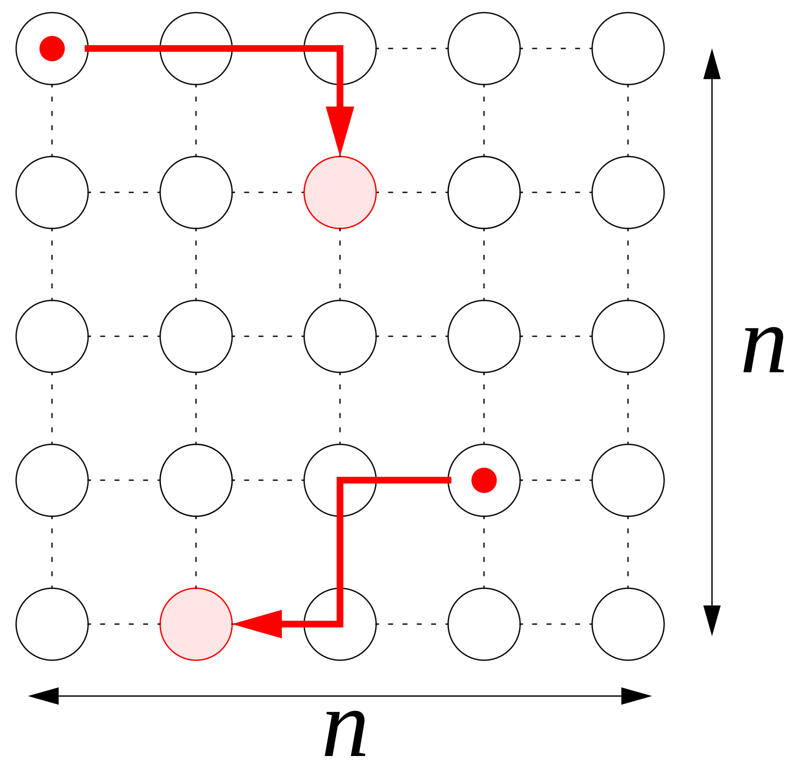
Randomized

Greedy Hot-Potato Routing

Costas Busch, Maurice Herlihy, and Roger Wattenhofer

Brown University

Network: $n \times n$ Mesh or Torus



Model:

- Synchronous network
- At most one packet per link per time step

Hot-Potato Routing [B68]

Definition: no buffers

Observation:

packets are forwarded immediately

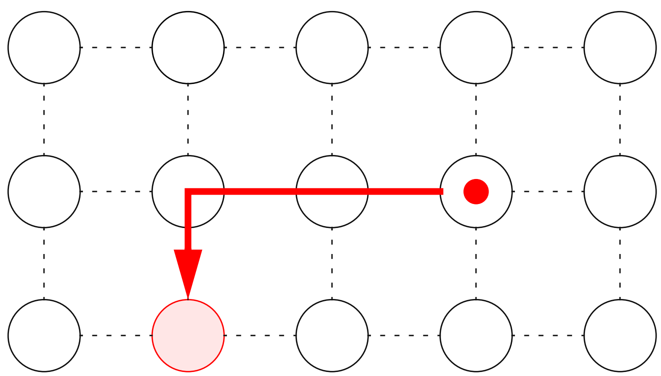
Applications

- Optical networks
  light cannot be stored

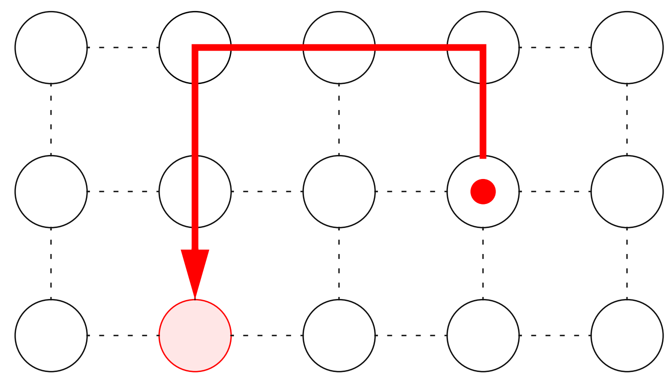
- Non-optical networks
  simple hardware

