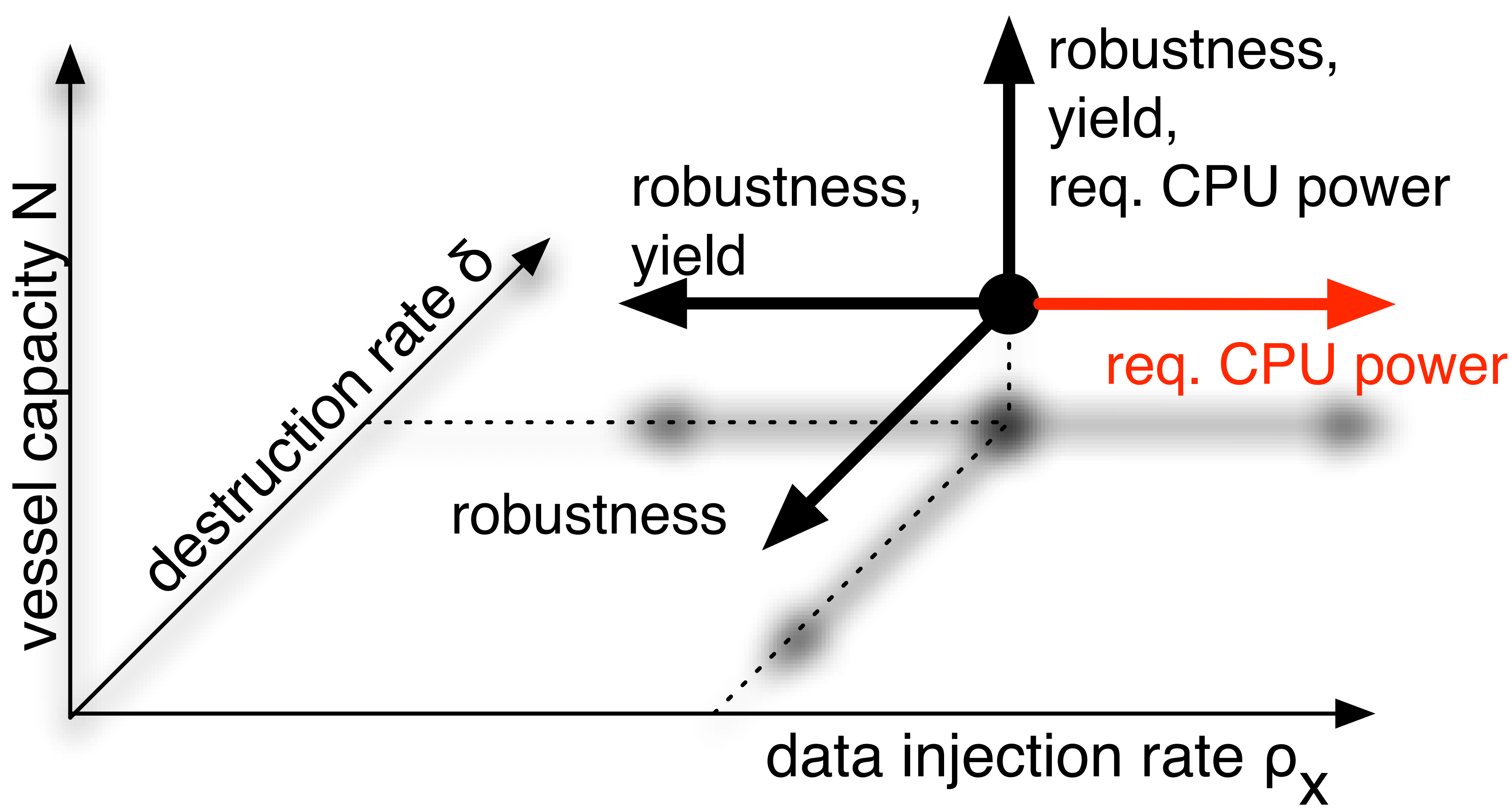
- fixed number of queues
- different scheduling alg. (e.g. FIFO, EDF)
- traffic modeled as a Markov process



