Greedy Hot-Potato Routing

Costas Busch

Brown University

Talk Outline

Greedy Hot-Potato Routing

One-Bend Algorithm

Multi-Bend Algorithm

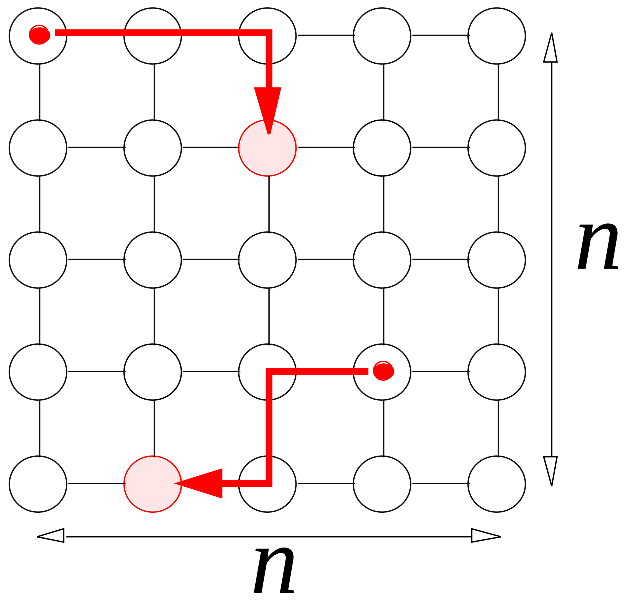
Future Research

Routing

Routing is essential: Internet,
multiprocessors

We consider multiprocessors

Our network: **Mesh**



Synchronous network

From a node to a link:

at most one packet per time step

Local decisions

Hot-Potato Routing

Definition: no buffers

Observation: packets are forwarded immediately

Applications

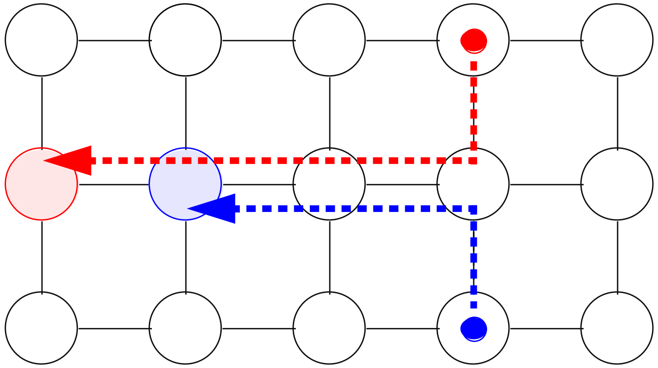
Optical networks

light is difficult to store

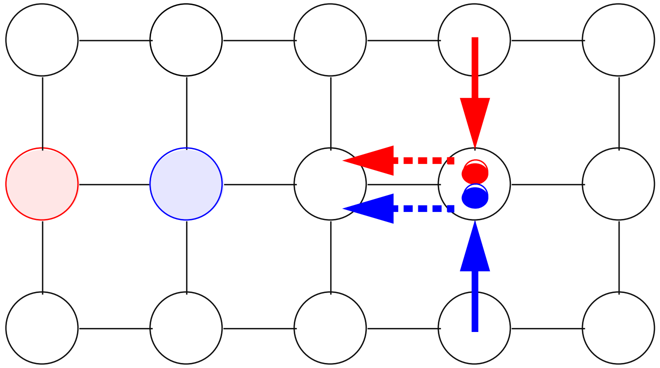
Non-optical networks

simple hardware

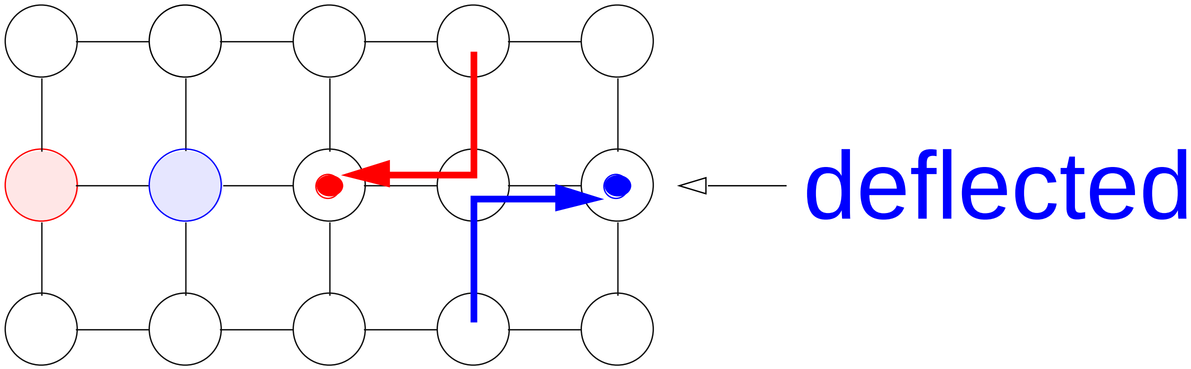
Conflicts in hot-potato routing



(1)



(2)



(3)