<u>Greedy</u> Hot-Potato Routing

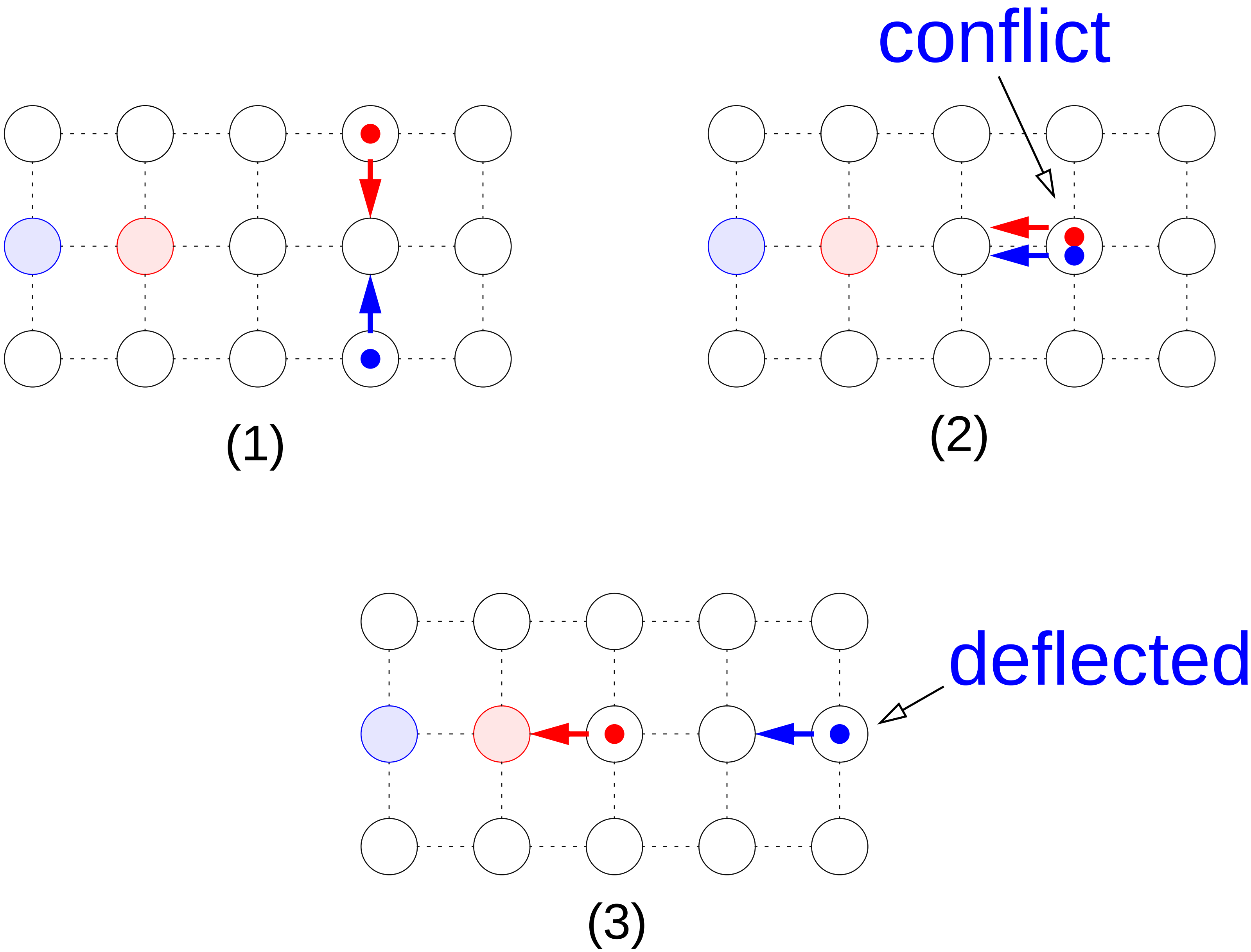
Packets follow links closer to destination



Greedy



Non Greedy


conflict
(1)
(2)
deflected
(3)

Greedy hot-potato algorithms:

specify how the conflicts are resolved

Advantage:

work extremely well in practice [AS91,M89]

Problem: hard to analyze

Contribution

- New greedy hot-potato algorithm

- Better bounds for batch problems

Batch routing:

at time 0 each node injects a packet

Question:

how much time until all packets reach destinations?