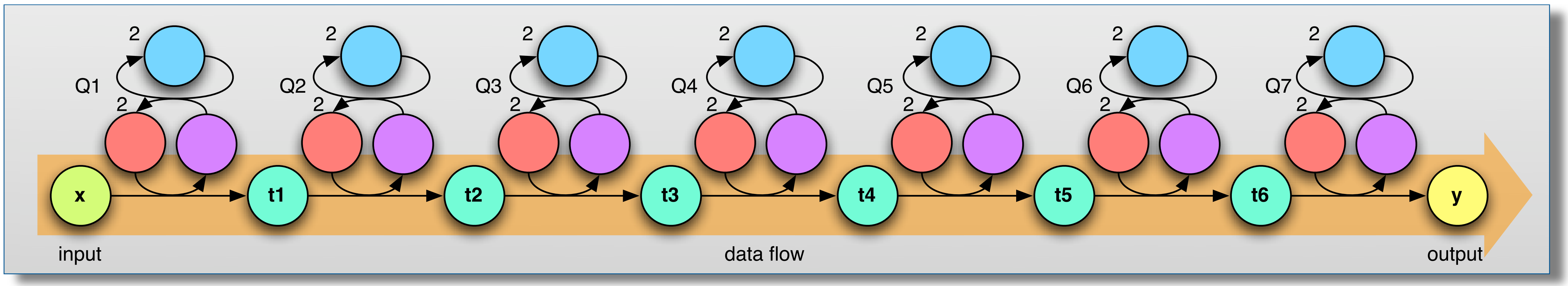
- var. number of substances
- a single scheduling alg. (Law of Mass Action)
- traffic ***and*** execution modeled as Markov proc.