Hot-potato algorithms

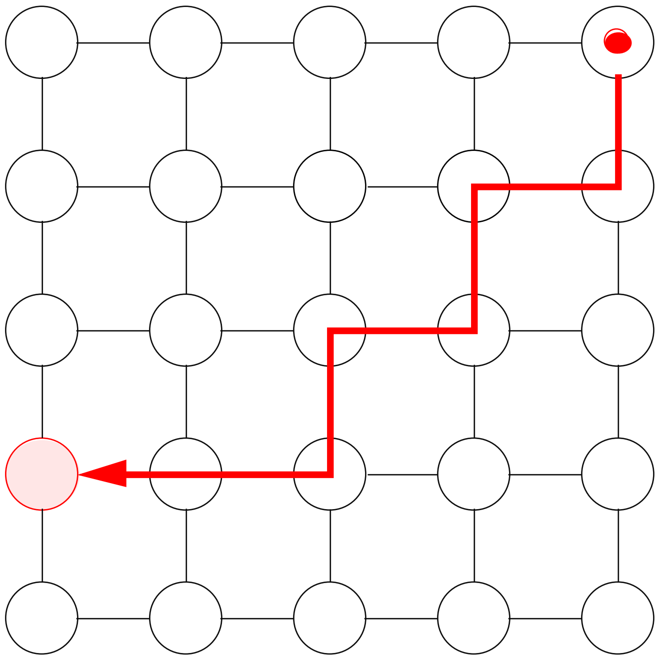
Specify how the packets move

Specify how the conflicts are resolved

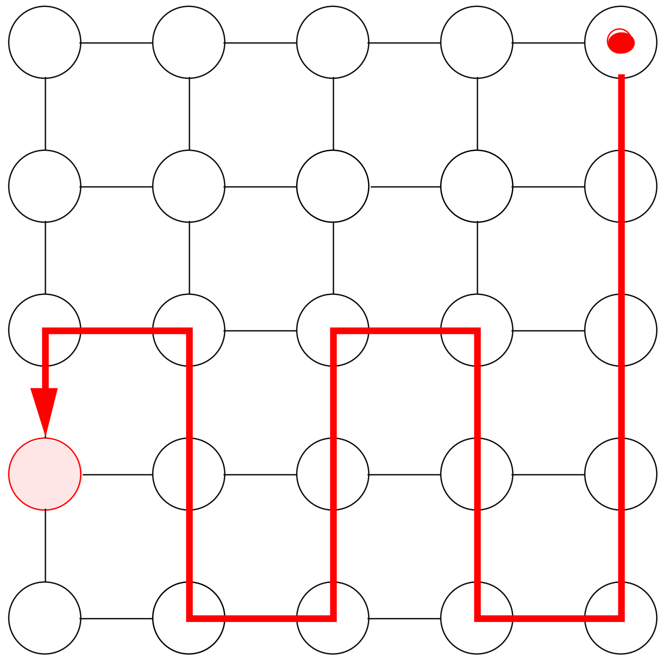
Greedy Algorithms

Packets prefer links towards destination

Greedy



Non-Greedy



Advantages of greedy algorithms

Adaptive

Simple implementations

Work very well in practice

Maxemchuk INFOCOM'89

Challenge: hard to analyze

Our Contribution

New greedy hot-potato algorithms

Better bounds for batch routing problems

Batch routing: at time 0 each node injects a packet

Question: how much time until all packets reach destinations?

Interesting batch problems

Permutation

Random destinations

General batch problem

Previous bounds for greedy algorithms

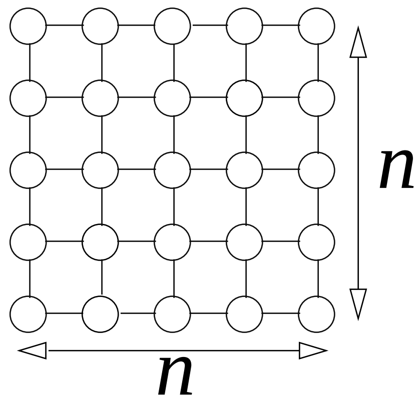
Ben-Dor, Halevi, Schuster PODC ’94

Borodin, Rabani, Schieber TPDS ’97

Permutation:
Random destinations: $O(n^2)$

Far from lower bound: $\Omega(n)$

Deterministic algorithms



Our bounds: one-bend algorithm

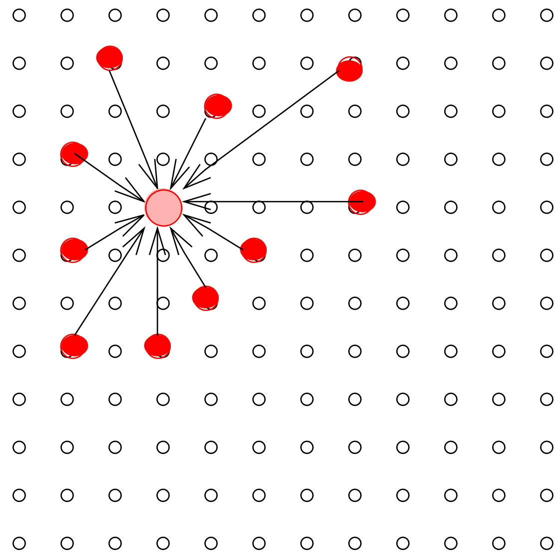
Busch, Herlihy, Wattenhofer SODA '00

Permutation:
Random destinations: $O(n \cdot \log n)$

Close to lower bound: $\Omega(n)$

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

General batch problems



Time needed:

$L = \Omega(n^2)$

“Hard” problem instances: $L = \Omega(n)$

L: lower bound

Our bounds: multi-bend algorithm

Busch, Herlihy, Wattenhofer STOC '00

Any "hard" general problem instance:

$$O(L \cdot \log^3 n)$$

L: lower bound

First competitive hot-potato algorithm

Talk Outline

Greedy Hot-Potato Routing

One-Bend Algorithm

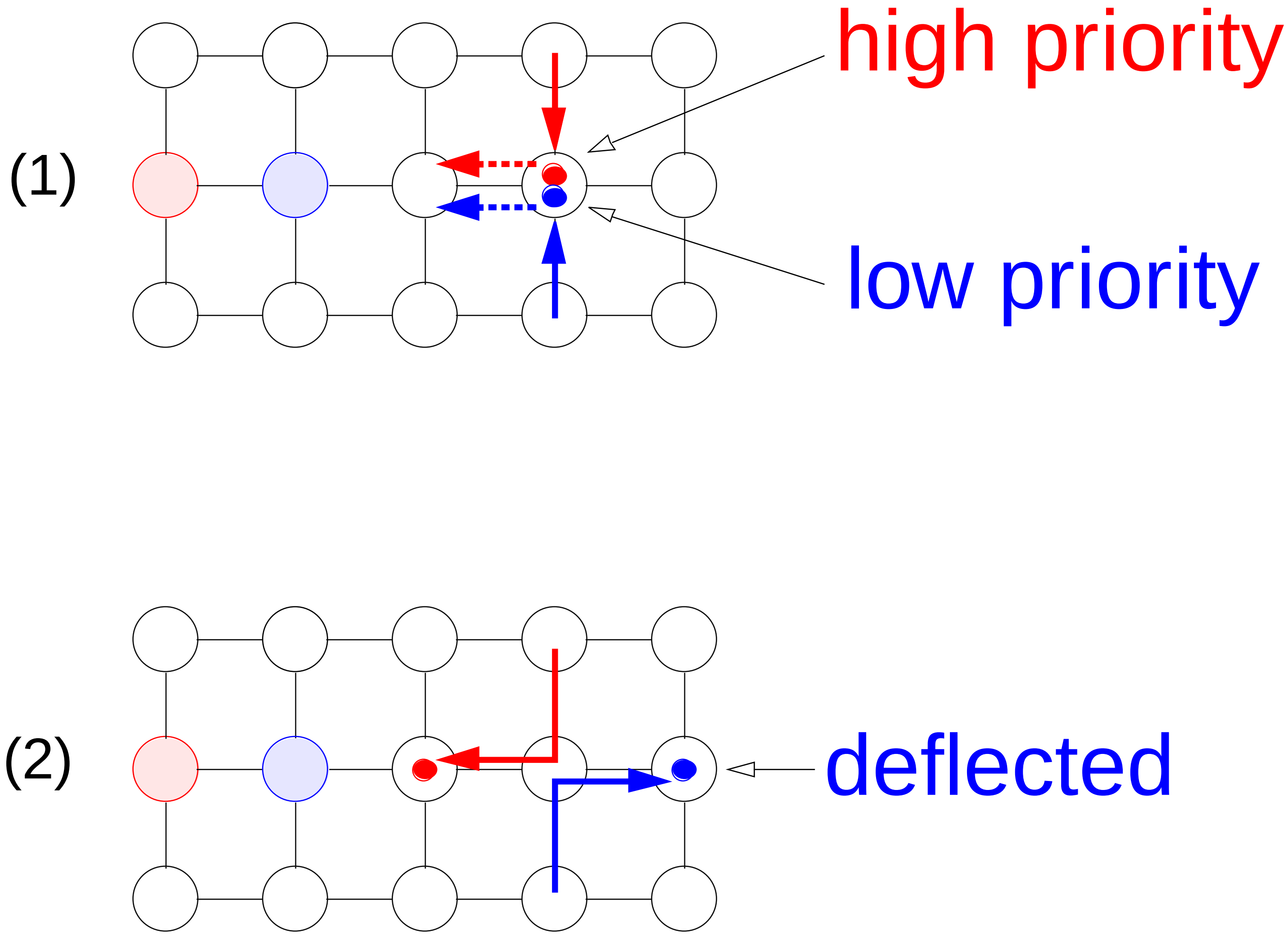
Multi-Bend Algorithm

Future Research

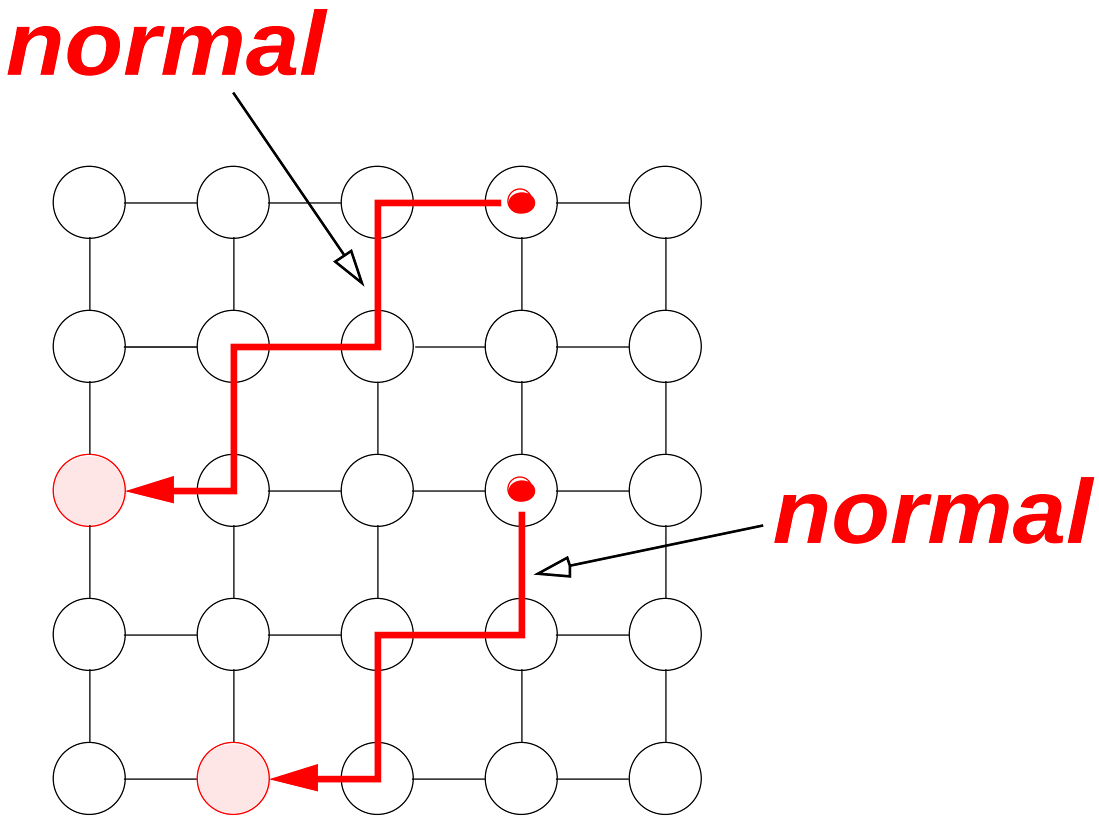
One-Bend Algorithm

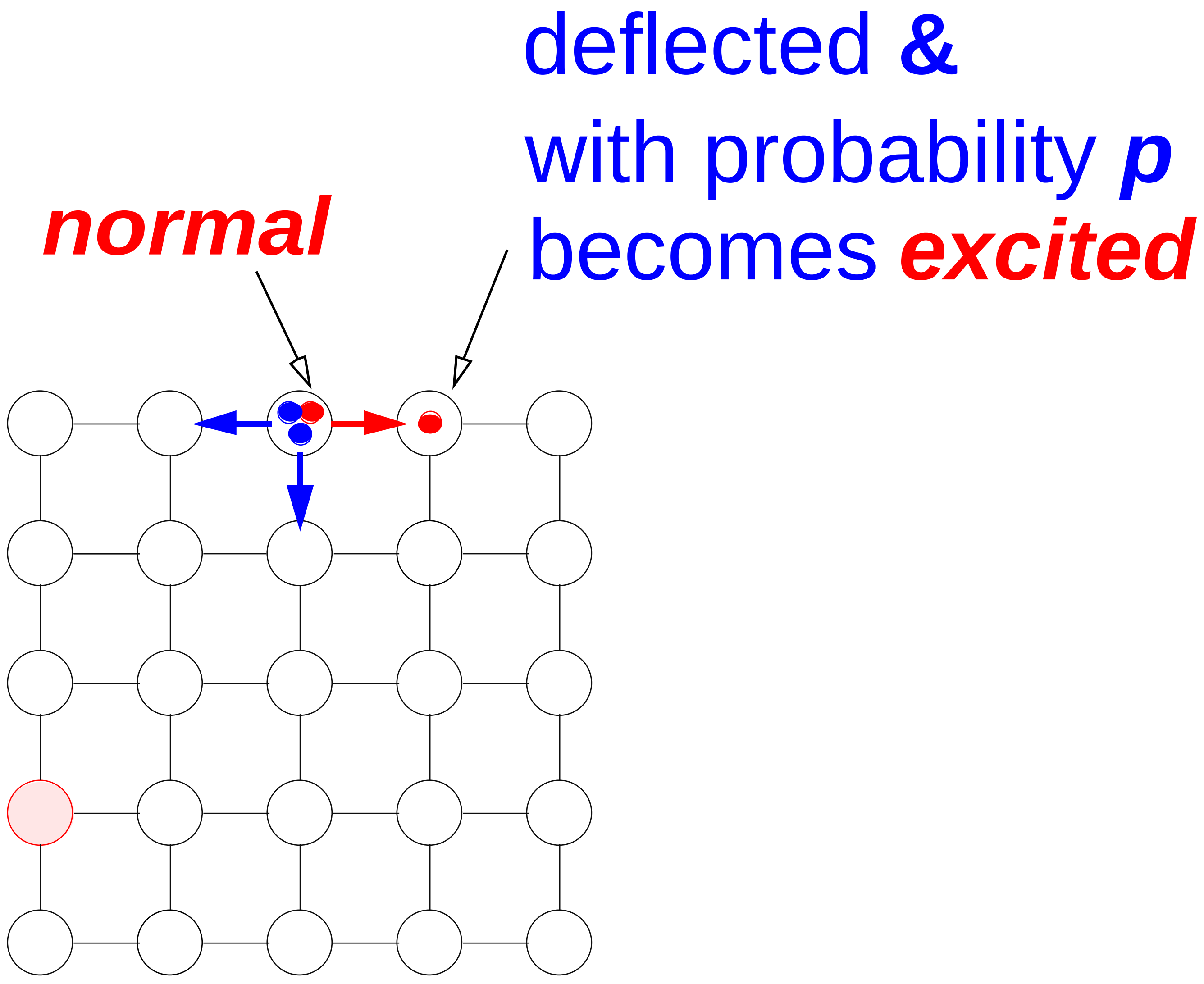
Packet States	Priority
running	↑ higher
excited	
normal	lower