Batch routing problems

- Permutation routing
  a node is the destination of one packet

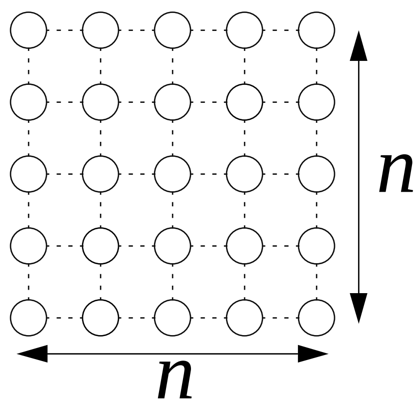
- Random destinations

- General batch routing

Previous best bound

Permutation; Random destinations;
General batch problem

$O(n^2)$ (deterministic)



Our bounds

- Permutation; Random destinations

  $O(n \cdot \log n)$

- General batch problem

  $O(m \cdot \log n)$

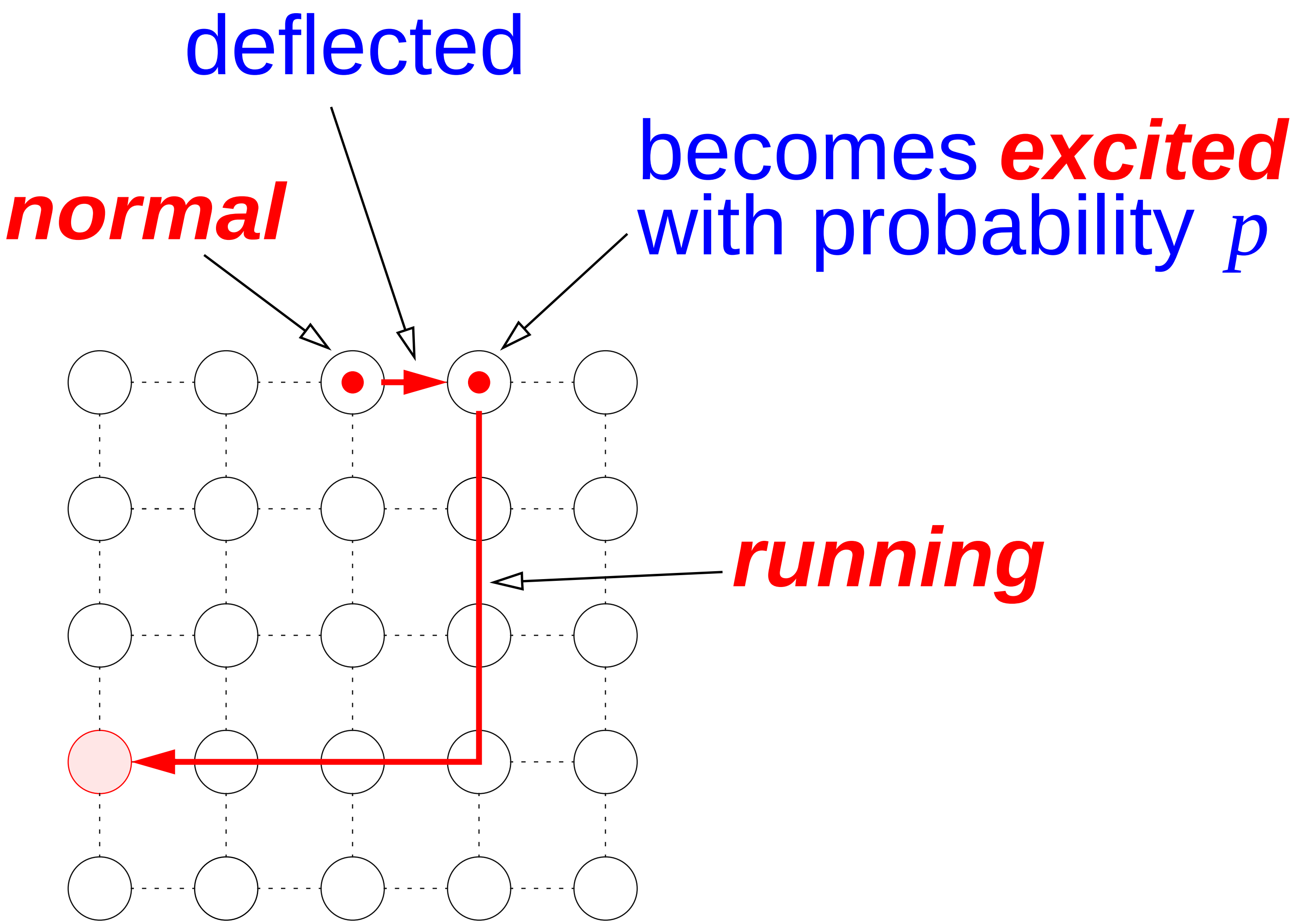
  m: max destinations in a row or column

