Guaranteed Throughput and Best Effort Streams in a Single Network on Chip Model

GDR!

ZUT

2009

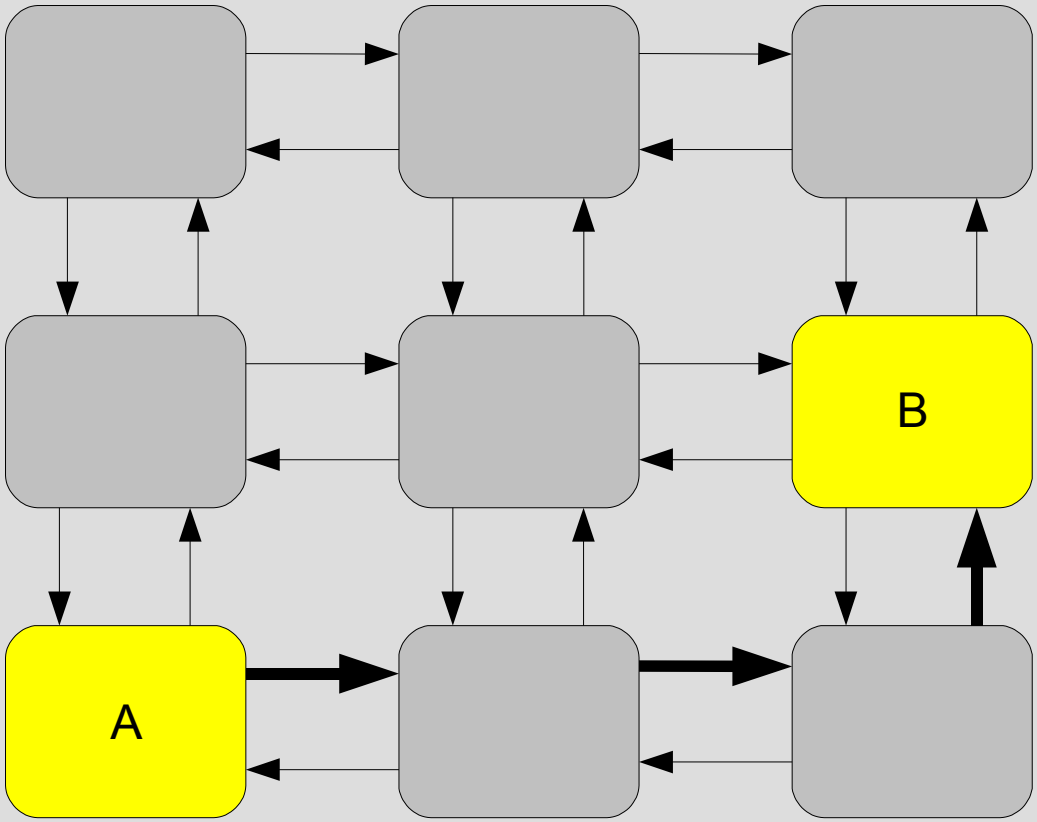
Problem description

Network on Chip
Balance the load
  Save queue length
  Reduce hotspots
Quality of Service

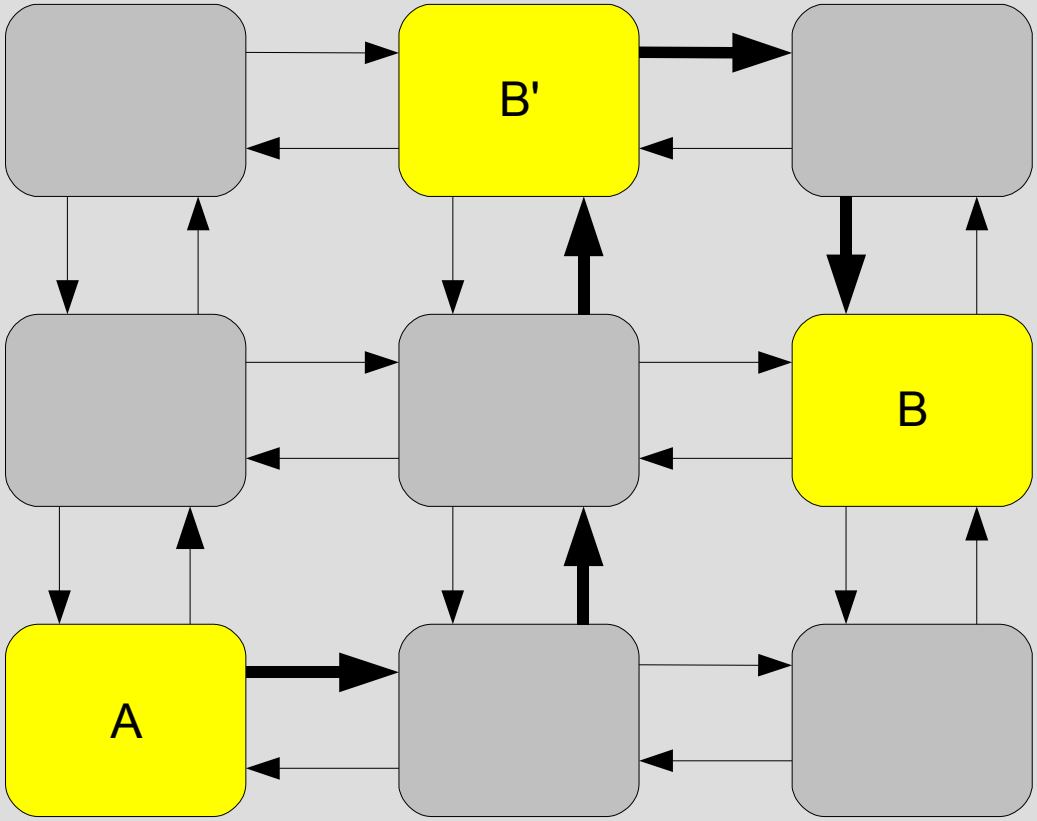
Previous work

Variants of XY algorithm
  Valiant algorithm
  Random XY/YX
    Described as near-optimal!?
Classical algorithms aren't QoS-aware