Conflict

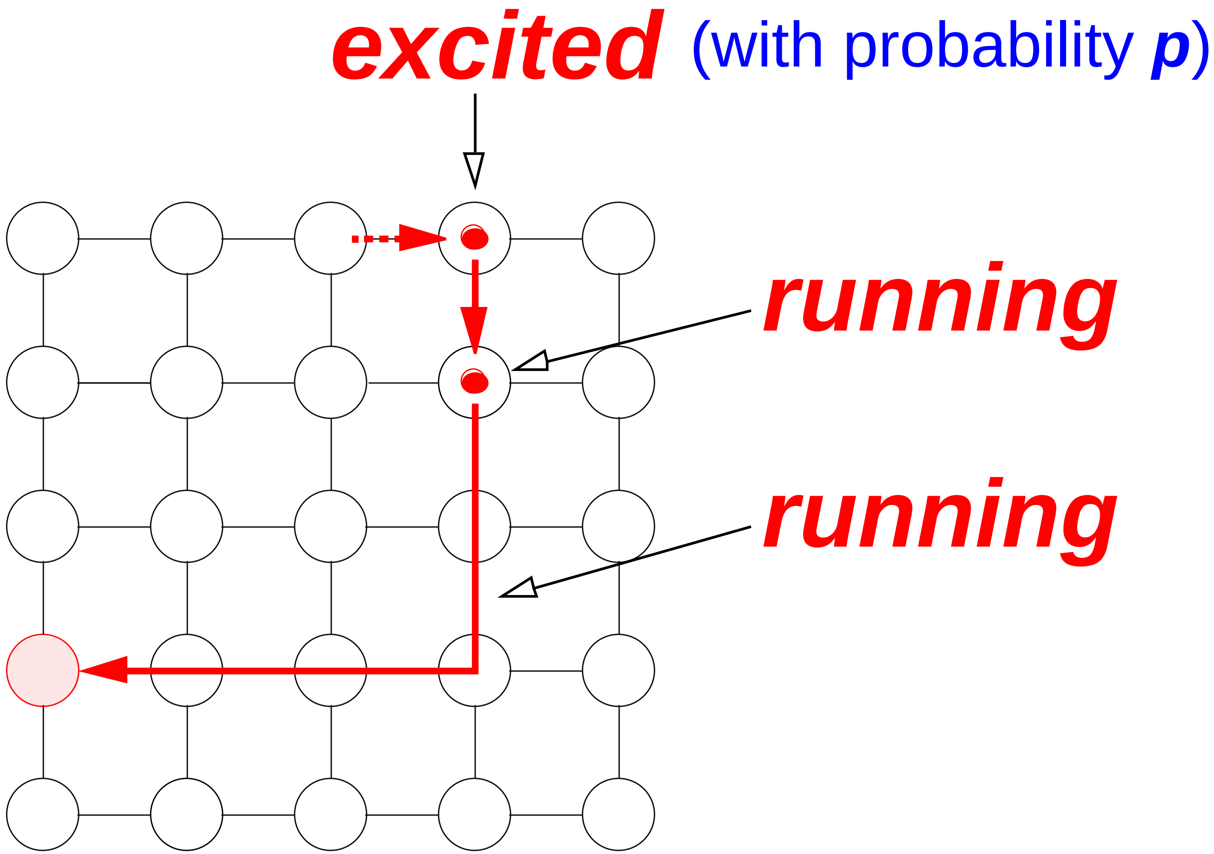


Initially




deflected **&**
with probability ***p***
becomes ***excited***
normal

Home run



If interrupted it becomes ***normal***

Time Analysis for a Packet

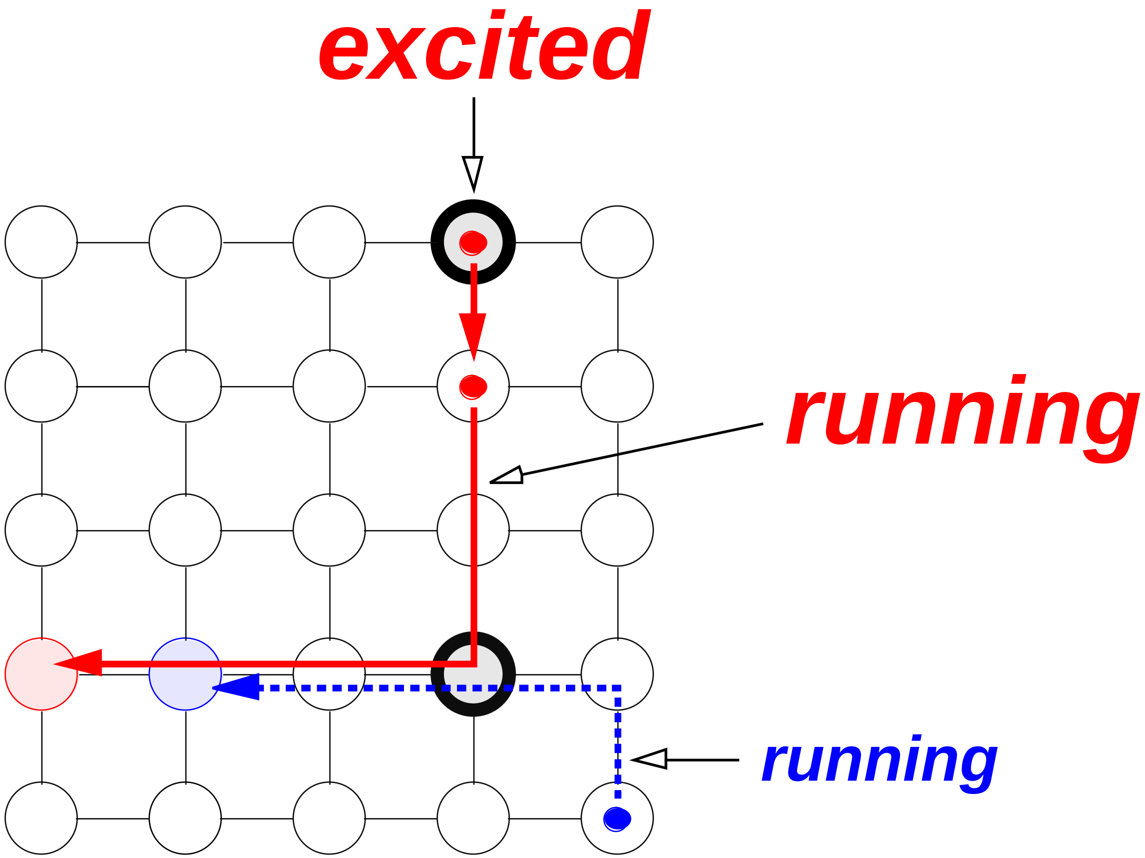
Permutation, Random destinations

Total time until a packet reaches destination:

$O(n)$ expected

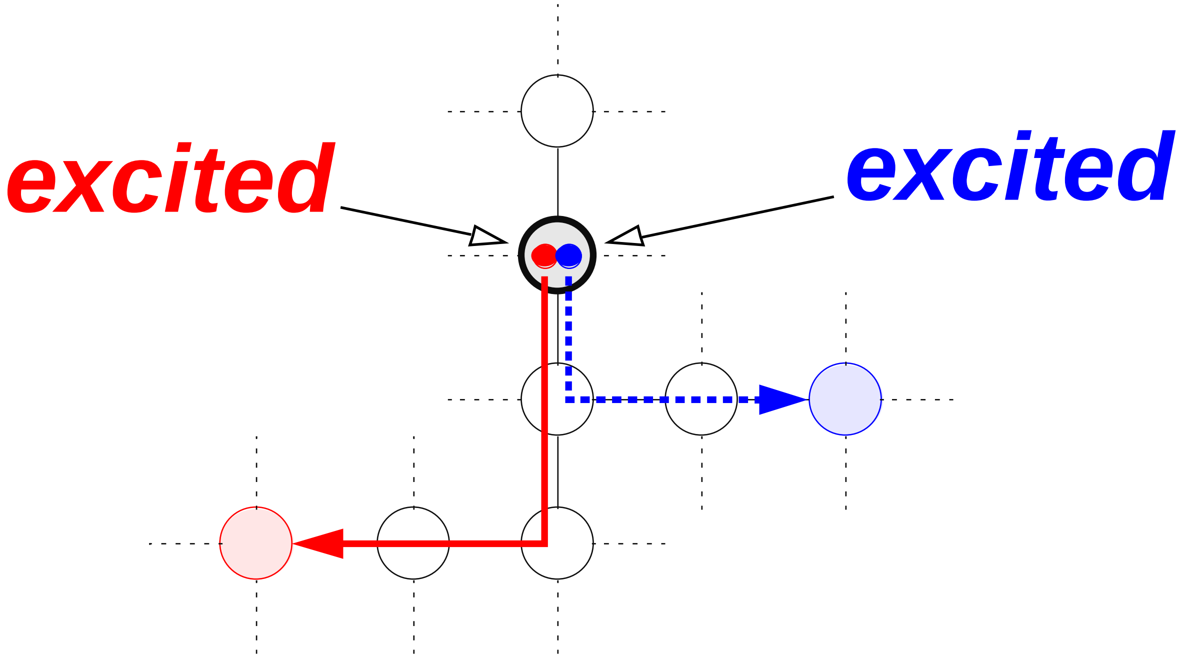
$O(n \cdot \log n)$ with high probability

Interrupting a home run



Interrupting a home run

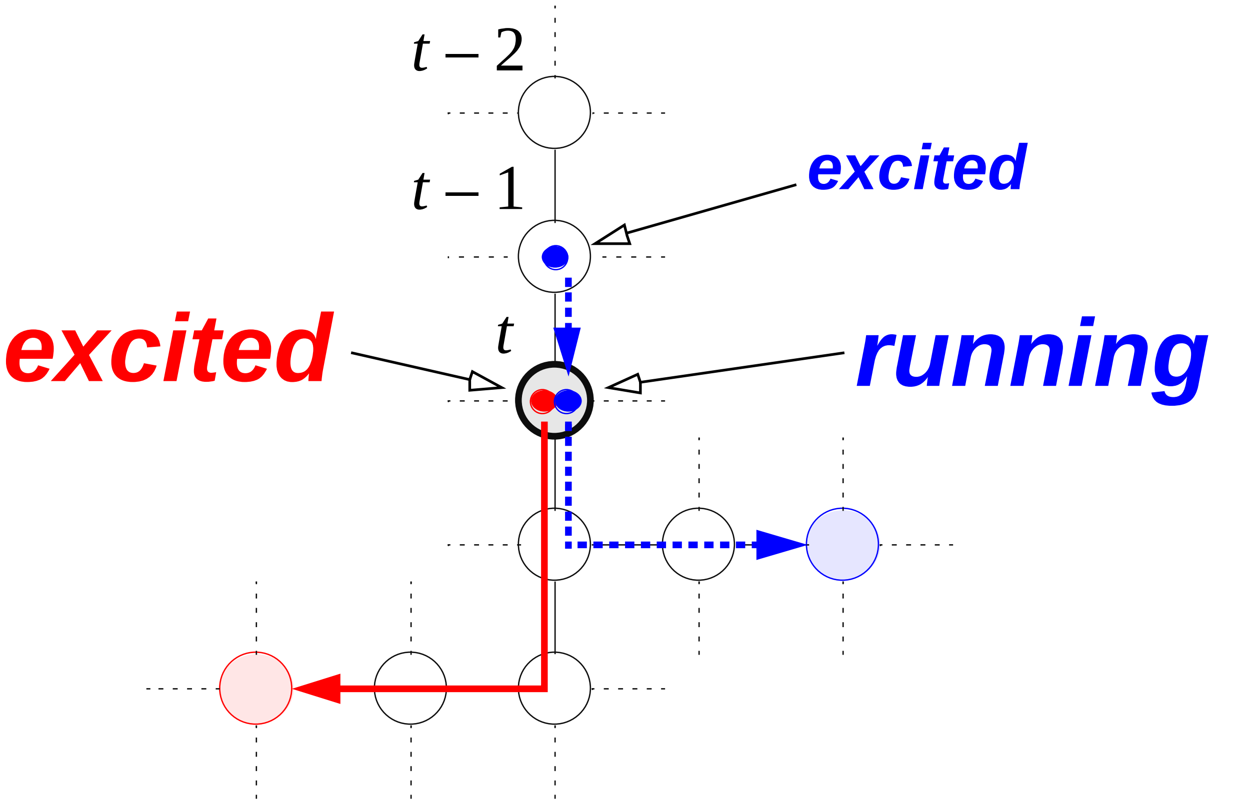
Case1: ***excited*** conflicts ***excited***



