

Universal Bufferless Routing

Marios Mavronicolas

University of Cyprus

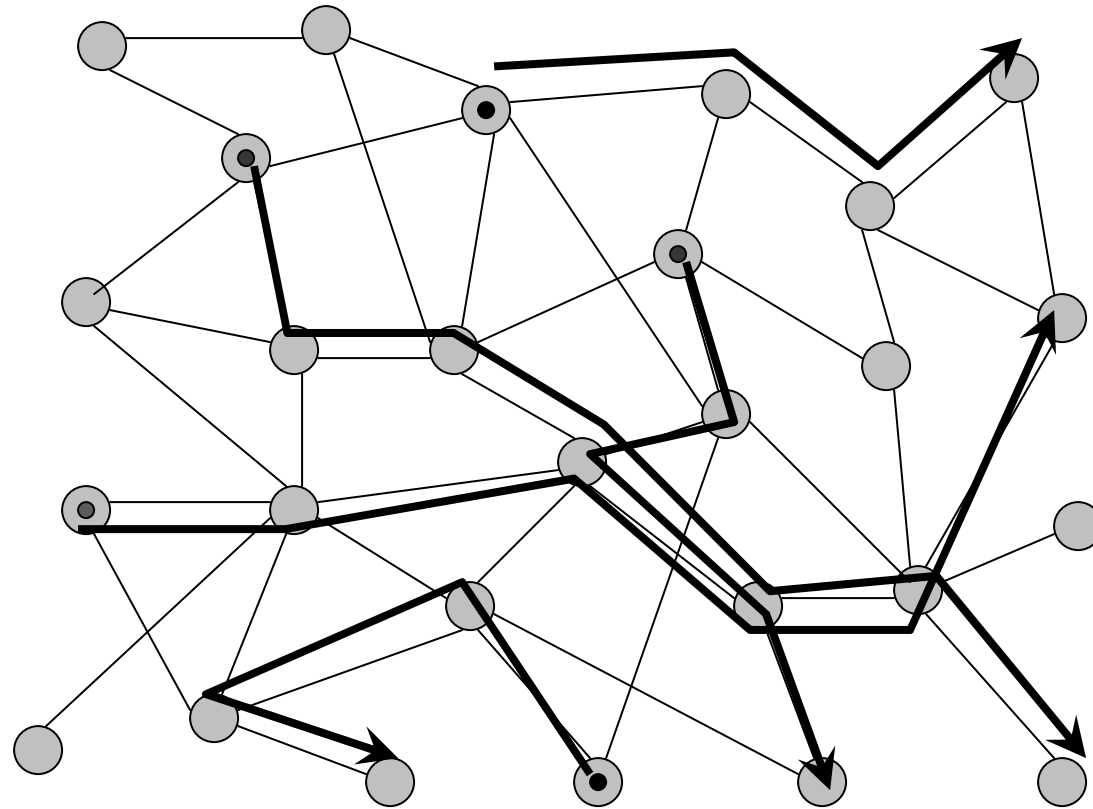
Costas Busch

Rensselaer Polytechnic Inst.

Malik Magdon Ismail

Rensselaer Polytechnic Inst.

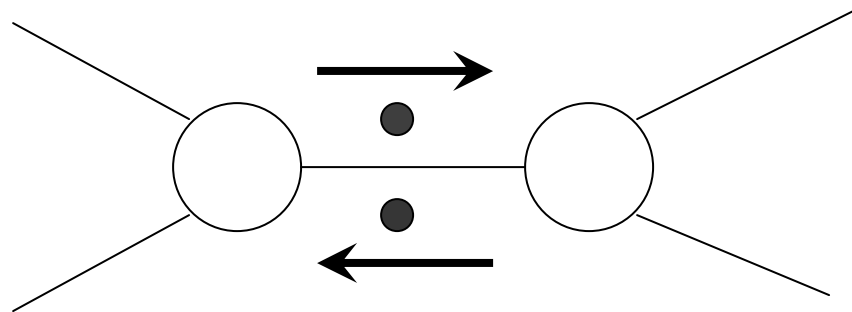
Routing Problem



- Arbitrary network
- Arbitrary set of packets

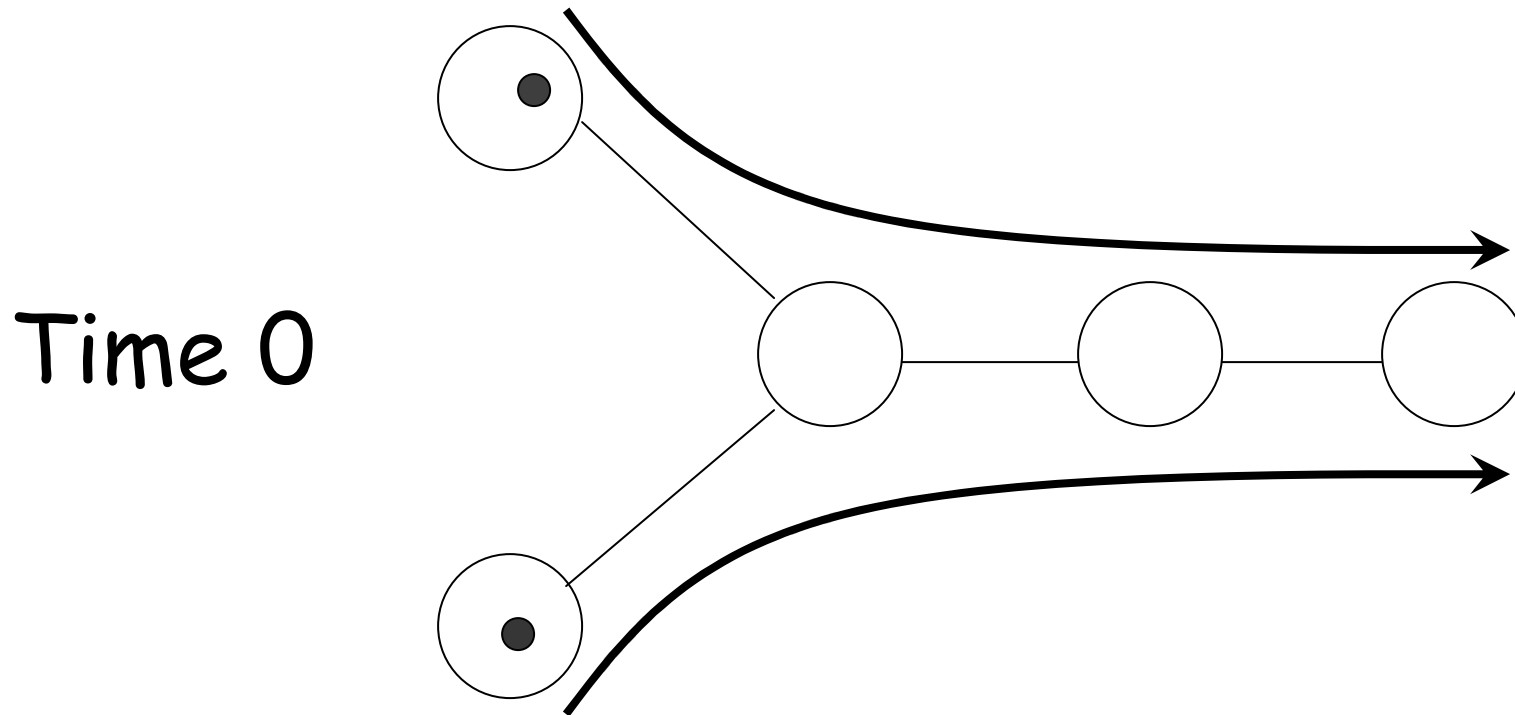
Network Model

- Synchronous network (time steps)
- One packet per time step
- Bi-directional links