With probability at least: $1 - \frac{1}{n}$

Our Algorithm

Packet States	Priority
running	higher
excited	
normal	lower

Home run path

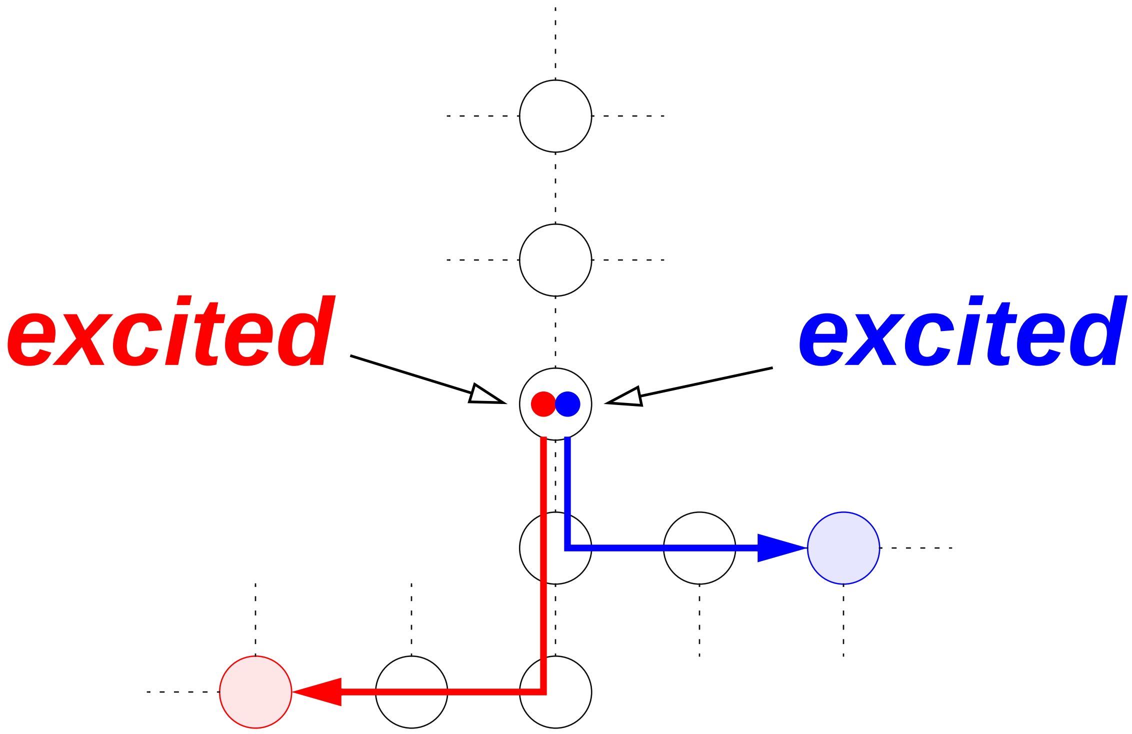


Time Analysis for One Packet

Permutation; Random destinations

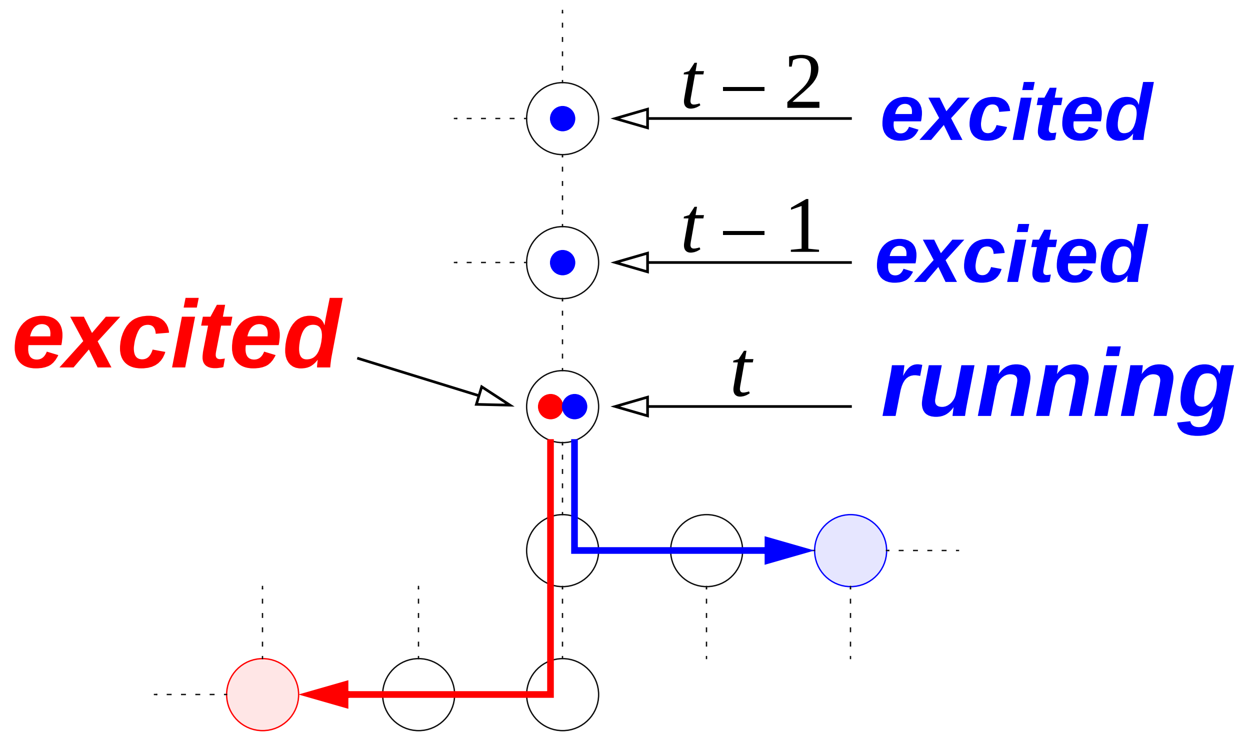
Expected time: $O(n)$