Probability of no conflict $\geq (1-p)^3$

Interrupting a home run

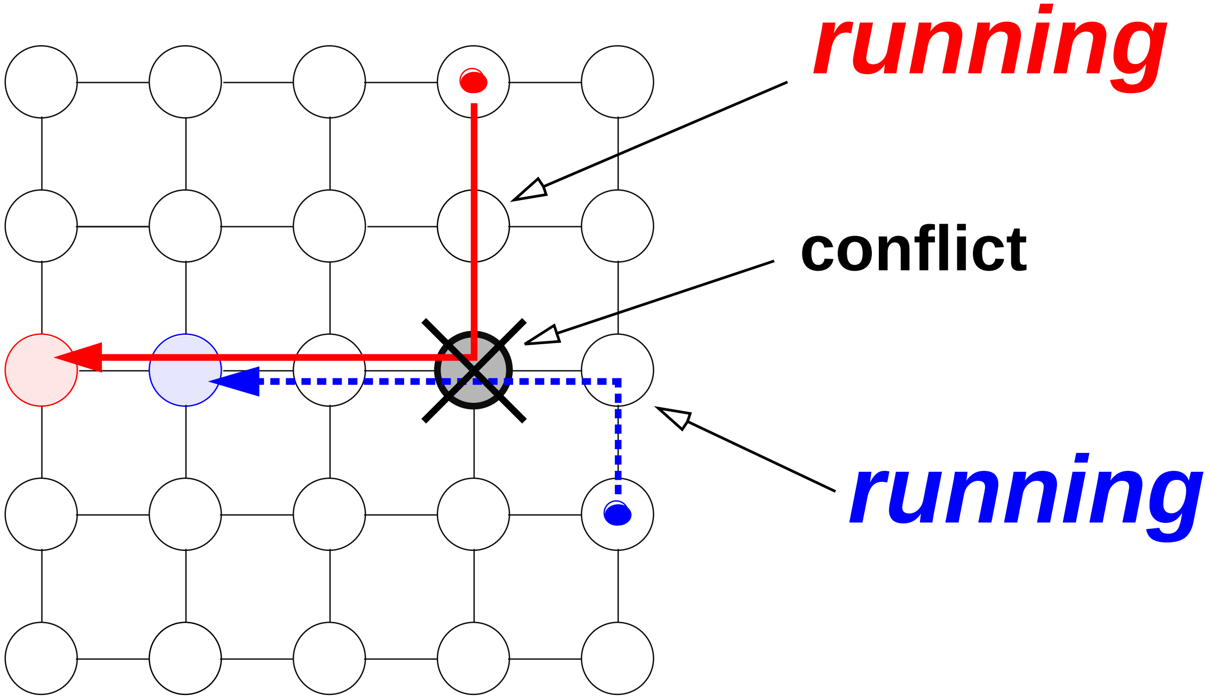
Case 2: ***excited*** conflicts ***running***



Probability of no conflict $\geq (1 - p)^{4 \cdot n}$

Interrupting a home run

Case 3: ***running*** conflicts ***running***



Probability of no conflict $\geq (1-p)^n$

Succeeding in a home run

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

Probability of success:

$$(1-p)^3 \cdot (1-p)^{4 \cdot n} \cdot (1-p)^n \geq \frac{1}{c}$$

case 1 case 2 case 3 constant

Success after a deflection

deflected → ***excited*** → successful home run

Probability of success: $p \cdot \frac{1}{c} = \frac{1}{n} \cdot \frac{1}{c}$

Expected deflections: $n \cdot c = O(n)$

Total time for a packet

Expected deflections $O(n)$ $\Rightarrow$ Expected total time $O(n)$

With similar analysis:

Total time: $O(n \cdot \log n)$

With probability $\geq 1 - \frac{1}{n^3}$

Total time for all packets

Permutation:
Random destinations:

$O(n \cdot \log n)$

With high probability $\geq \left(1 - \frac{1}{n^3}\right)^{n^2} \geq 1 - \frac{1}{n}$