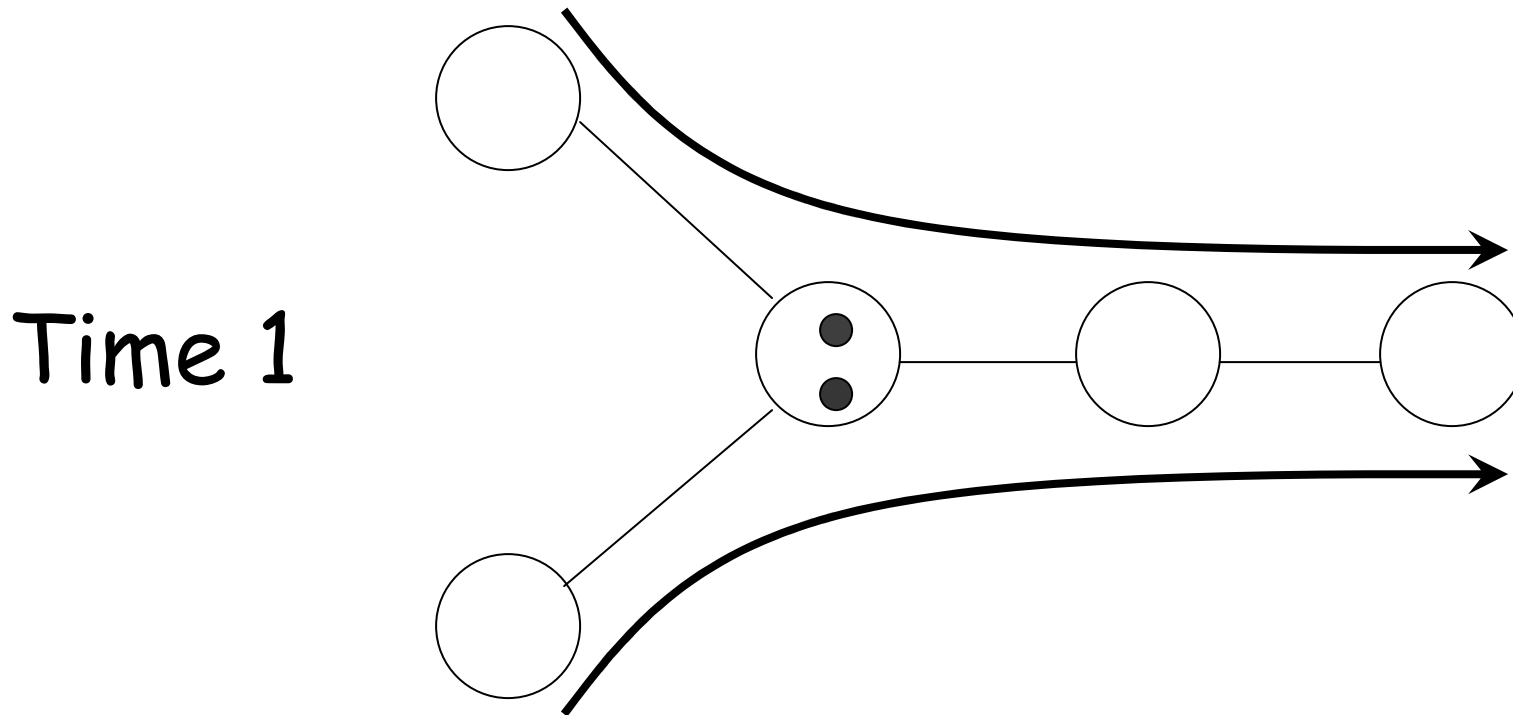
Network Model

Buffer-less nodes



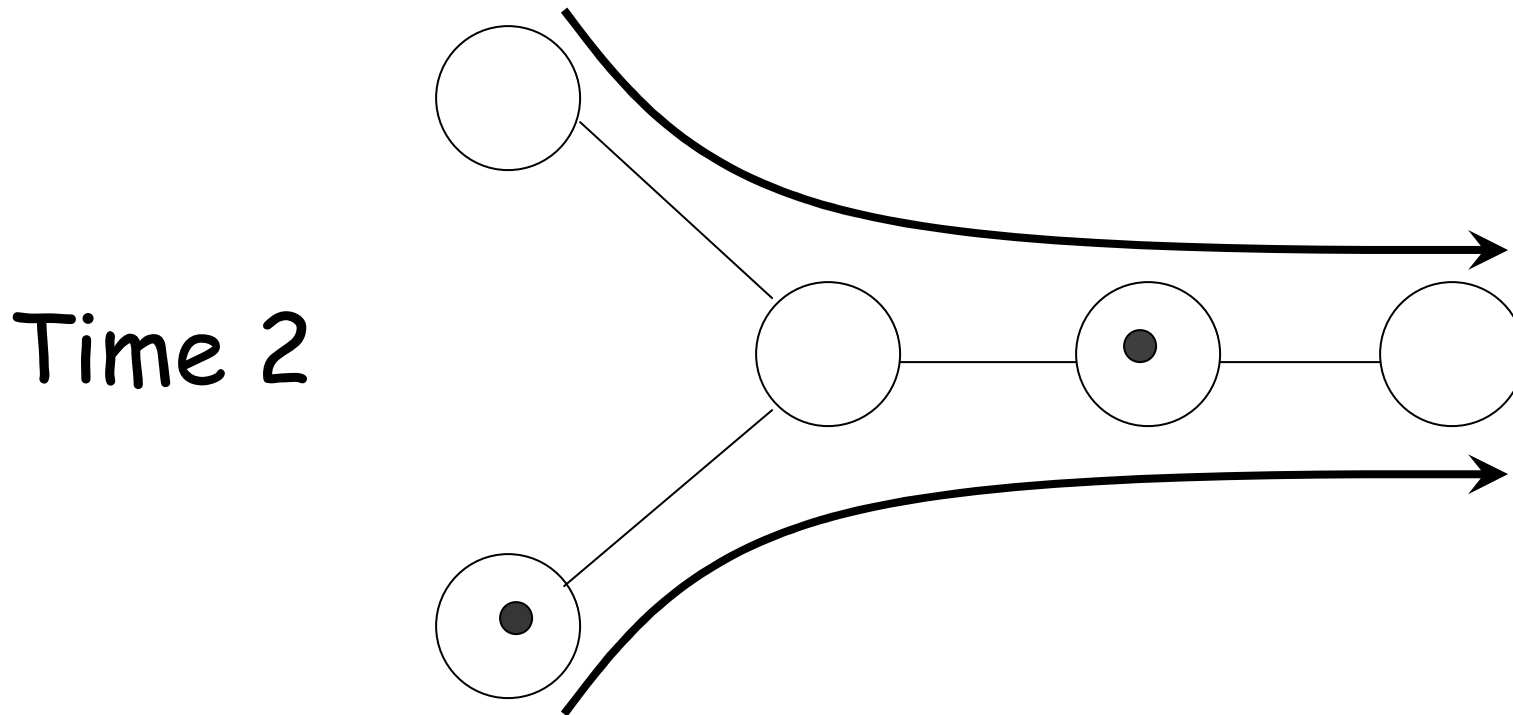
Packets are always moving

Buffer-less nodes



Packets are always moving

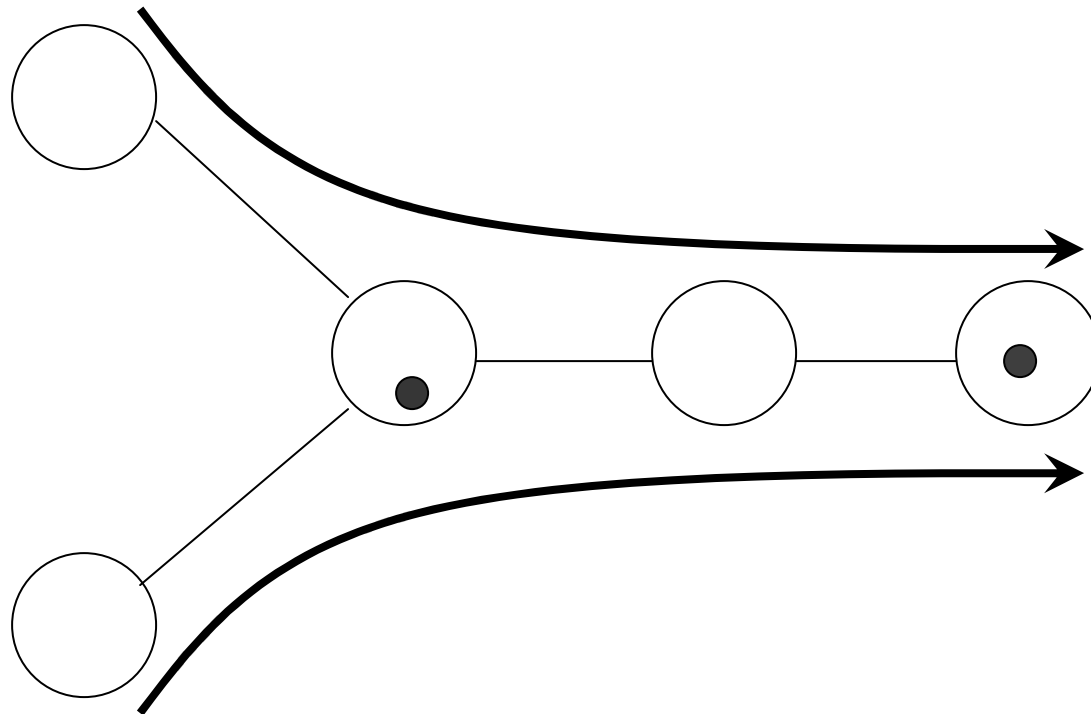
Buffer-less nodes



Packets are always moving

Buffer-less nodes

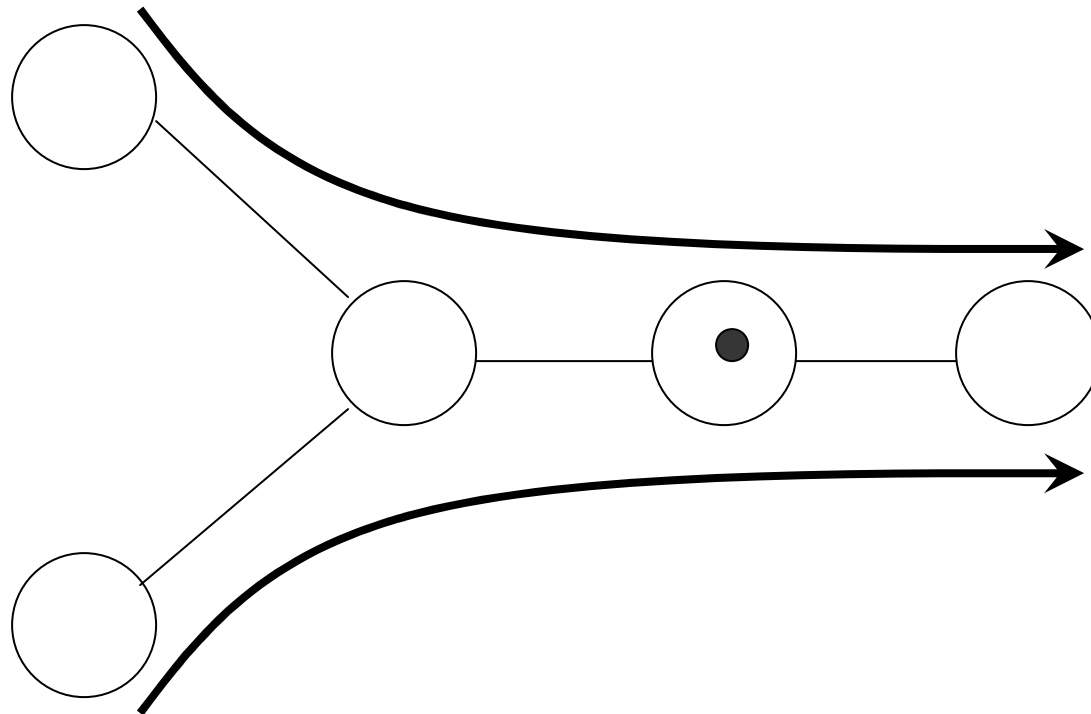
Time 3



Packets are always moving

Buffer-less nodes

Time 4



Packets are always moving

Bufferless routing is interesting:

- Optical networks
- Simple hardware implementations
- Works well in practice:

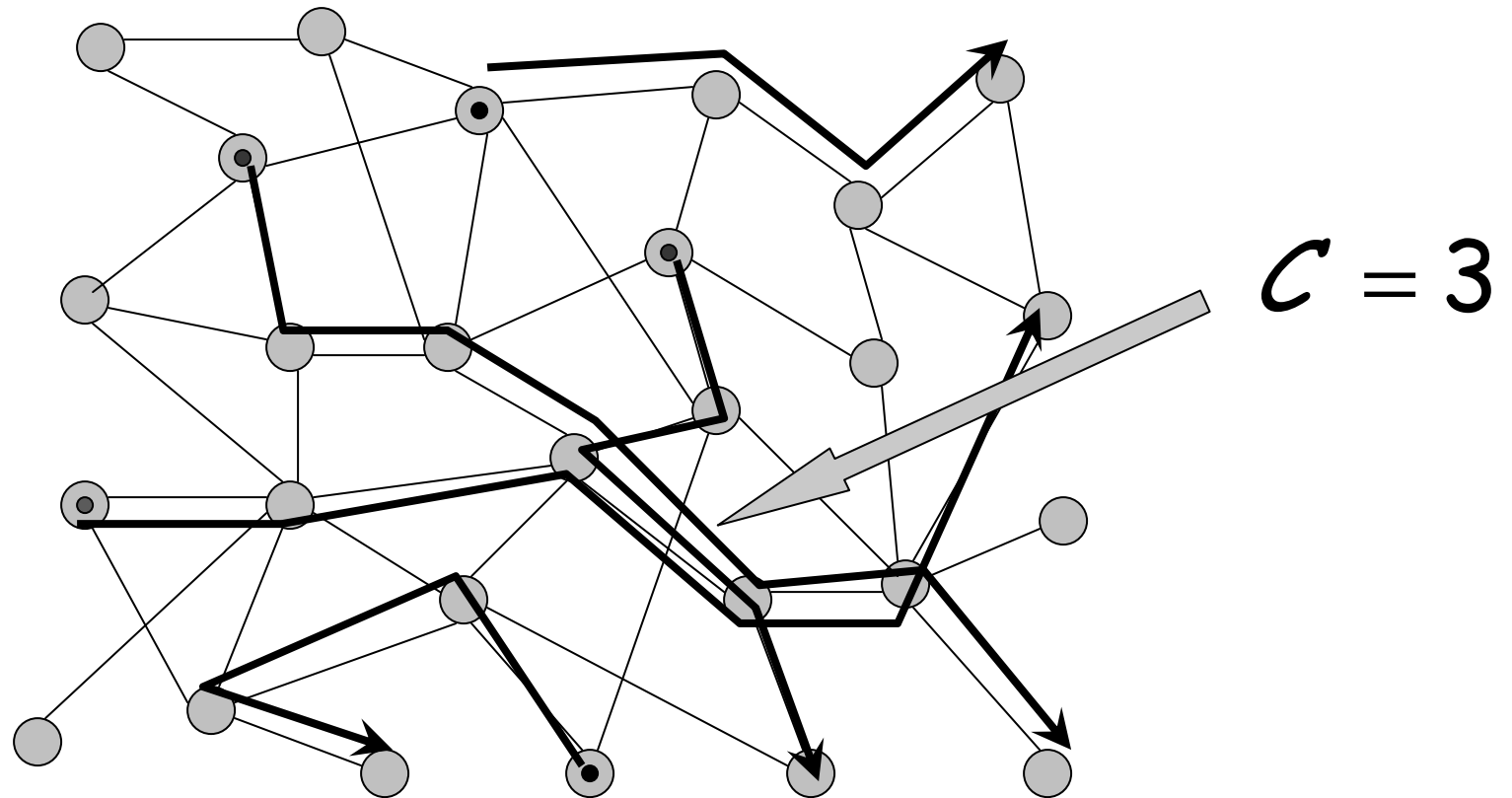
Bartzis et al: EUROPAR 2000

Maxemchuck: INFOCOM 1989

Routing Time: the time until the last Packet is absorbed

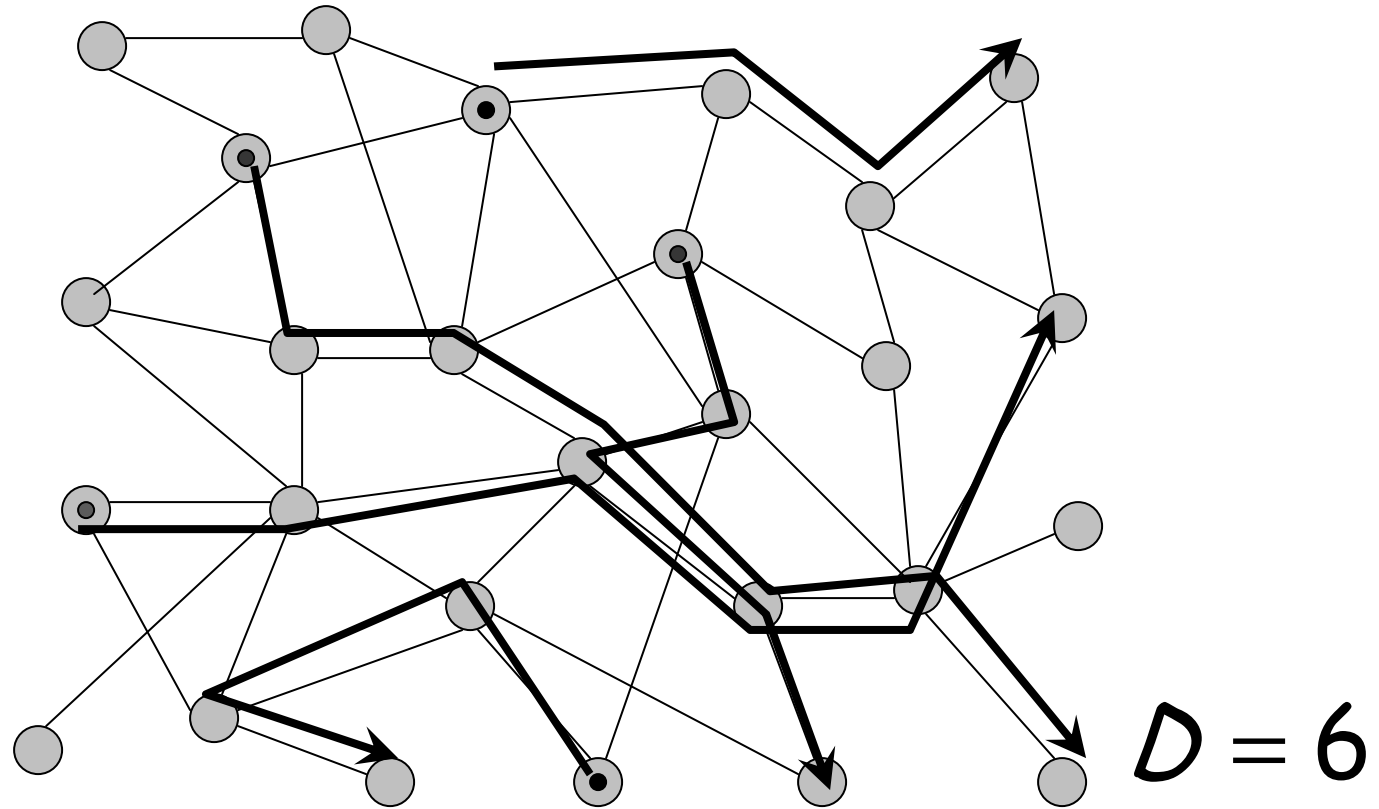
Objective: Minimize Routing Time

Congestion



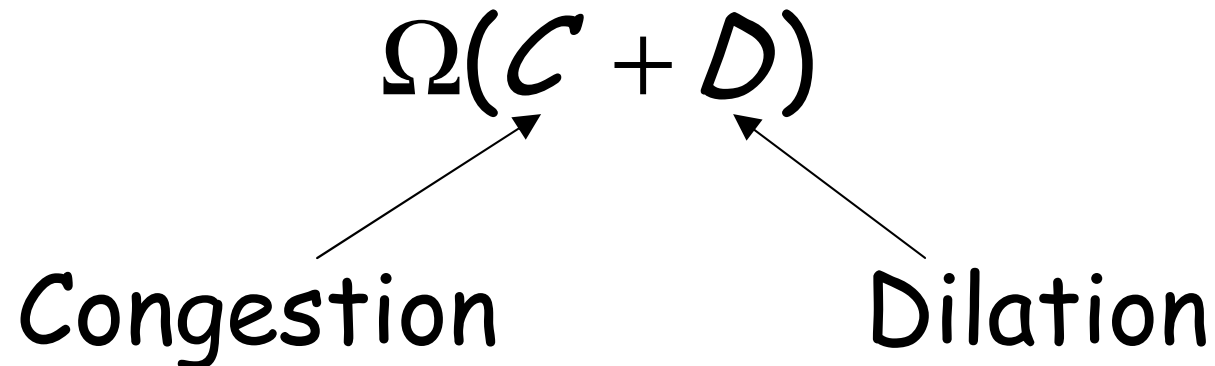
Maximum number of packets
that cross an edge