XY



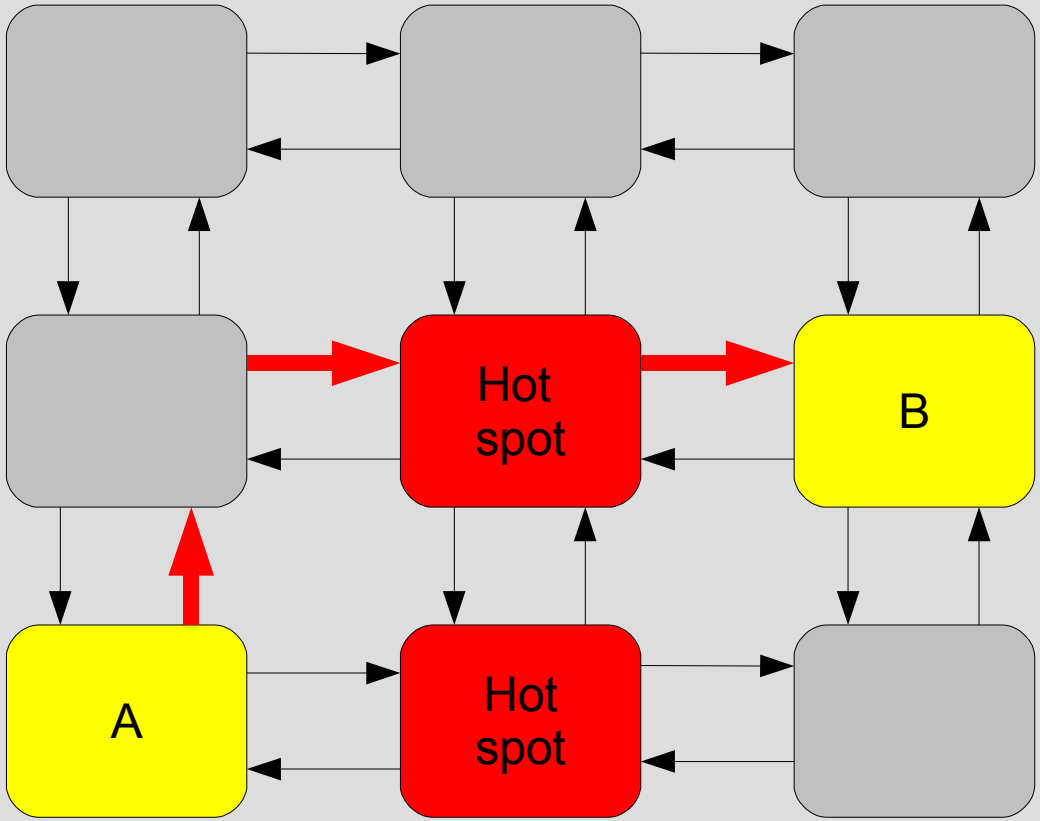
Valiant



My proposition (PARouting)

Different routes for different priorities
Best Effort packets don't need minimal routes
Local algorithm
  Could improve balancing
  Not aware of global load spread
    Could make balancing ever worse!

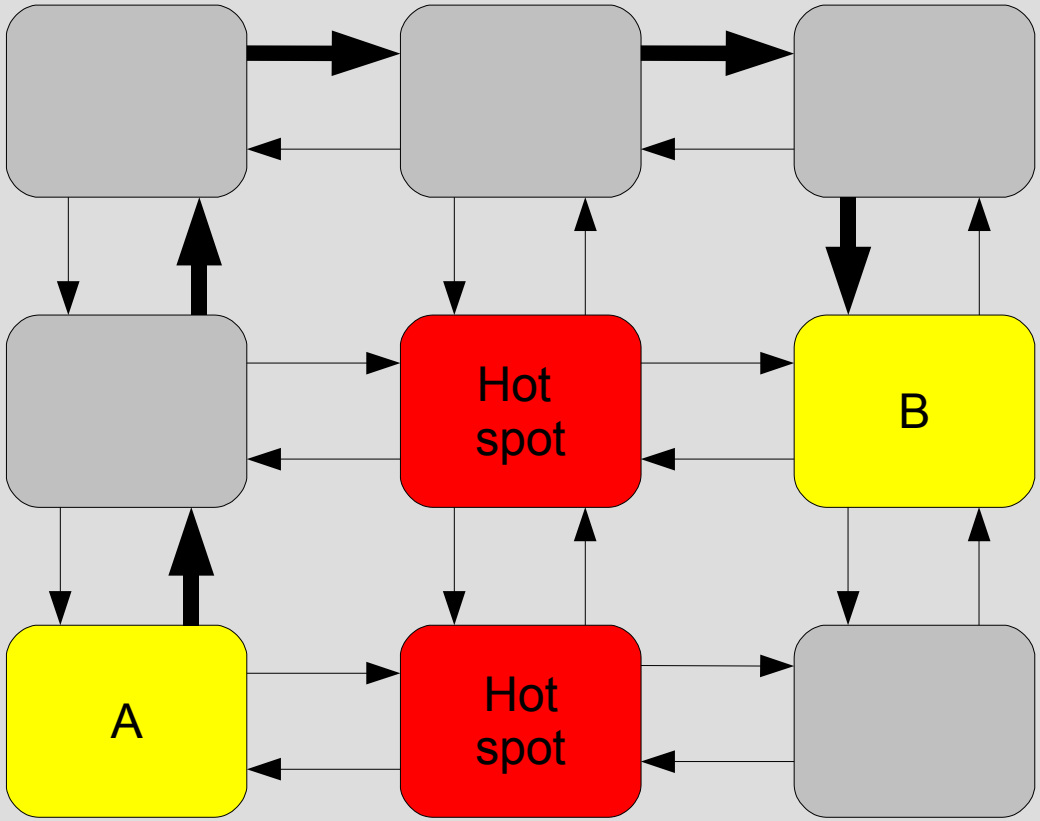
My proposition (PARouting)