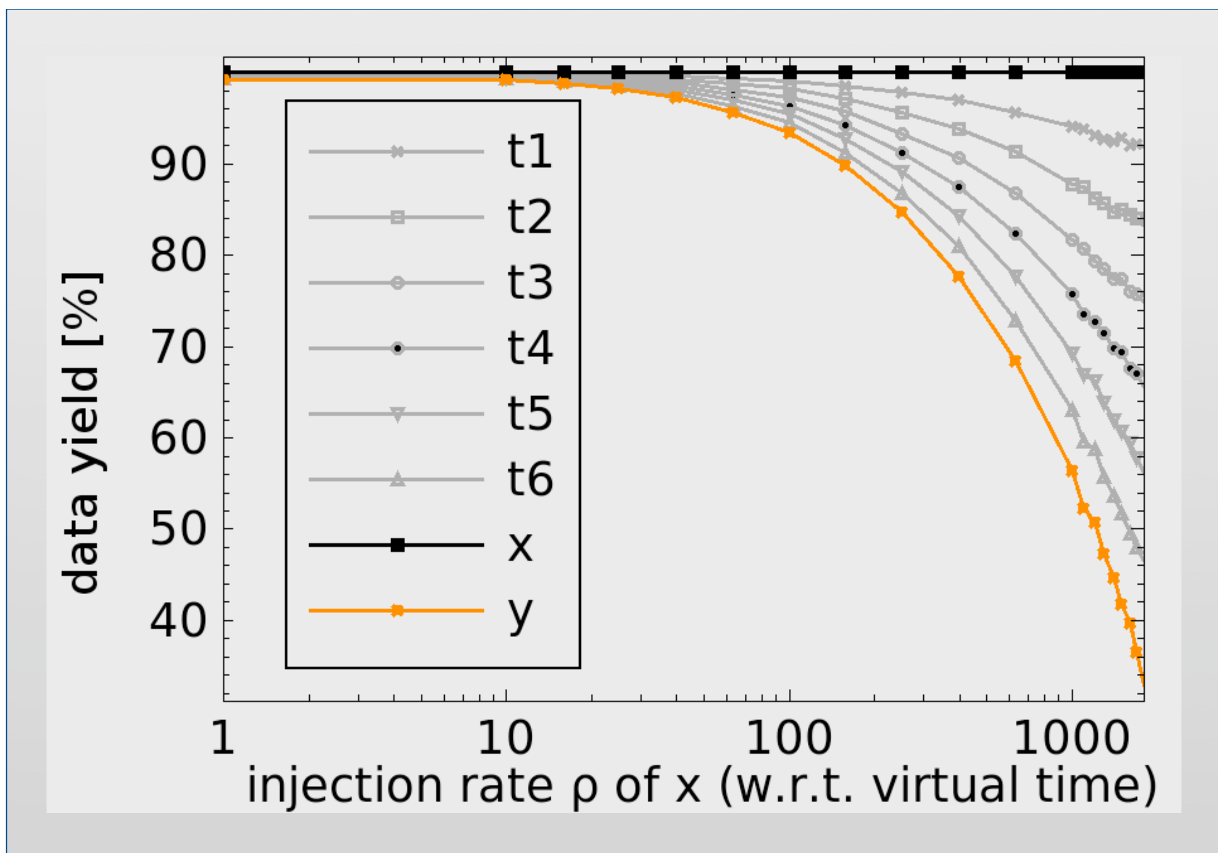


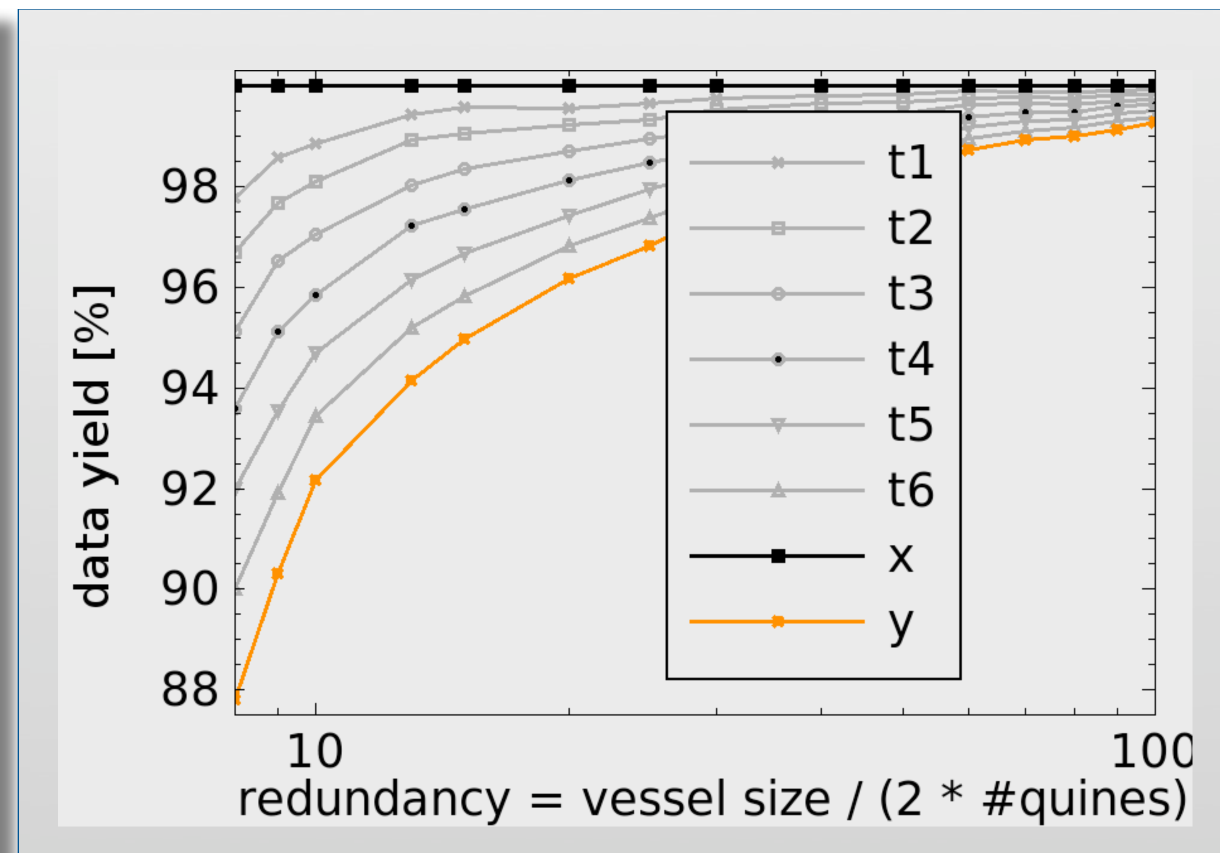
Robustness of Quines to Deletion (qualitatively)



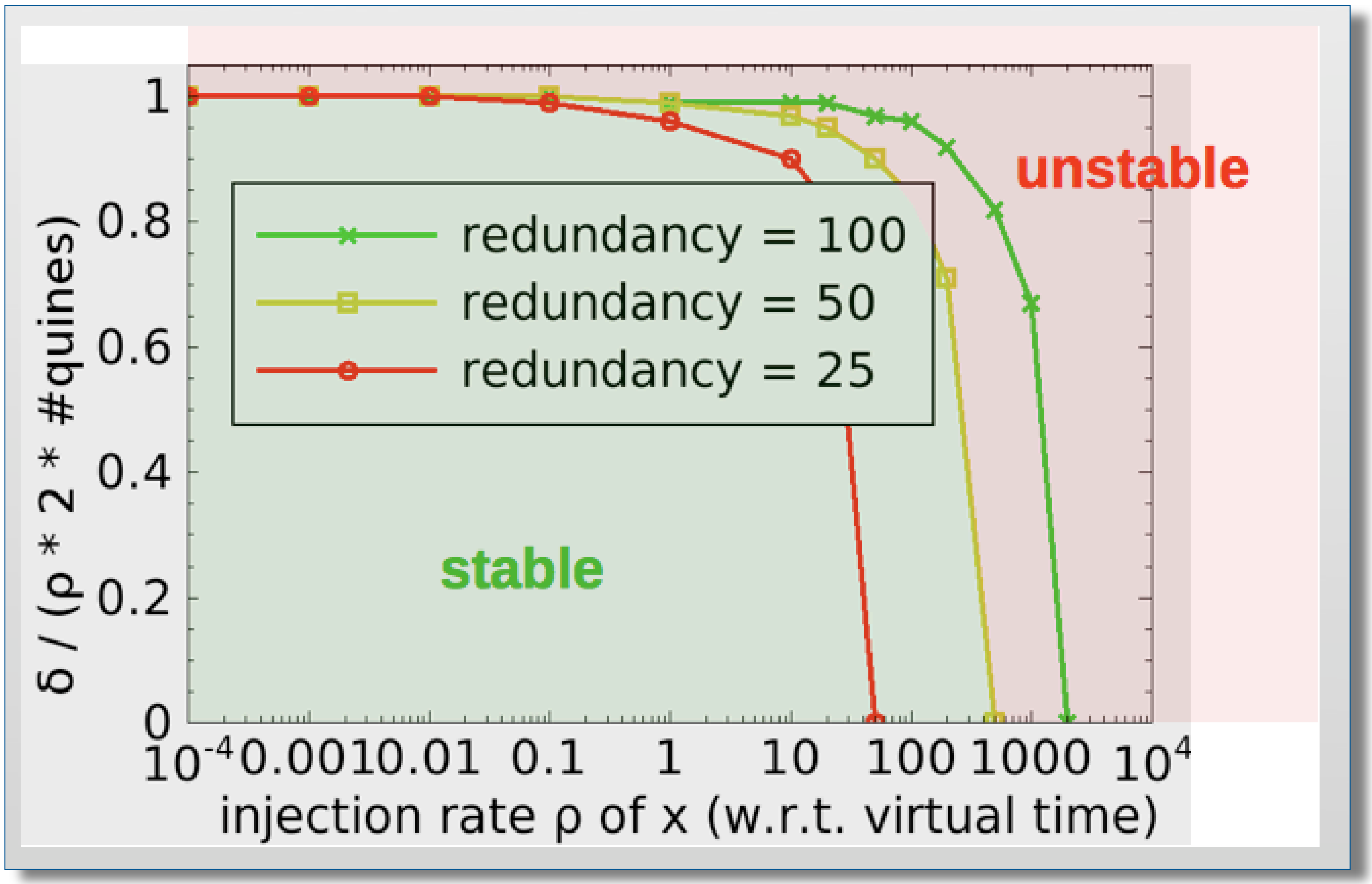
Robustness of Quines to Deletion (quantitatively)







Robustness of Quines to Deletion (quantitatively)



Robustness of Quines to Mutation