Interrupting a home run: **case 1**



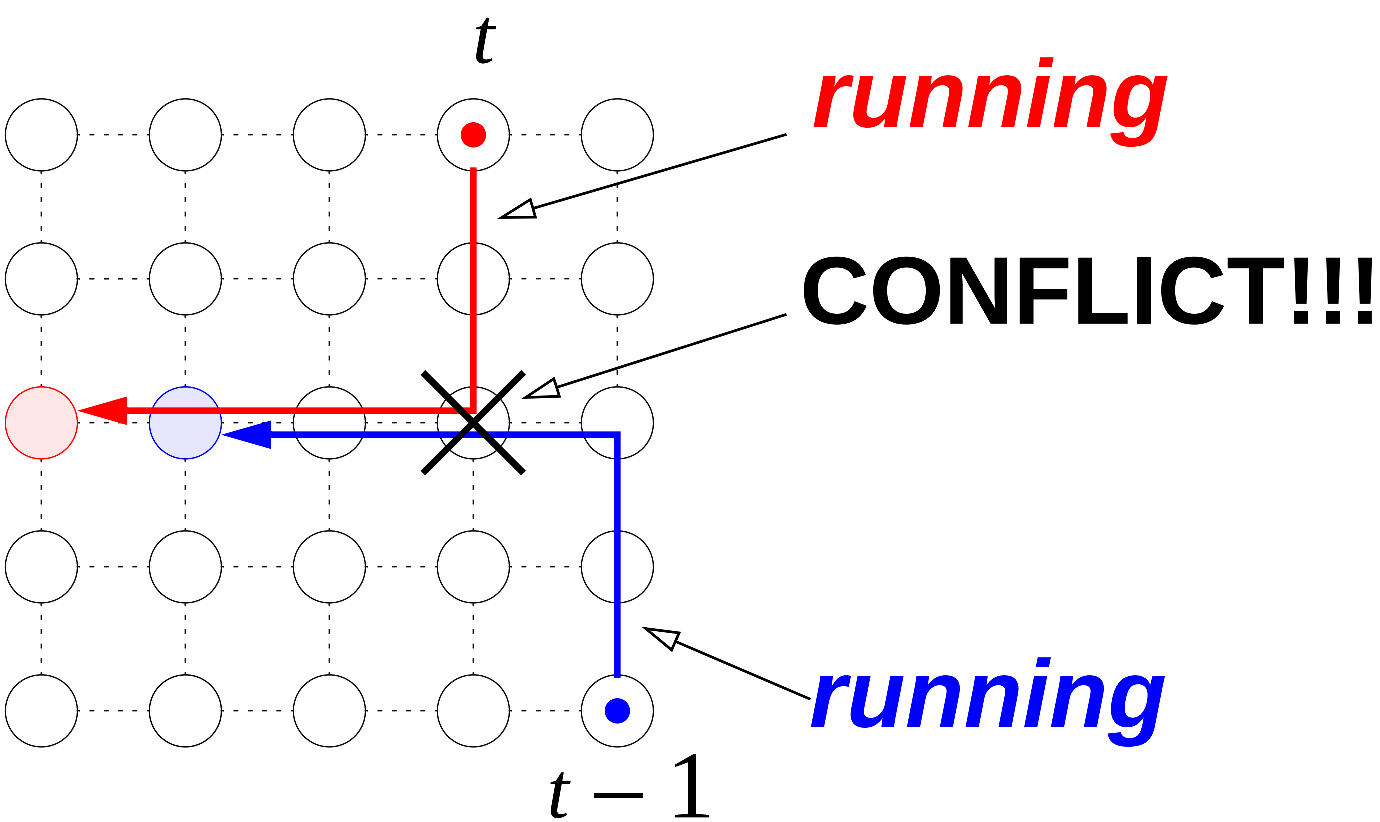
Probability of not interrupting: $(1-p)^3$

Interrupting a home run: **case 2**



Probability: $(1-p)^{4 \cdot n}$

Interrupting a home run: **case 3**



Probability: $(1-p)^n$

Probability of succeeding in a home run