Lo pri

Hi pri

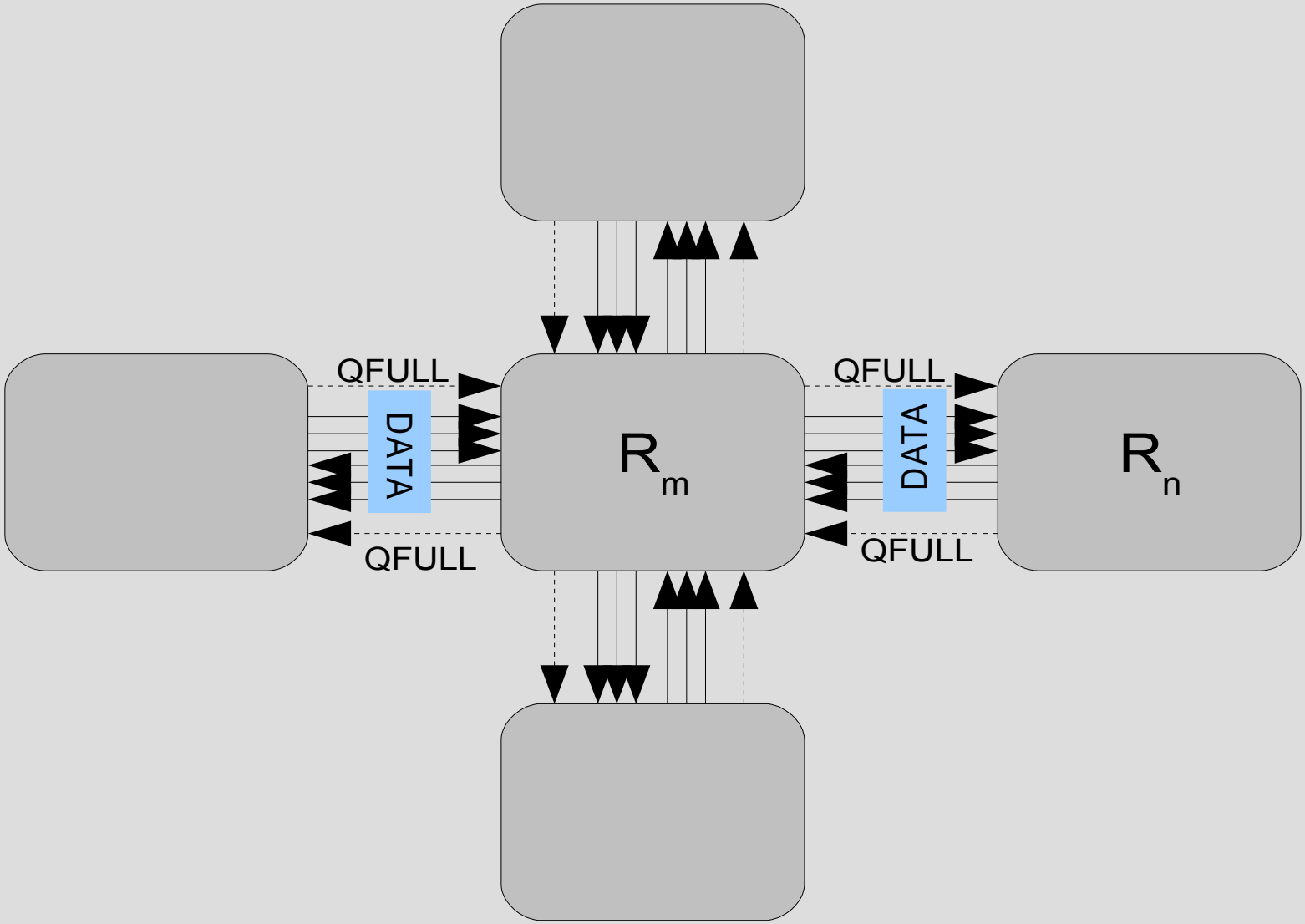
My proposition (PARouting)



Lo pri

Hi pri

PARouting – Q_{FULL} signal



PARouting - N_{bad}

Extra field in a packet – N_{bad}

2-5 bits

If N_{bad} non-zero:

"Bad" hop may be performed

If "Bad" hop performed:

N_{bad} is decreased

PARouting - N_{bad}

N_{bad} initialized depending on priority

- Depending on mesh size
- Lower priority packets get higher N_{bad}
- Highest priority packets have $N_{bad}=0$