Lower bound: $\Omega(n)$ Previous algorithms: $O(n^2)$

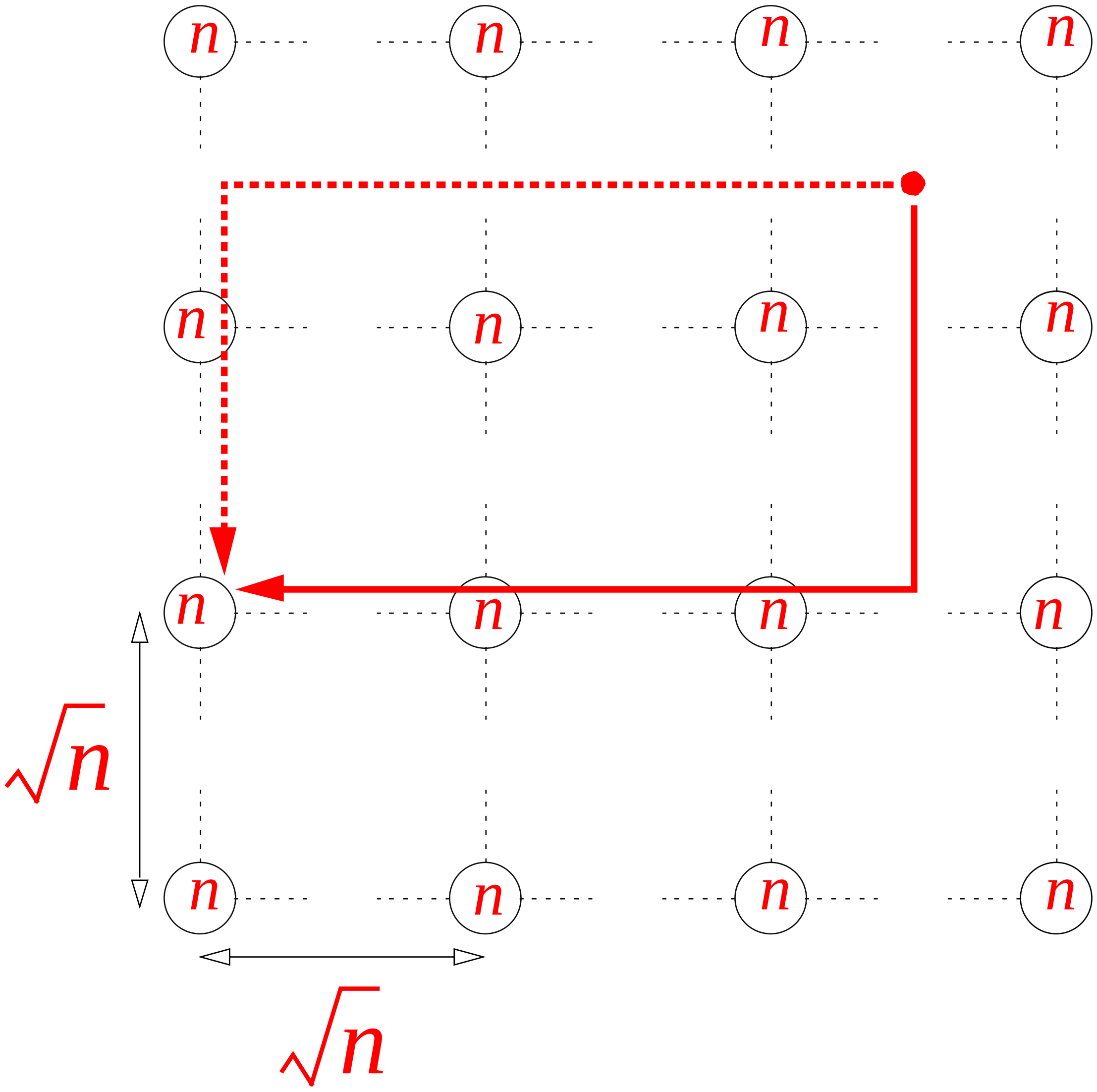
General Batch Problems

Time for all packets:

$$O(m \cdot \log n)$$

m = maximum number of destinations in any row

A pathological case



$m = n\sqrt{n}$

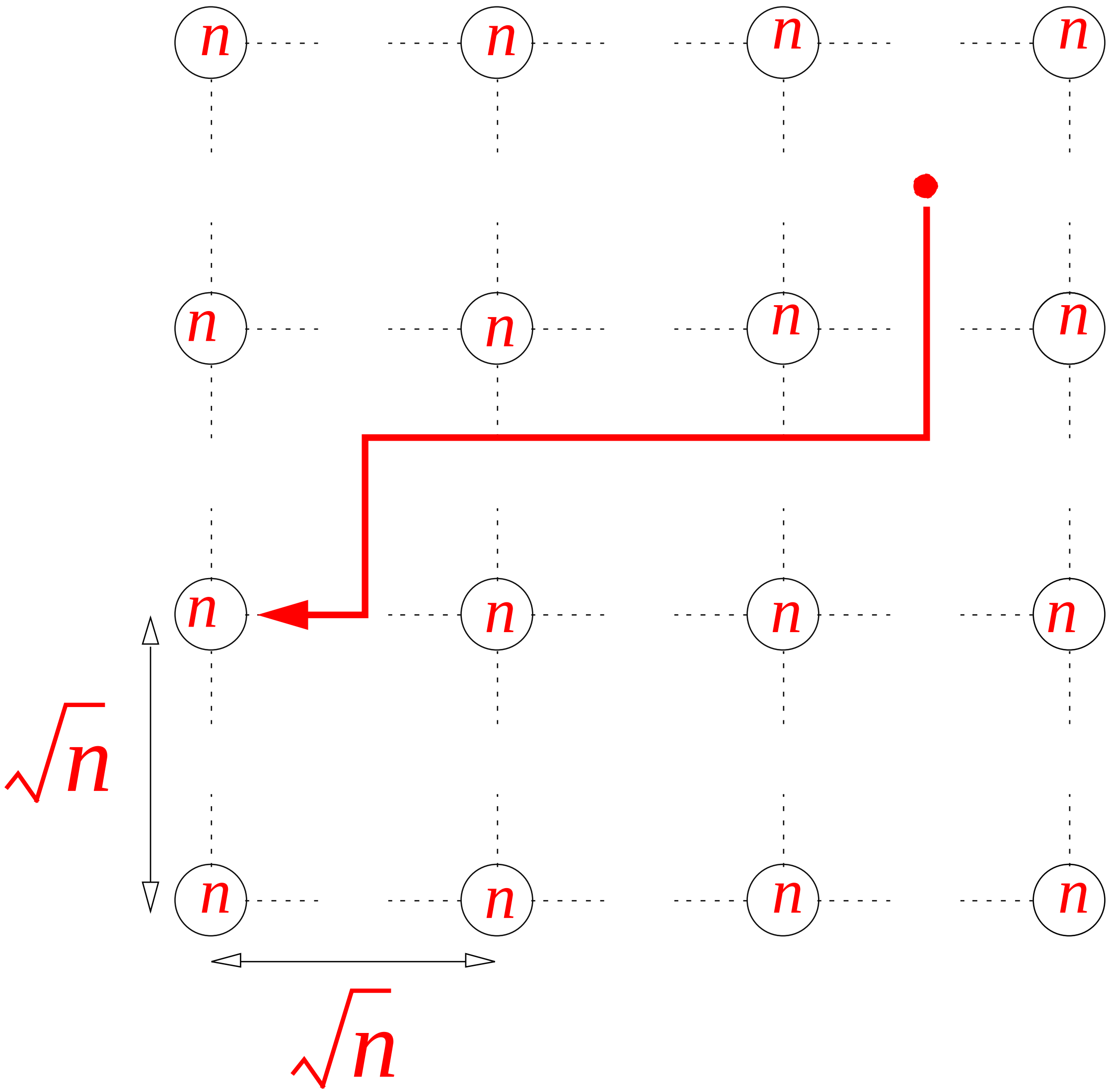
Total time

$O(n\sqrt{n} \cdot \log n)$

Far from

$\Omega(n)$

We need more bends



Total time closer to

$$\Omega(n)$$

Talk Outline

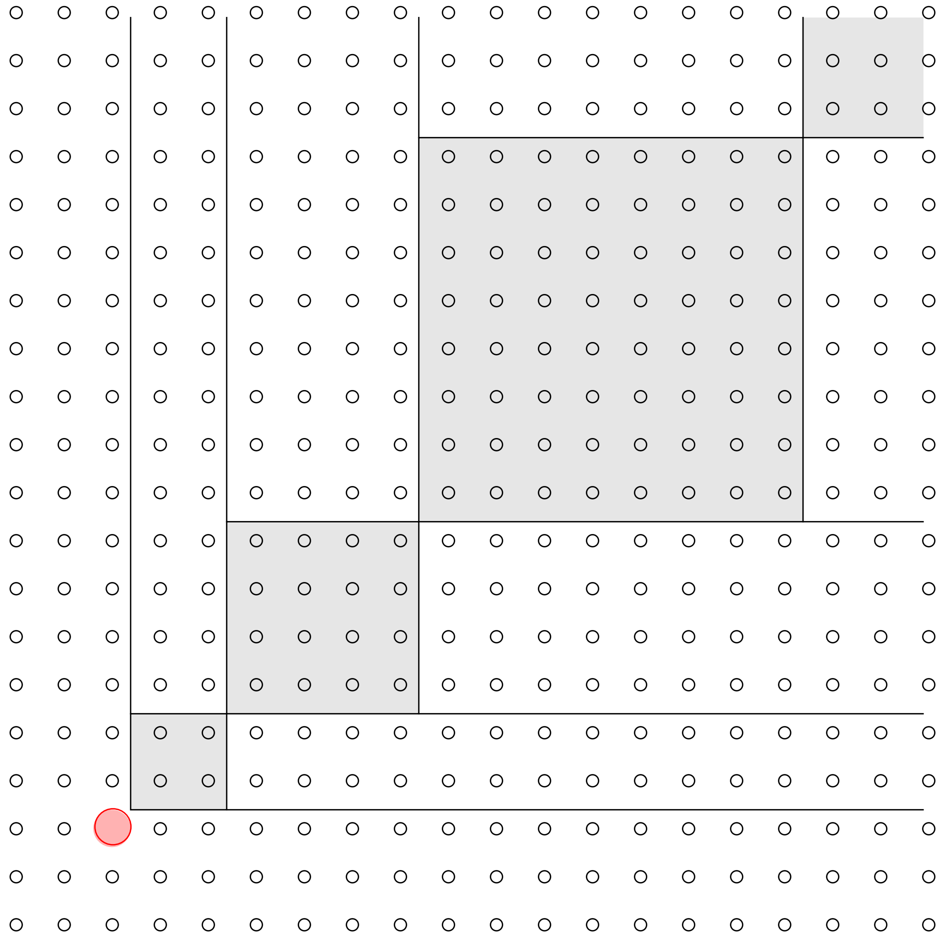
Greedy Hot-Potato Routing

One-Bend Algorithm

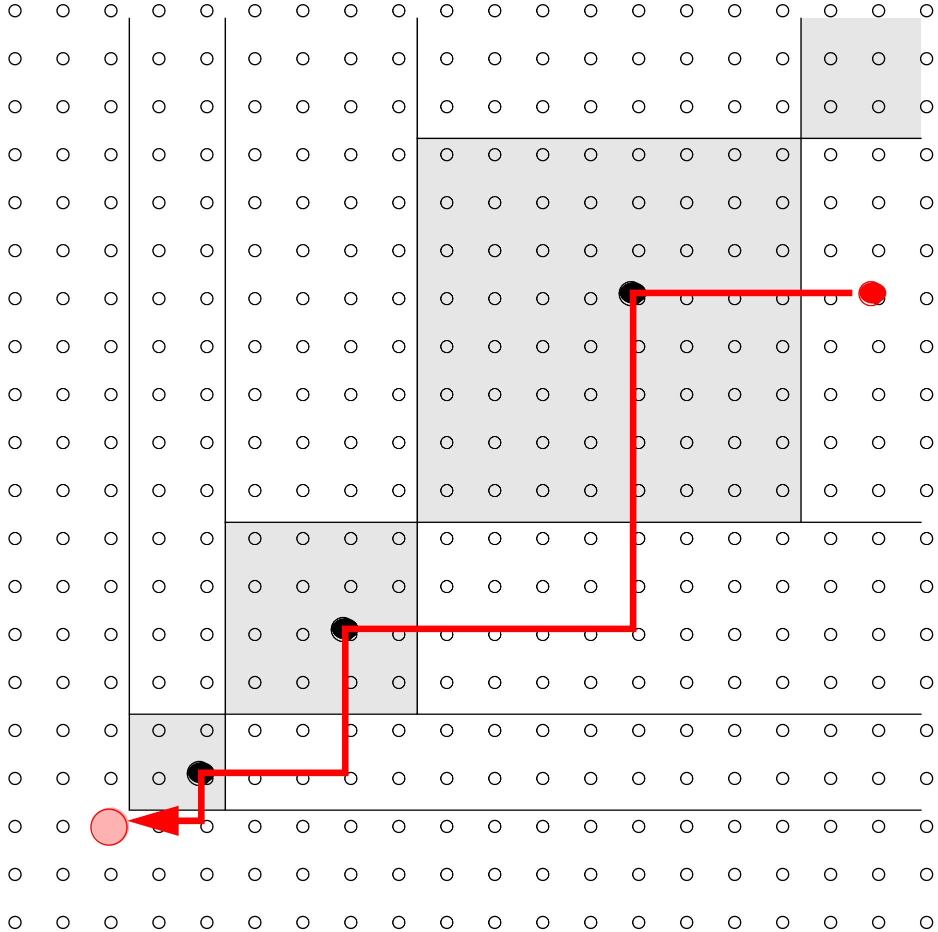
Multi-Bend Algorithm

Future Research

Multi-Bend Algorithm



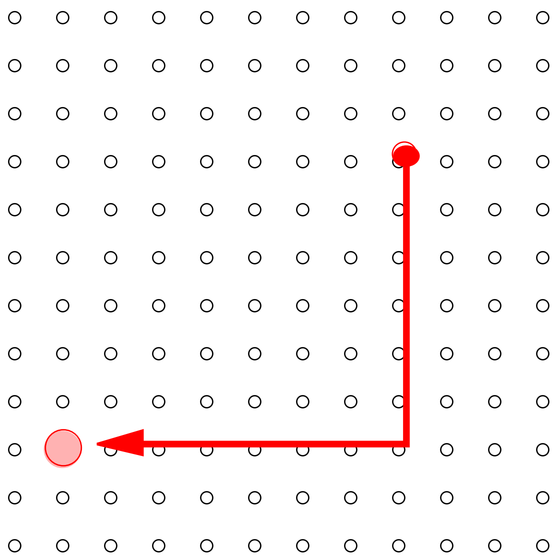
Home Run



Random intermediate nodes

Lower bound
for general batch problems

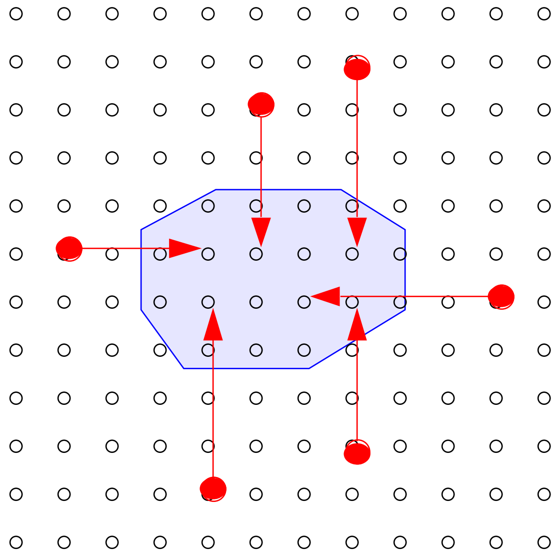
Initial distance from destination



MAX

D

Lower bound: bandwidth



$$\text{MAX} \quad W = \frac{\text{incoming packets}}{\text{incoming links}}$$

Lower bound

$$L = \Omega(D + W)$$

initial distance from destination → D

bandwidth → W

"Hard" general problem instances:

$$L = \Omega(n)$$

Total time: $O(L \cdot \log^3 n)$

Talk Outline

Greedy Hot-Potato Routing

One-Bend Algorithm

Multi-Bend Algorithm

Future Research

Future Research

More dimensions

"Easy" general problem instances: $o(n)$

Dynamic analysis

Arbitrary network topologies