Dilation



Maximum path length

Lower bound on Routing Time:



We want to find algorithms with routing time close to lower bound

Our contribution:

A buffer-less algorithm with
routing time

$$O((C + D) \cdot \log^3(n + N))$$

nodes

packets

Related Work

For routing with buffers
there exist optimal $O(C + D)$ algorithms:

[Leighton - Maggs - Rao, Combinatorica 94]

[BS99, LMR99, MV99, OR97, RT96]

What about buffer-less routing?

Related Work

Buffer-less Routing has been studied for specific topologies

Mesh	[BRST93, BES97, BHS98, BU96, BHW00]
Hypercube	[BH85, BC95, FR92, H91]
Trees	[BMMW04, RSW00]
Leveled	[BBPRRS96, B02]
Vertex-symmetric	[MS95]

No previous work for $O(C + D)$ time on arbitrary networks

Presentation Outline

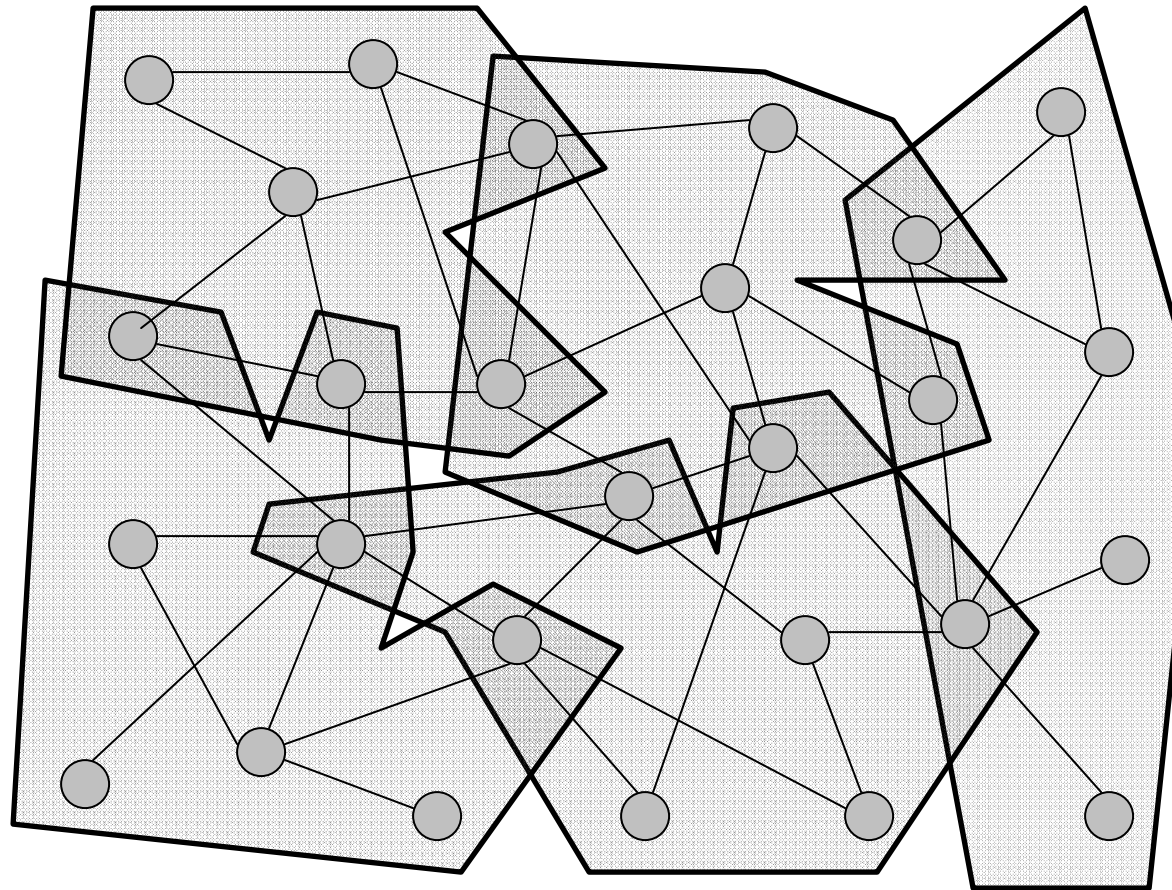


• Algorithm

• Analysis

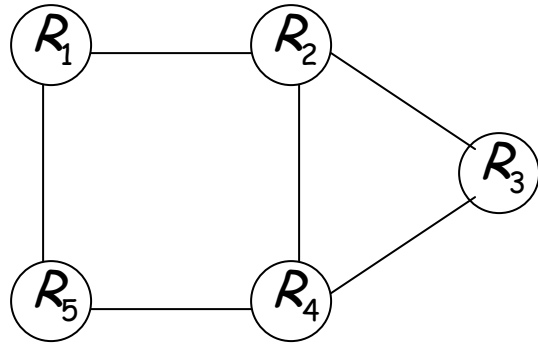
• Conclusion

Partition the Network into Regions

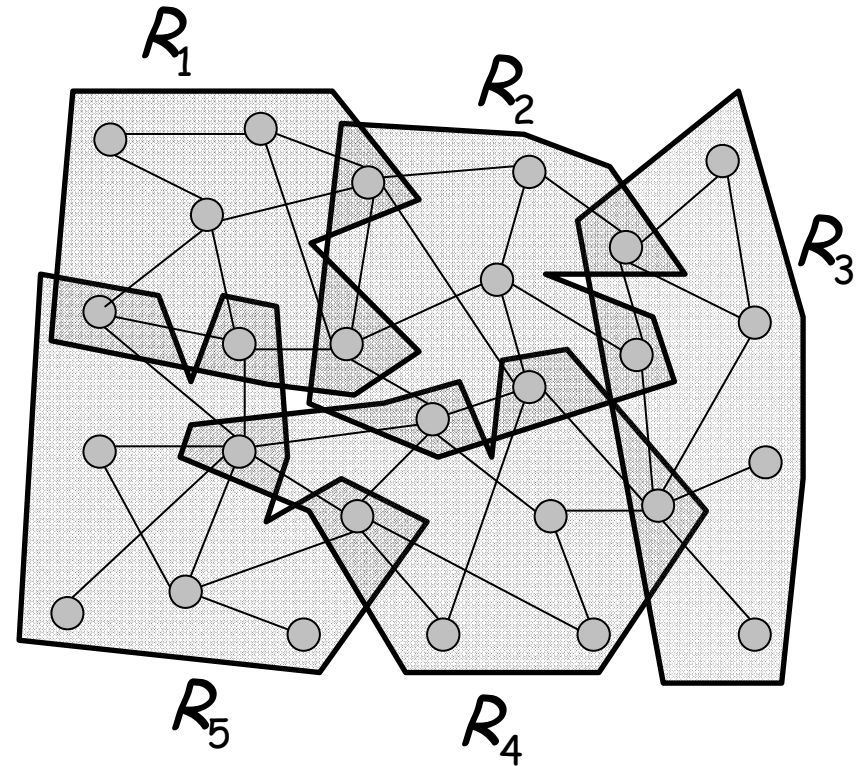


- Each edge belongs to only one region
- Regions share nodes

Construct Region Graph

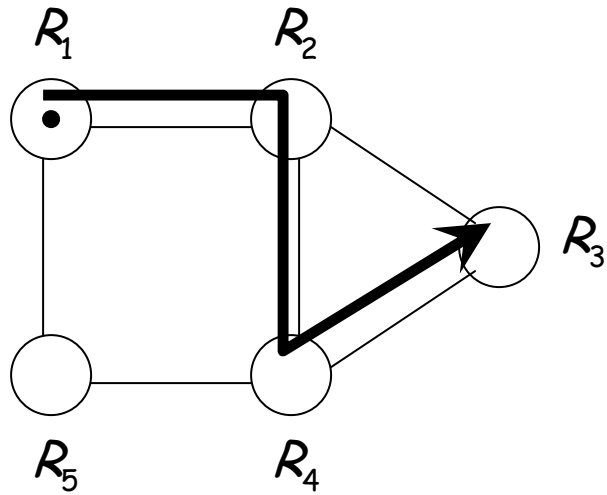


Region Graph

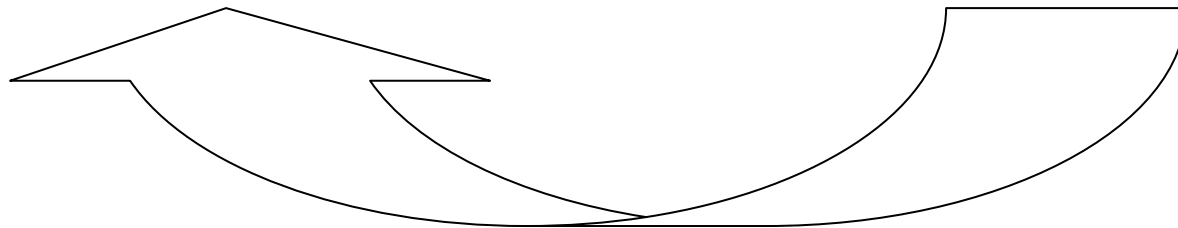
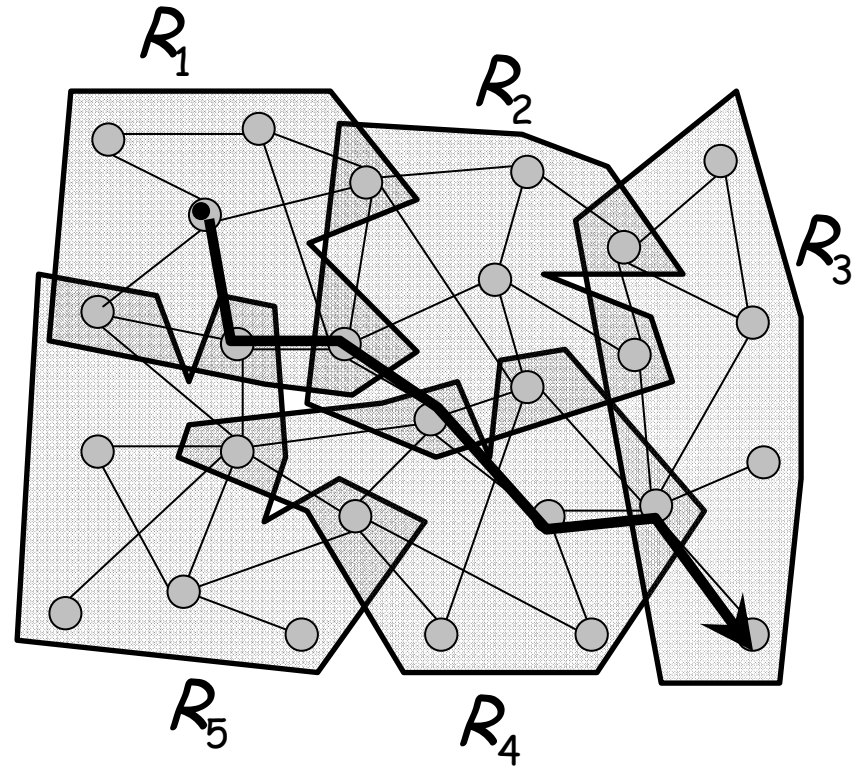


Regions are connected if they share a node in the original graph

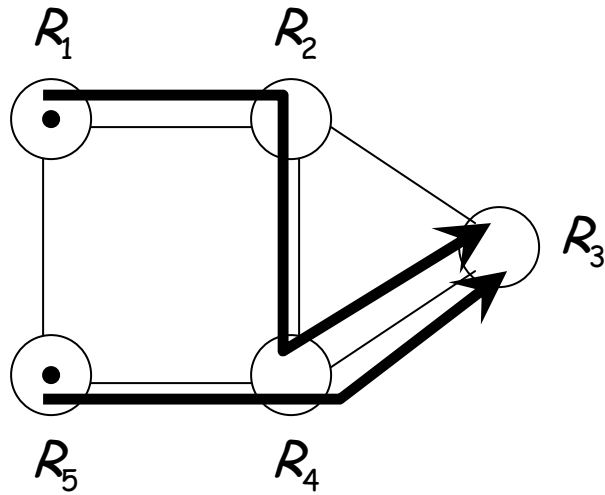
Translate Packet Paths



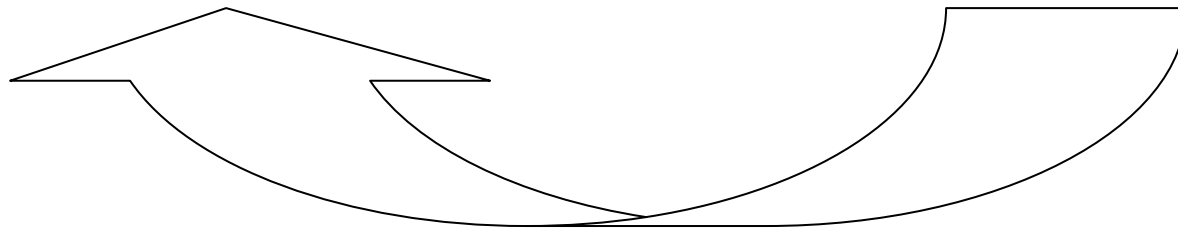
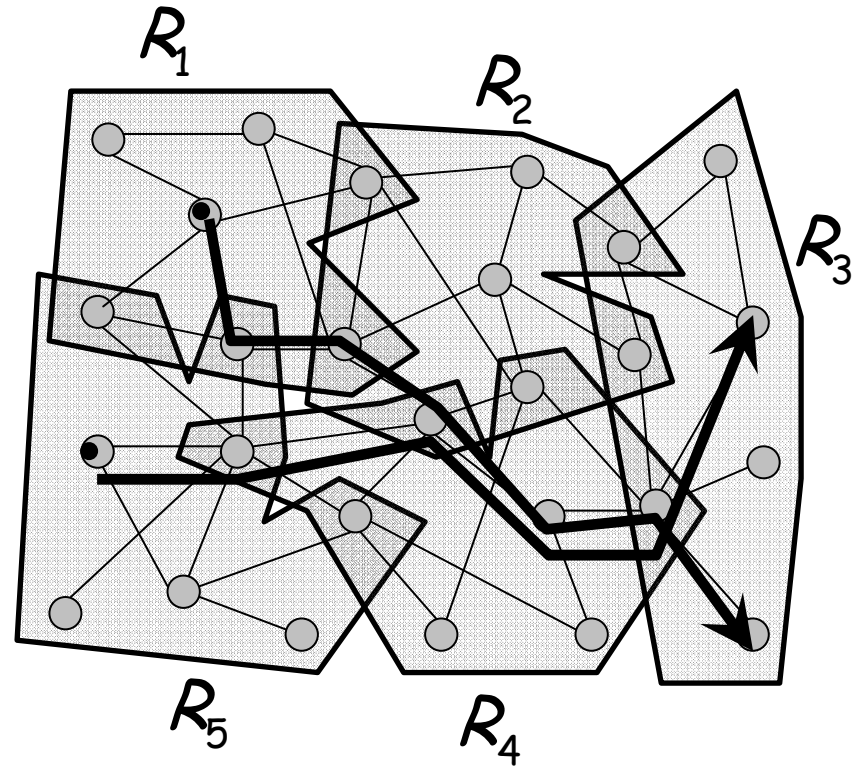
Region Graph