PARouting – selecting direction

Calculate D_{XY} and D_{YX} directions for XY/YX

If any of routers at D_{XY}, D_{YX} don't emit QFULL

  Send the packet there

Else

  If $N_{bad} > 0$

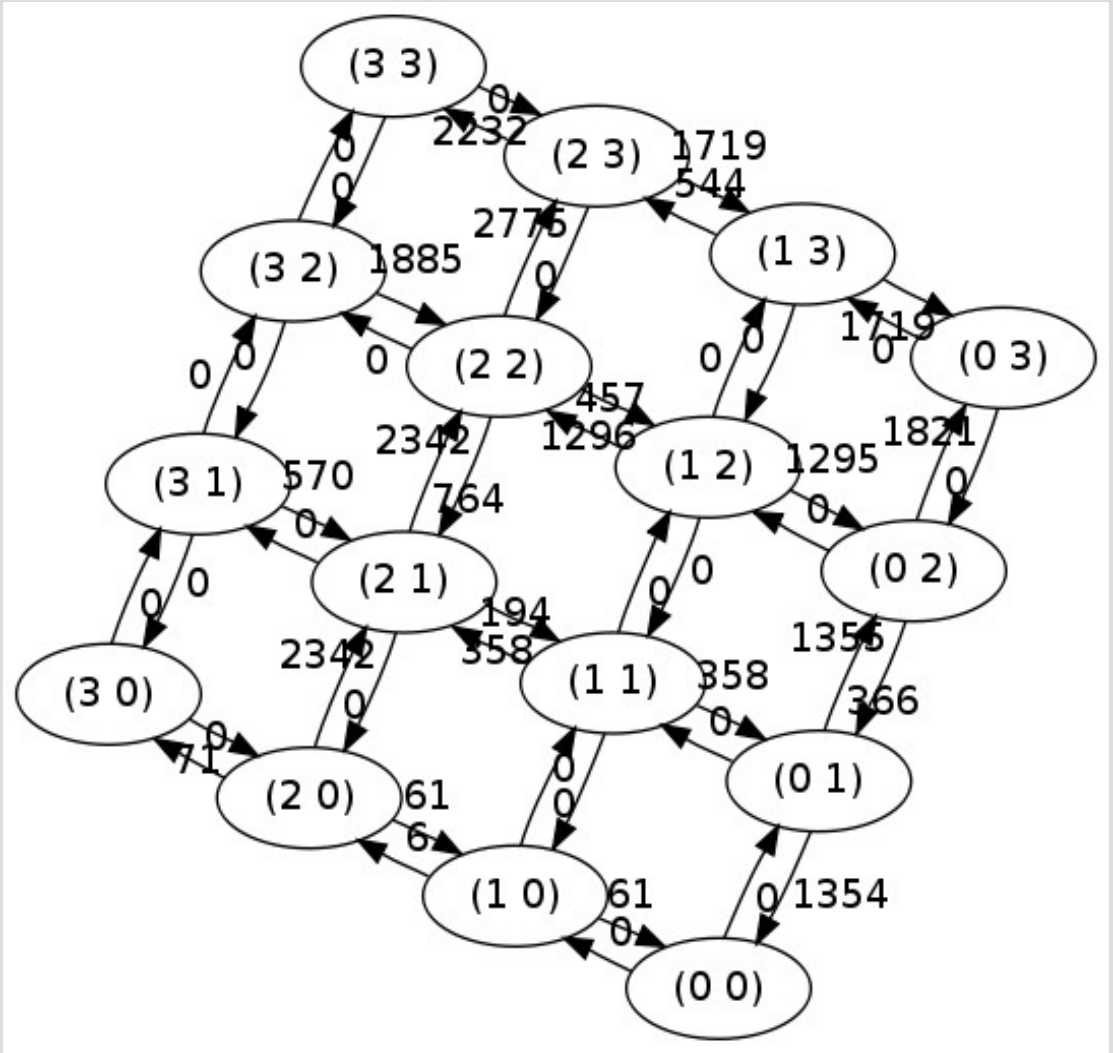
    Send packet to a non-minimal direction

  Else ($N_{bad} = 0$)