Take $p = \frac{1}{n}$

$$\underbrace{(1-p)^3}_{\text{case 1}} \cdot \underbrace{(1-p)^{4 \cdot n}}_{\text{case 2}} \cdot \underbrace{(1-p)^n}_{\text{case 3}} \geq \frac{1}{c}$$

Good event after a deflection:

getting ***excited*** and then reaching destination

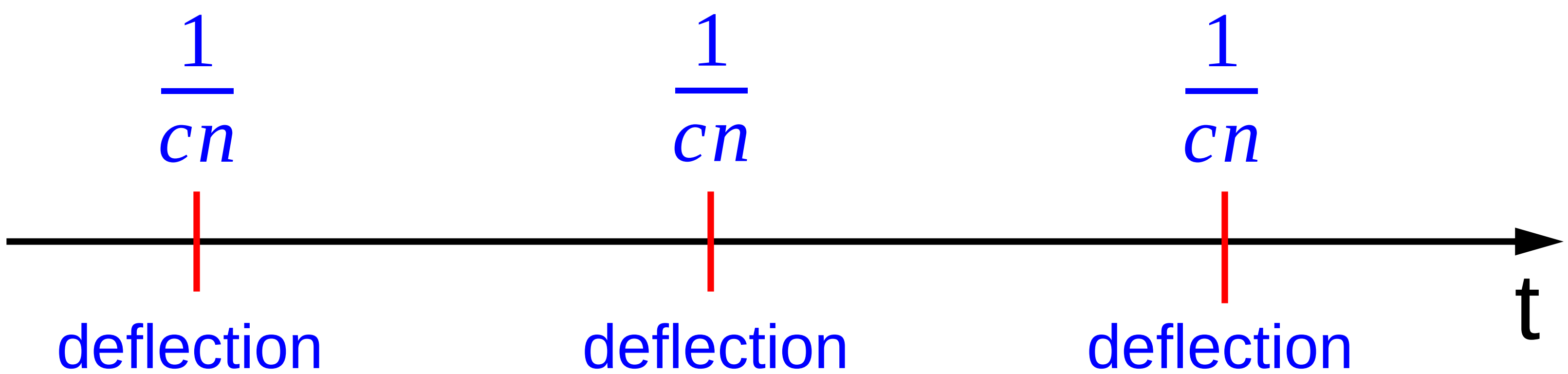
Probability:

$$p \cdot \frac{1}{c} = \frac{1}{n} \cdot \frac{1}{c} = \frac{1}{cn}$$

excited

succeeding in a home run

Counting deflections



Expected number of deflections: cn

Expected total time

$$2n + 2 \cdot cn \;=\; O(n)$$

initial distance
from destination

expected
deflections

Time for all packets

$O(n \cdot \log n)$

With high probability: $1 - \frac{1}{n}$

General Batch Problems

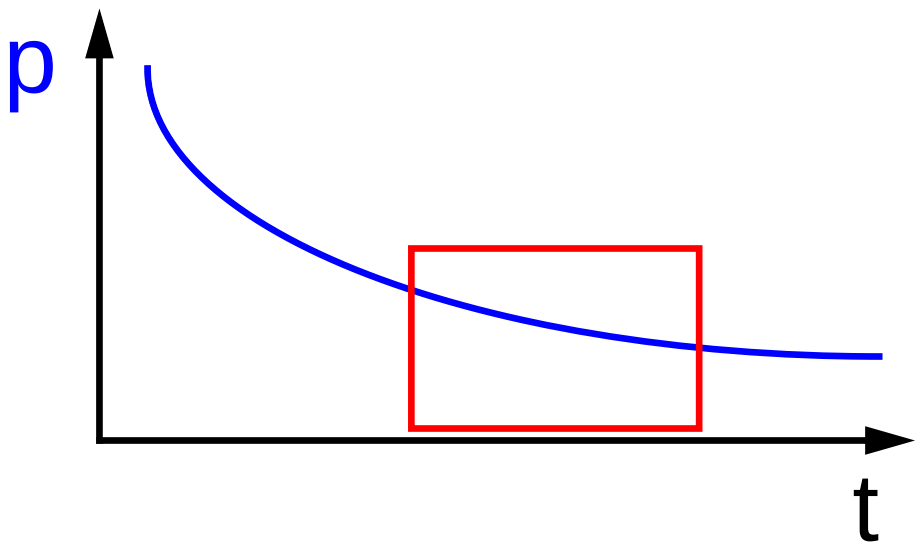
Time for all packets: $O(m \cdot \log n)$

m = max destinations in a row

Take $p = \frac{1}{m}$

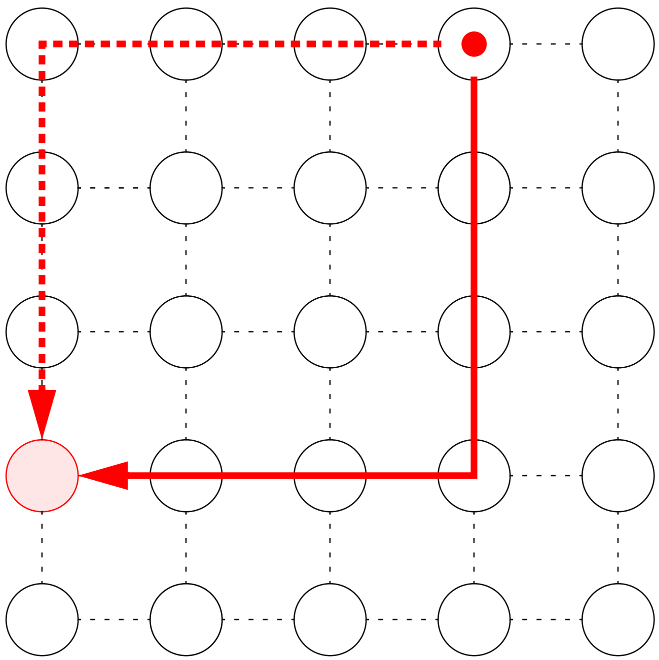
Problem: nodes do not know m

Solution: take $p = \dfrac{\log t}{t}$



Improved Algorithm

Two home run paths



Future Research

Dynamic analysis

More dimensions

Arbitrary network topology