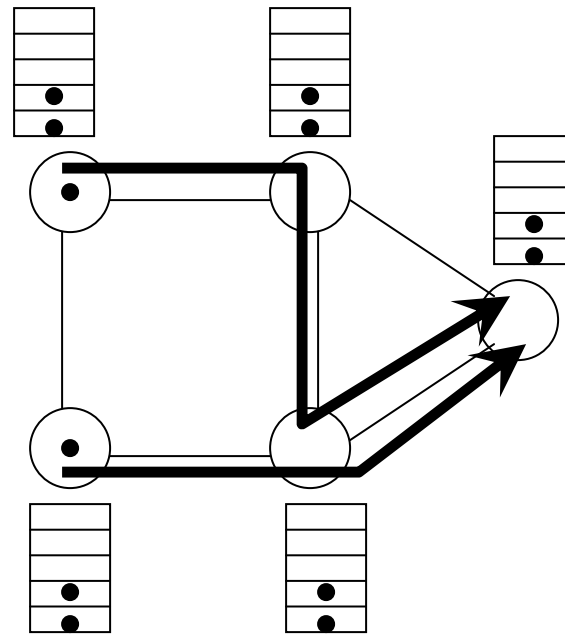
Translate Packet Paths



Region Graph

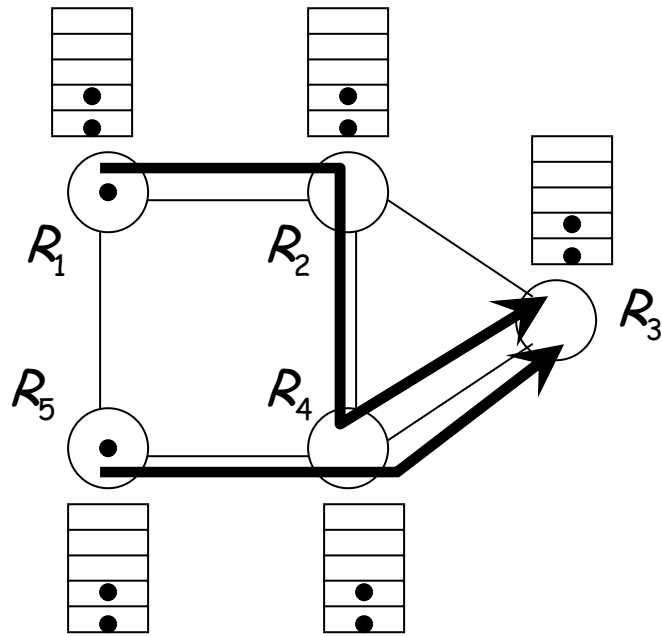


Execute a Buffered Algorithm on Region Graph



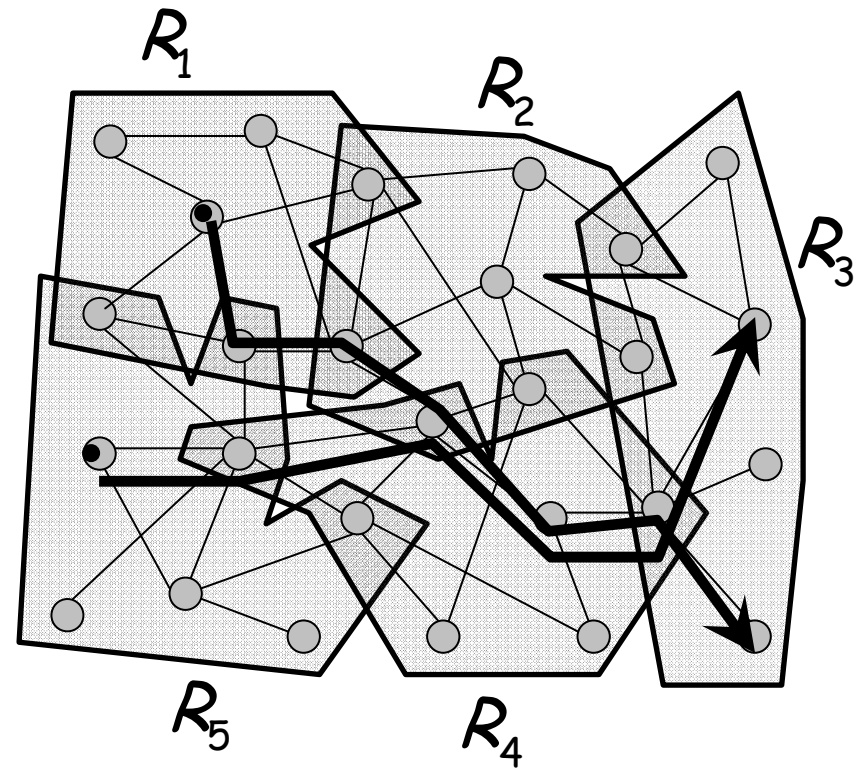
Packet Buffer

Buffered Algorithm



Buffers

Simulate Buffered Algorithm



No Buffers

Region Graph

Buffered Algorithm
Action for a Packet

1 time step

- Buffer in node R_i
- Move from node R_i to R_j
- Inject/Absorb at node R_i

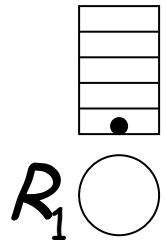
Original Graph

Bufferless Algorithm
Simulation Action

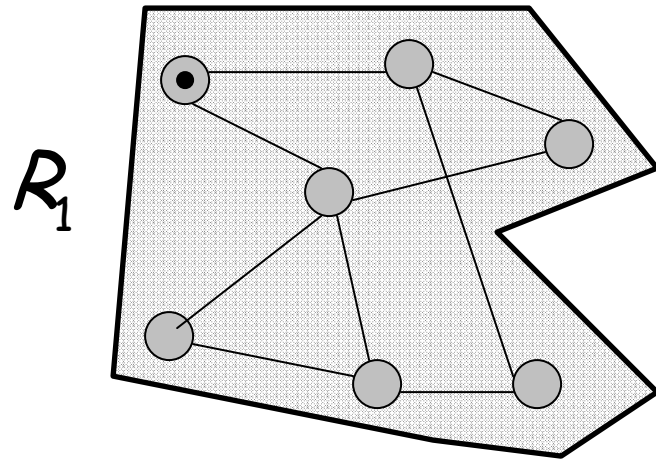
1 phase

- Circular in region R_i
- Move from region R_i to R_j
- Inject/Absorb at source/destination in region R_i

Packet Buffer

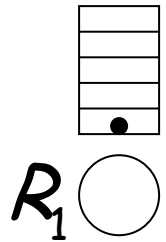


Simulation- Circulation

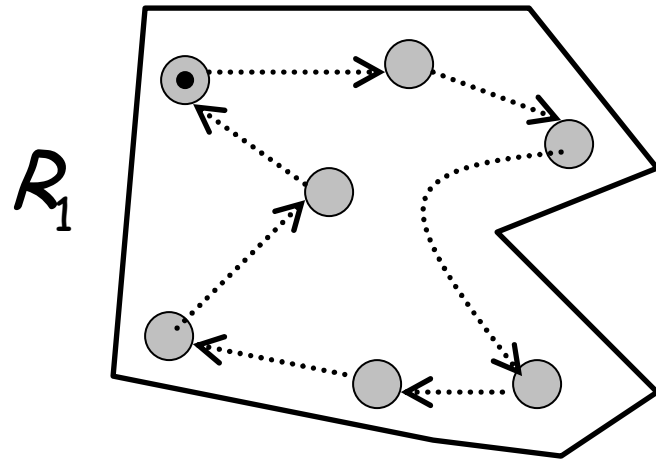


Follow repeatedly
an Euler tour

Packet Buffer

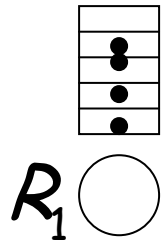


Simulation- Circulation



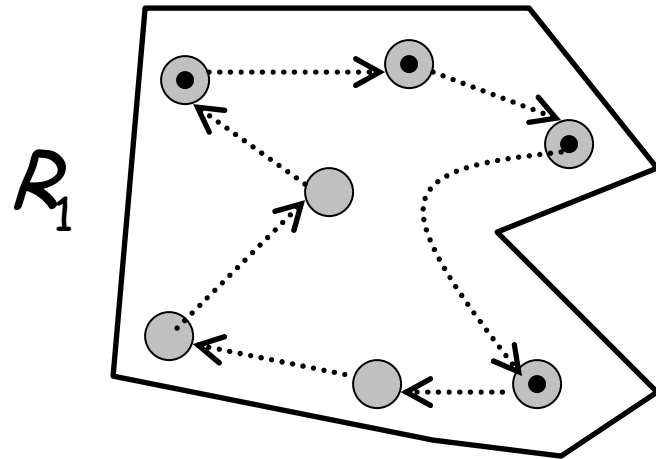
Euler tour
(abstract)