    Send packet to D_{XY}

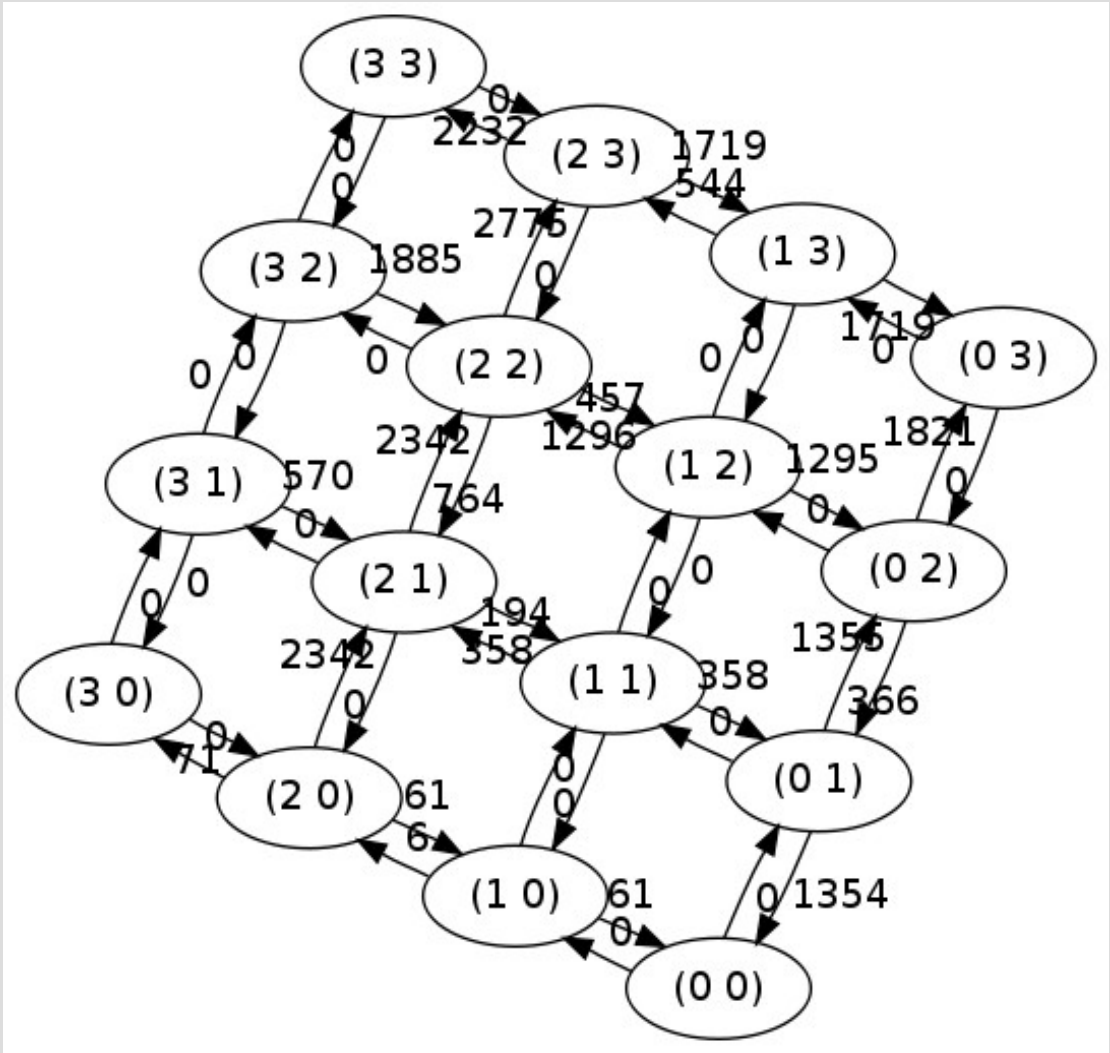
Simulation

Number of flits passed
Queue lengths
Hops
Average trip time



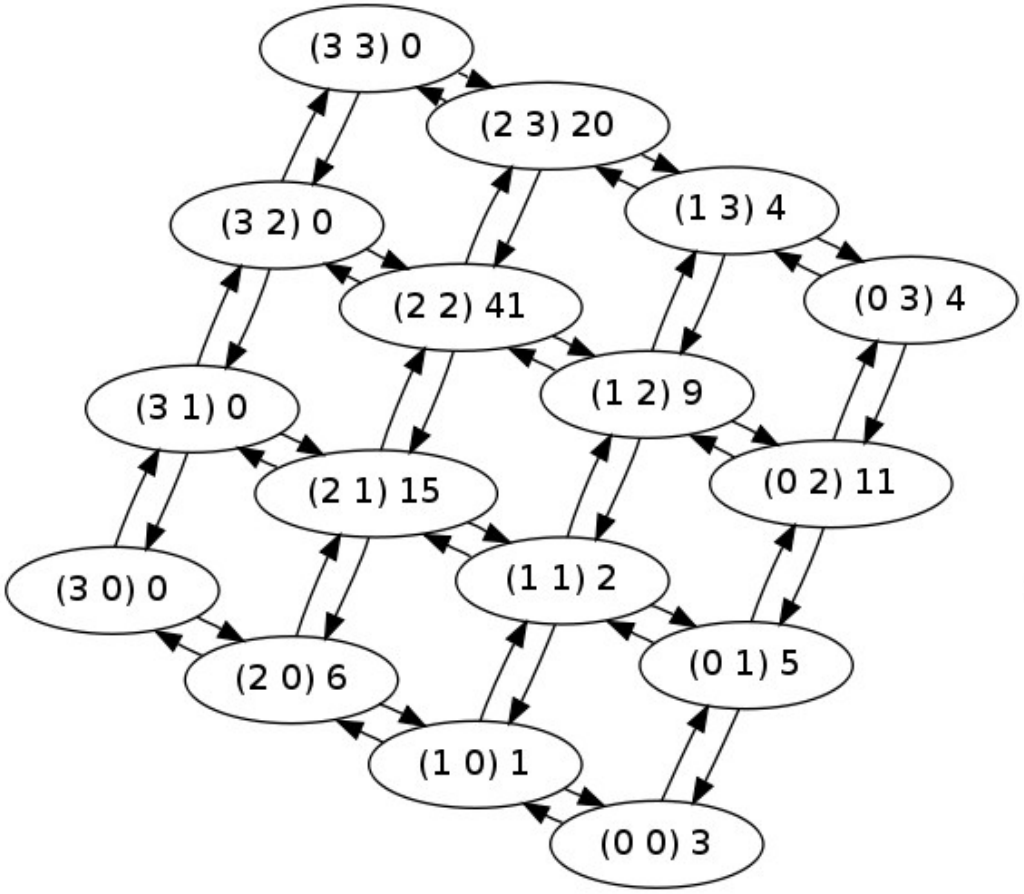
Simulation

Python
GraphViz
Gnuplot

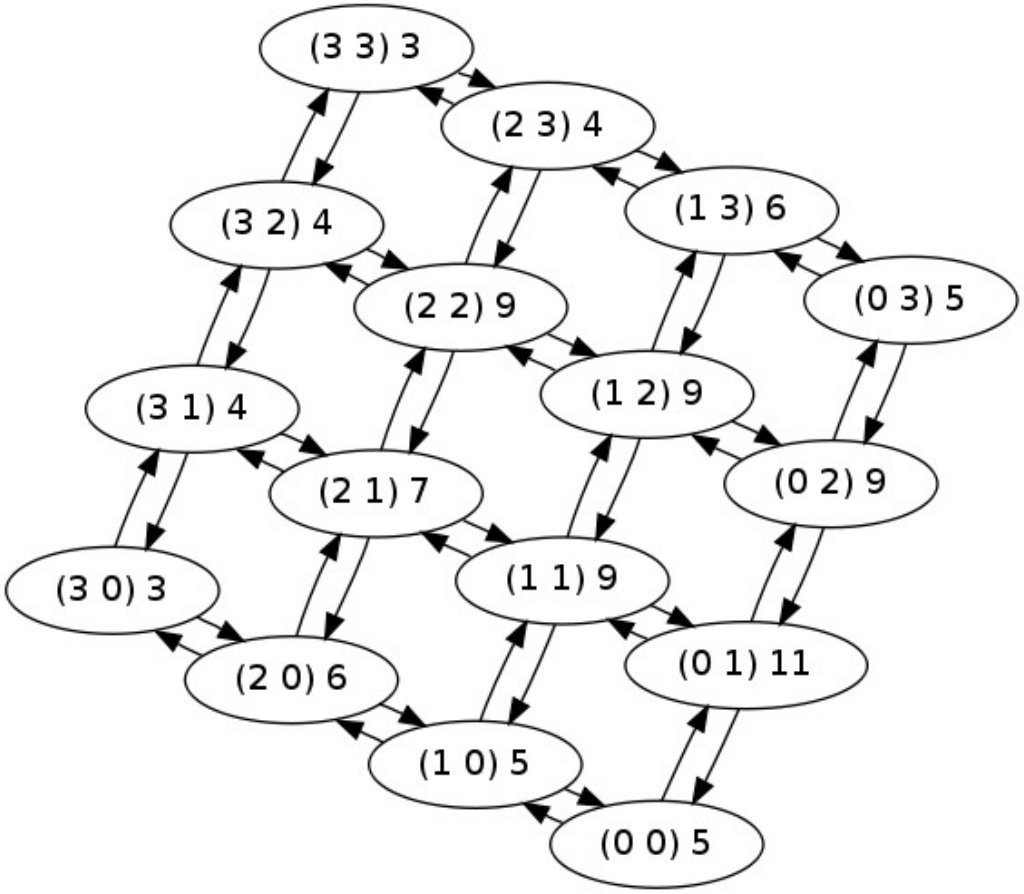


Results: reducing hotspots

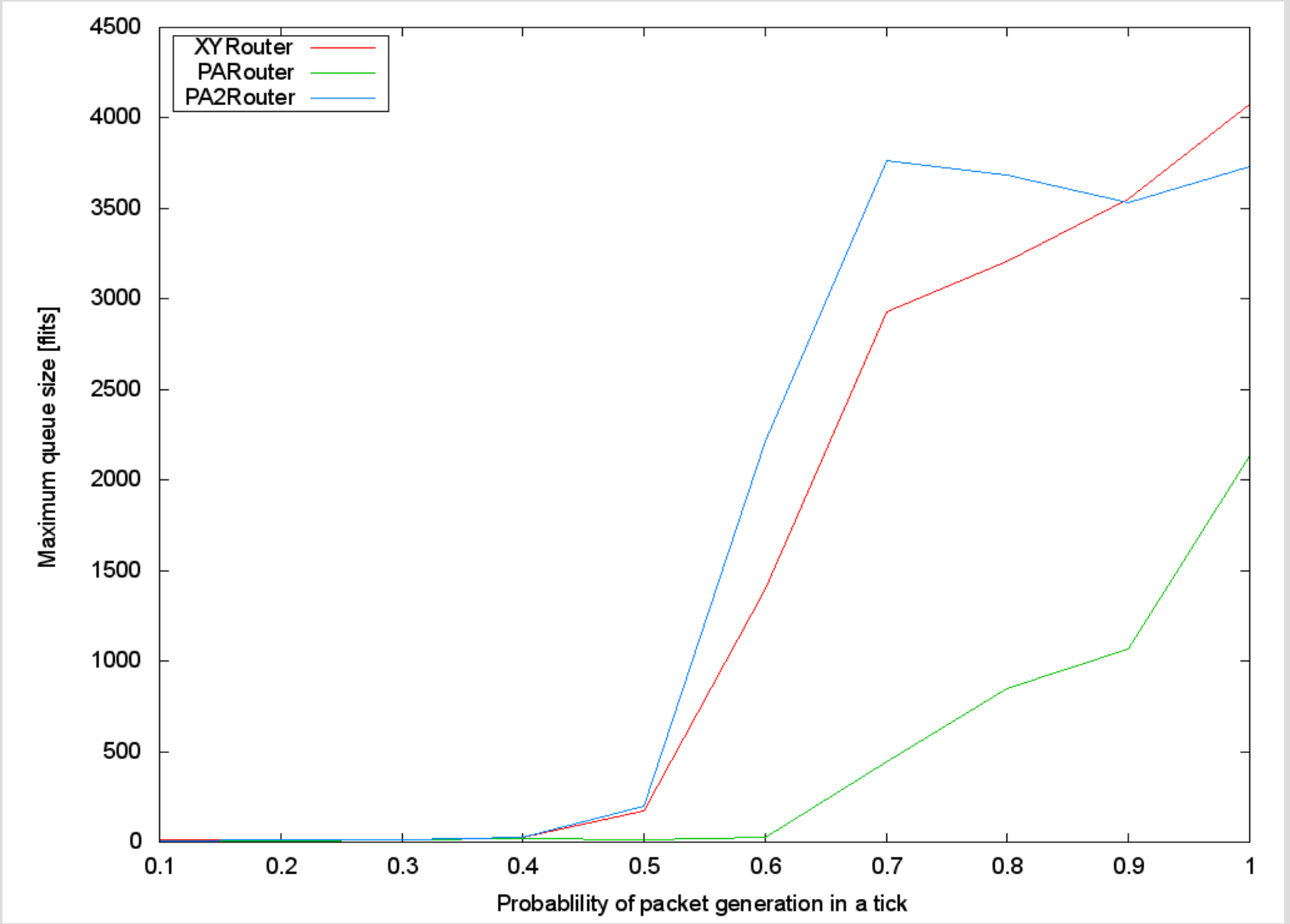
XY router