Packet
Buffer



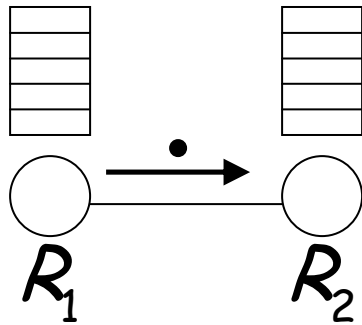
Multiple
packets

Simulation-
Circulation

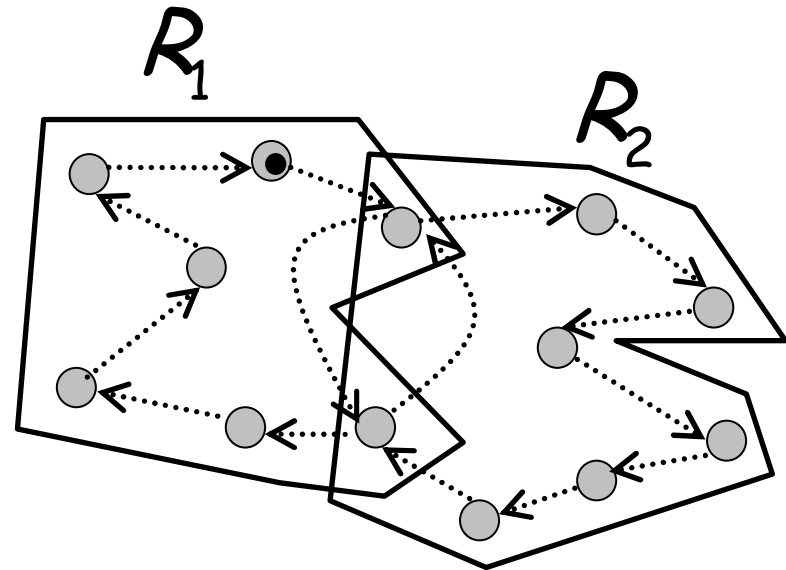


Pipeline packets
on Euler Tour

Packet Move

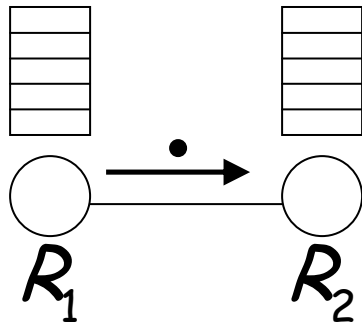


Simulation of move

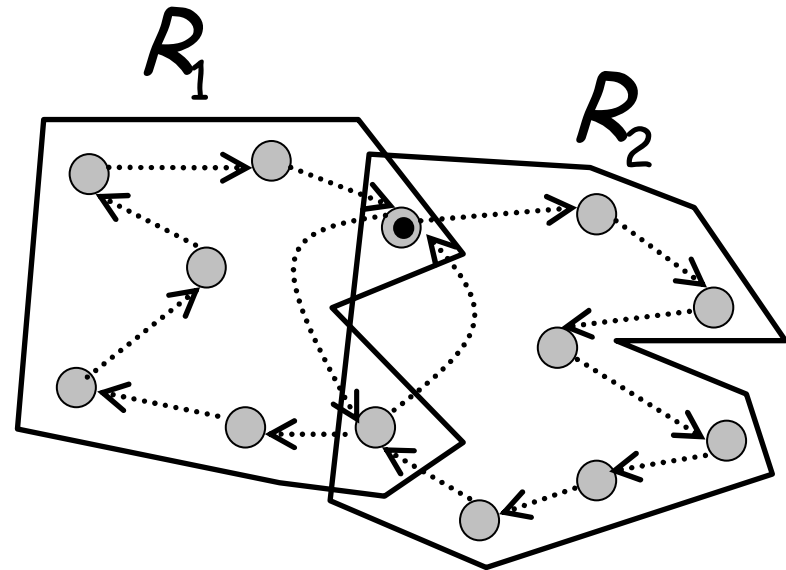


Packet moves to new region when it reaches common node

Packet Move

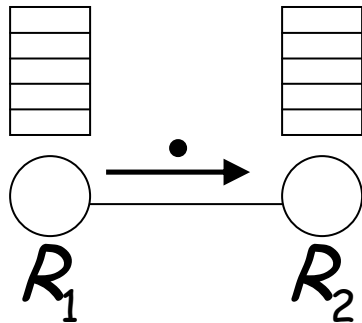


Simulation of move

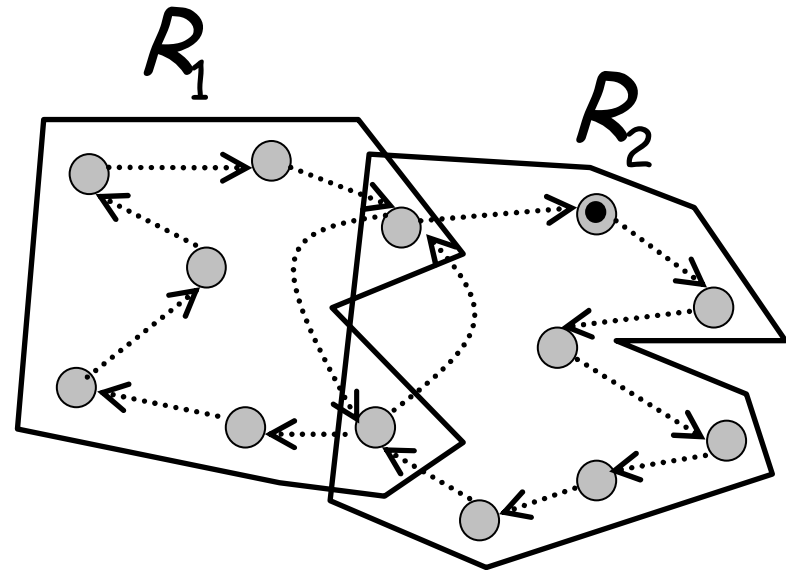


Packet moves to new region when it reaches common node

Packet Move

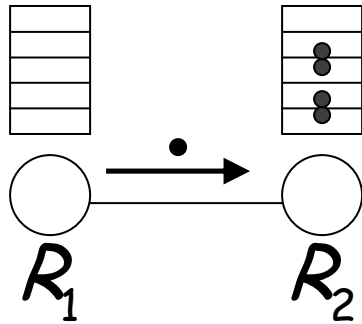


Simulation of move



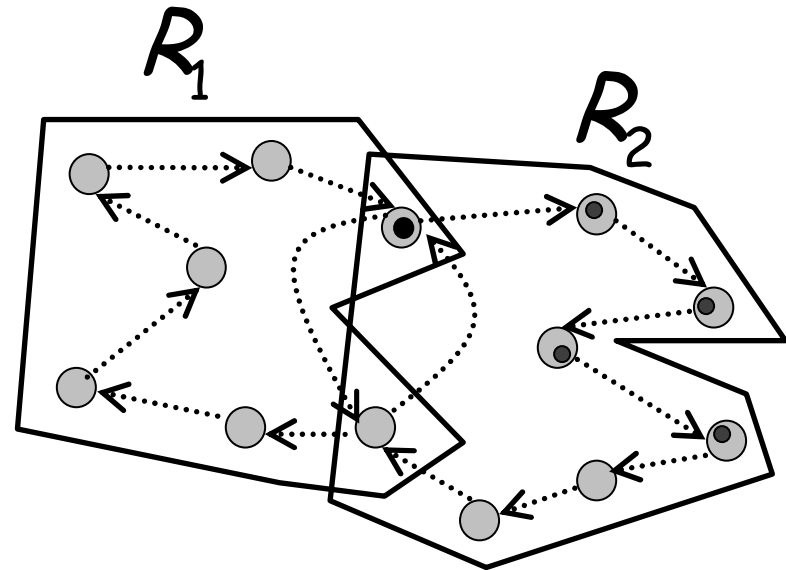
Packet moves to new region when it reaches common node