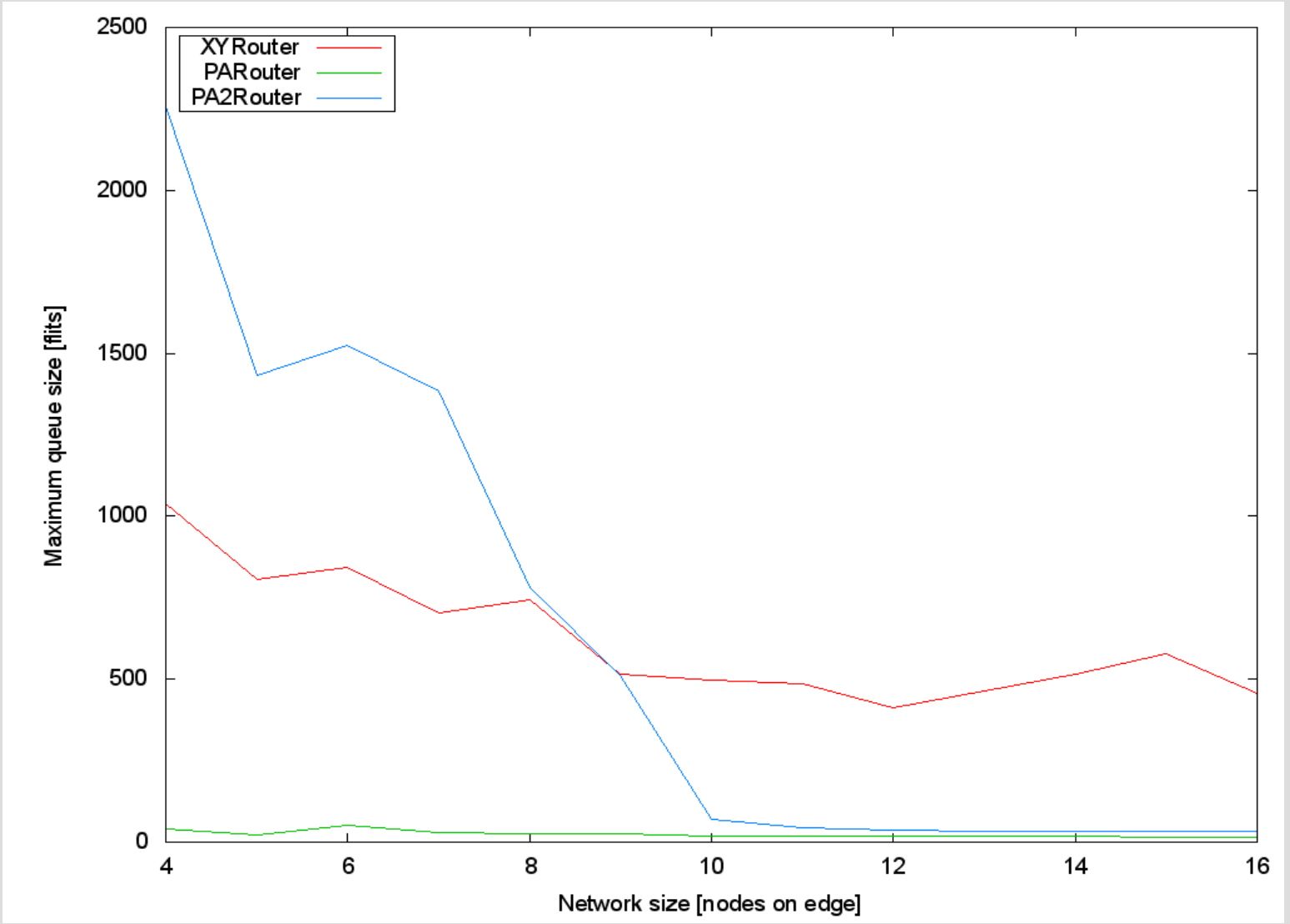
PA router



Results: queue sizes

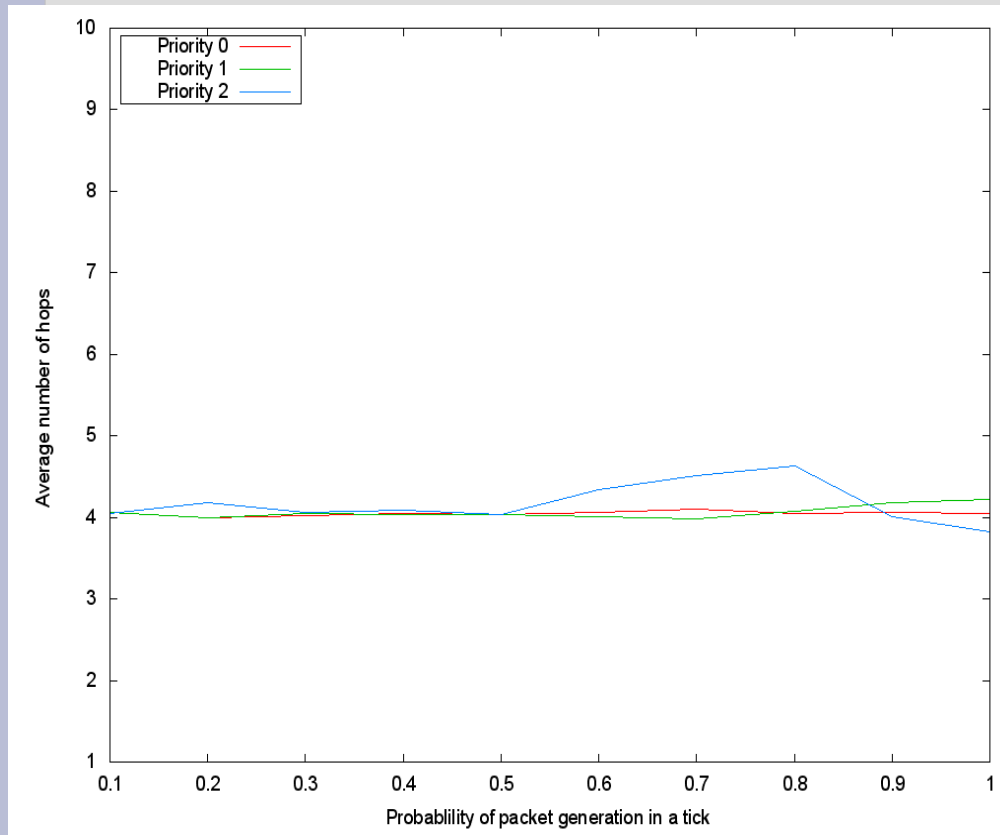


Results: queue sizes

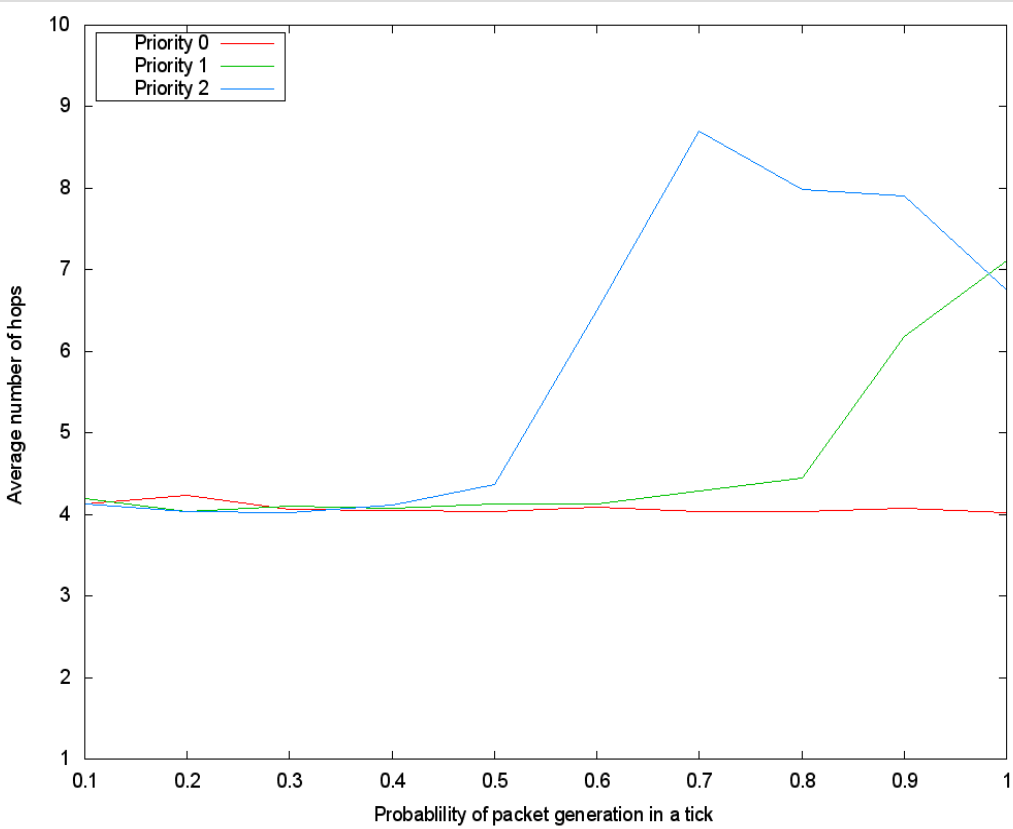


Results: hops

XY router



PA router



Summary

Better load distribution between nodes
Smaller queue sizes
At the cost of latency
  But not of high-priority packets