Packet Move



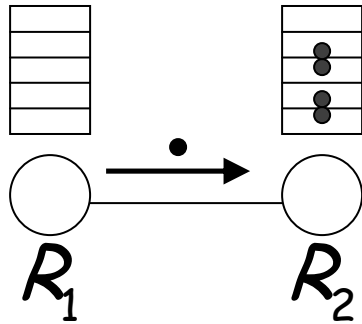
multiple packets in R_2

Simulation of move

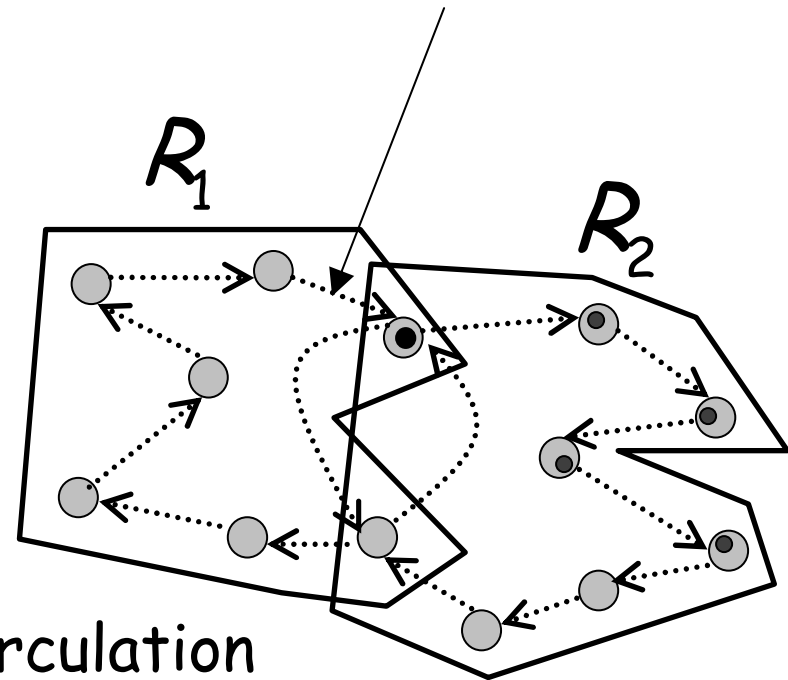


Insert packet when there is an empty spot in R_2

Packet Move



Oscillate here until empty slot is found

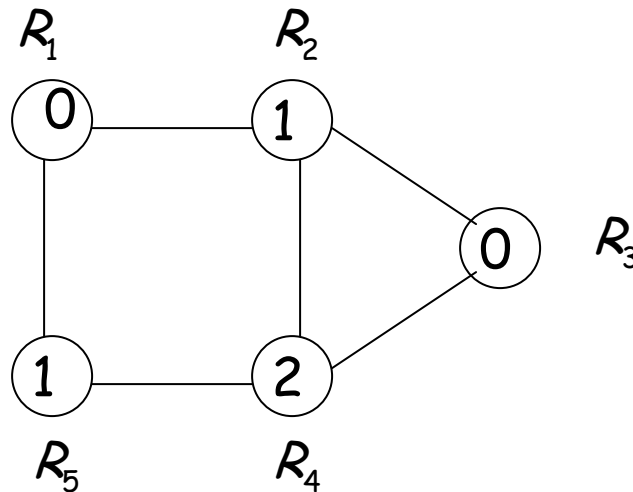


Slow Circulation

Sending
Mode

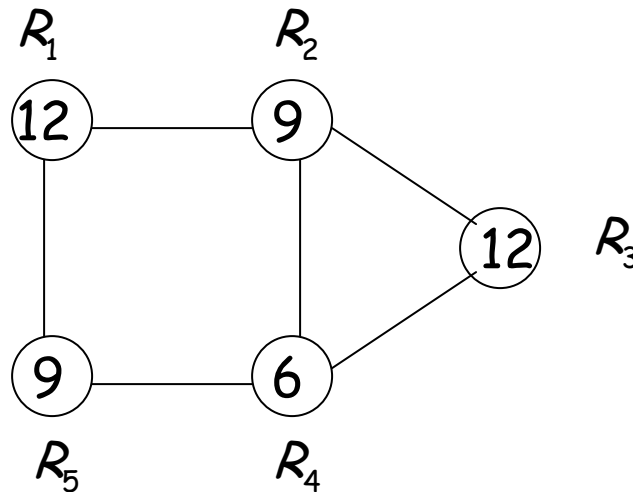
Fast Circulation
Receiving
Mode

Color the Region Graph

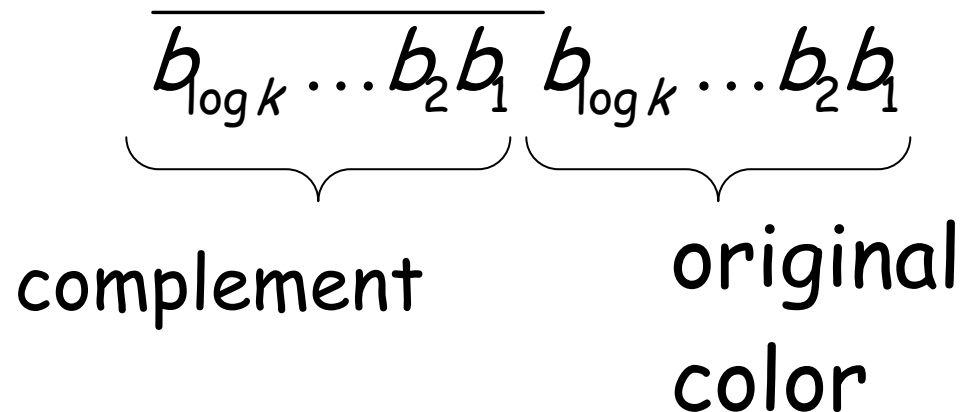


Let k be the maximum color

Change Colors



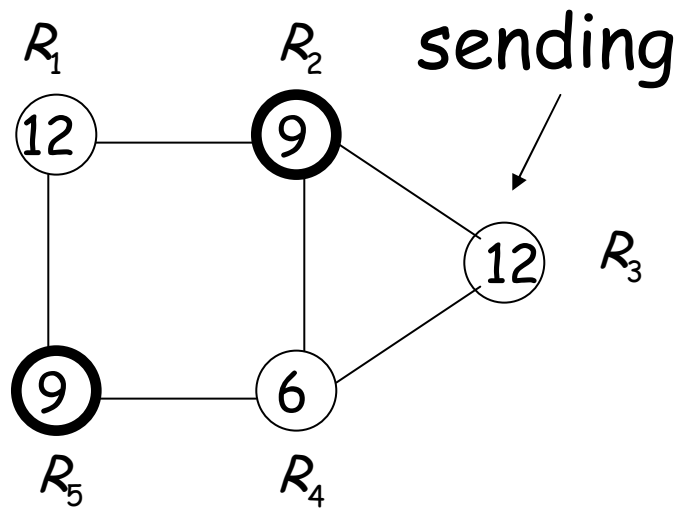
Replace each color with $2\log k$ bits



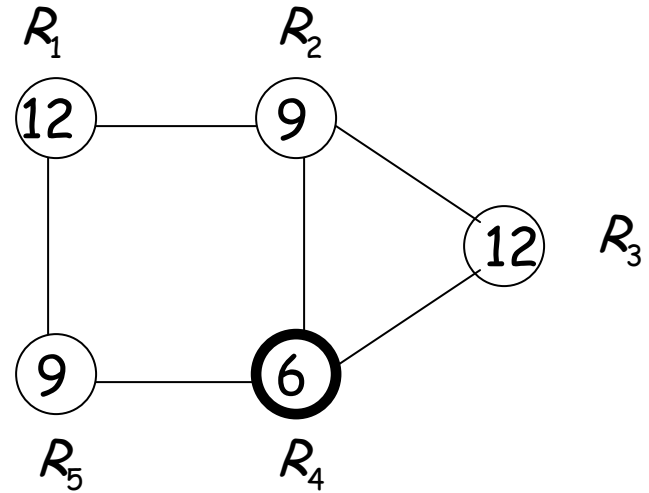
Each phase consists of $2\log k$ rounds

At round i a node is in sending mode if its i color bit is equal to 1

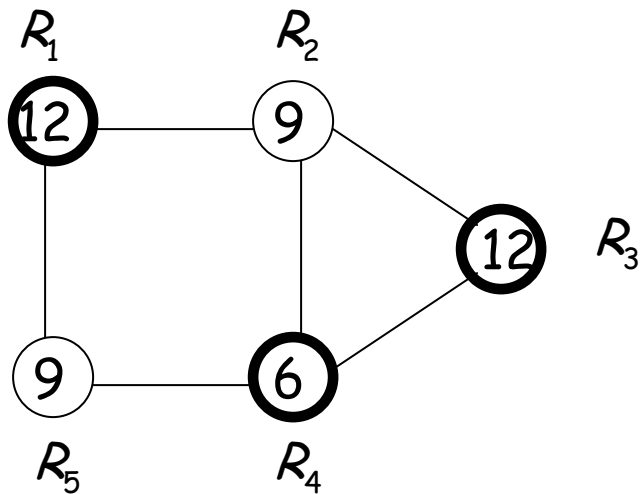
Round 0



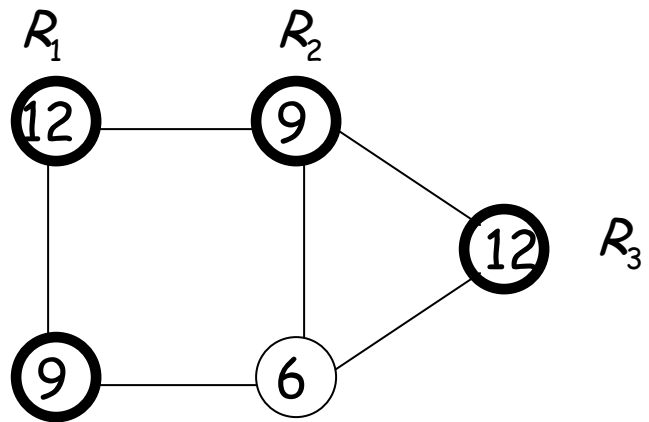
Round 1



Round 2



Round 3





There is a round that

R_i is receiving

R_j is sending

And vice-versa

Presentation Outline

- Algorithm



- Analysis

- Conclusion

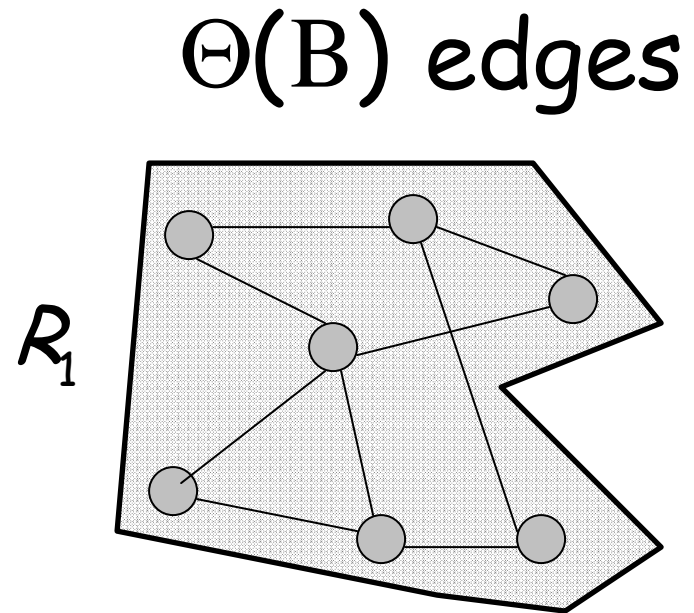
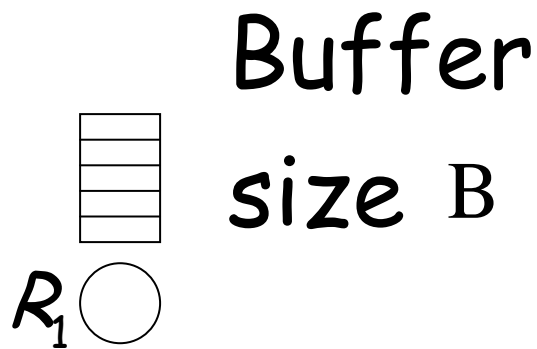
We give a buffered algorithm with

Buffer size: $B = 6 \log(n' + N)$

Routing time: $O\left(\frac{C'}{B} + D'\right)$ "Time Steps"

n', C', D' : #nodes, node congestion, dilation in region graph

We partition original graph
so that each region
has $\Theta(B)$ edges



It holds:

$$n' \leq n$$

$$C' \leq BC$$

Edges in region

Edge congestion

$$D' \leq D$$

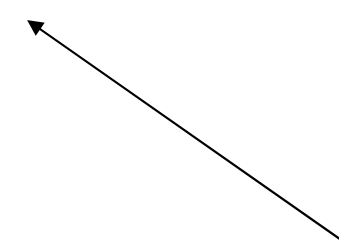
Each simulation phase has duration

$$B^2 \cdot 2\log k = O(\log^3(n + N))$$

duration of
round



number of
rounds



Total Routing Time

$$O\left(\left(\frac{C'}{B} + D'\right) \cdot \log^3(n + N)\right) = O((C + D) \cdot \log^3(n + N))$$

Number
of phases

Phase
Duration

Presentation Outline

- Algorithm

- Analysis



- Conclusion

We presented the first bufferless algorithm for arbitrary networks within logarithmic factors from optimal

Future research:

- Improve logarithmic factors
- Maintain original paths