Adaptive Stochastic Routing in Fault-tolerant On-chip Networks

Wei Song[1], Doug Edwards[1],
Jose Luis Nunez-Yanez[2], and Sohini Dasgupta[1]

1. School of Computer Science, University of Manchester
2. Dep. of Electrical and Electronic Engineering, Bristol University

MANCHESTER 1824

University of BRISTOL

EPSRC
Engineering and Physical Sciences Research Council

Research Background
"A Future FPGA proposal"

- ***NoC*** the new on-chip communication paradigm
- ***FPGA*** the fast prototype platform
- ***Fault-tolerance***
  – Cope with the errors of the nanoscale transistors
- ***NoRC*** = FPGA + NoC + Fault-tolerance
  – Tens to hundreds of programmable tiles connected by hardwired on-chip networks
  – Broadband communication fabric
  – Support multiple multimedia applications
  – Online reconfigurable hardware accelerators
  – Fault-tolerance

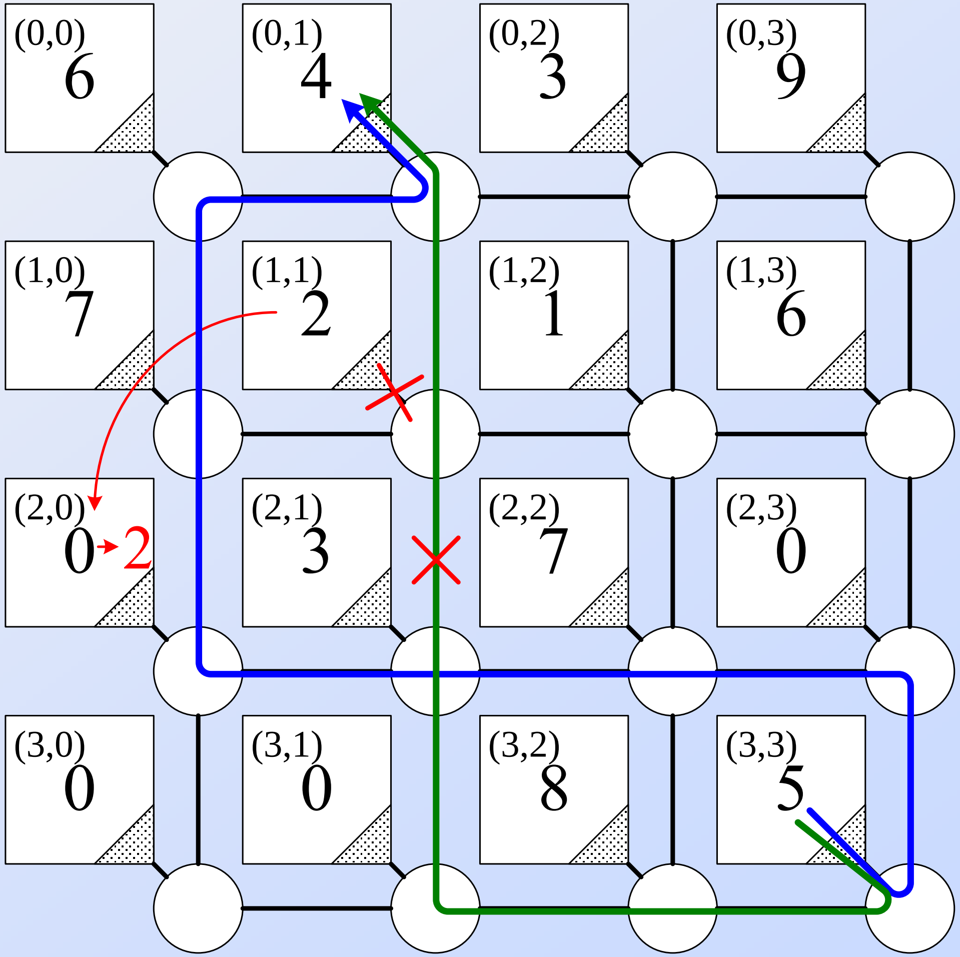
The NoRC Platform

- **Online reconfigurable processor node**
- **Connection oriented flow control**
  - Build a path before the data transmission
  - Delay guaranteed
- **Communicate by the Function Identifier (FID)**
  - Directly request a function instead of using the address
- **Stochastic routing algorithm (COSR)**
  - Directly search for the function on an unknown node
- **Fault-tolerance**
  - Random paths selected by the stochastic routing algorithm
  - Reconfigure the network when some nodes fail

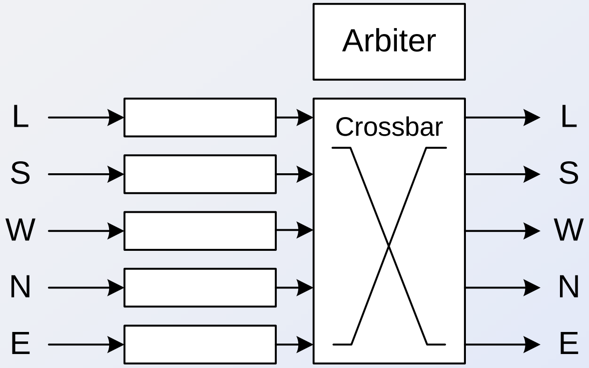
A NoRC Example

- The Connection Oriented Stochastic Routing (COSR) Algorithm

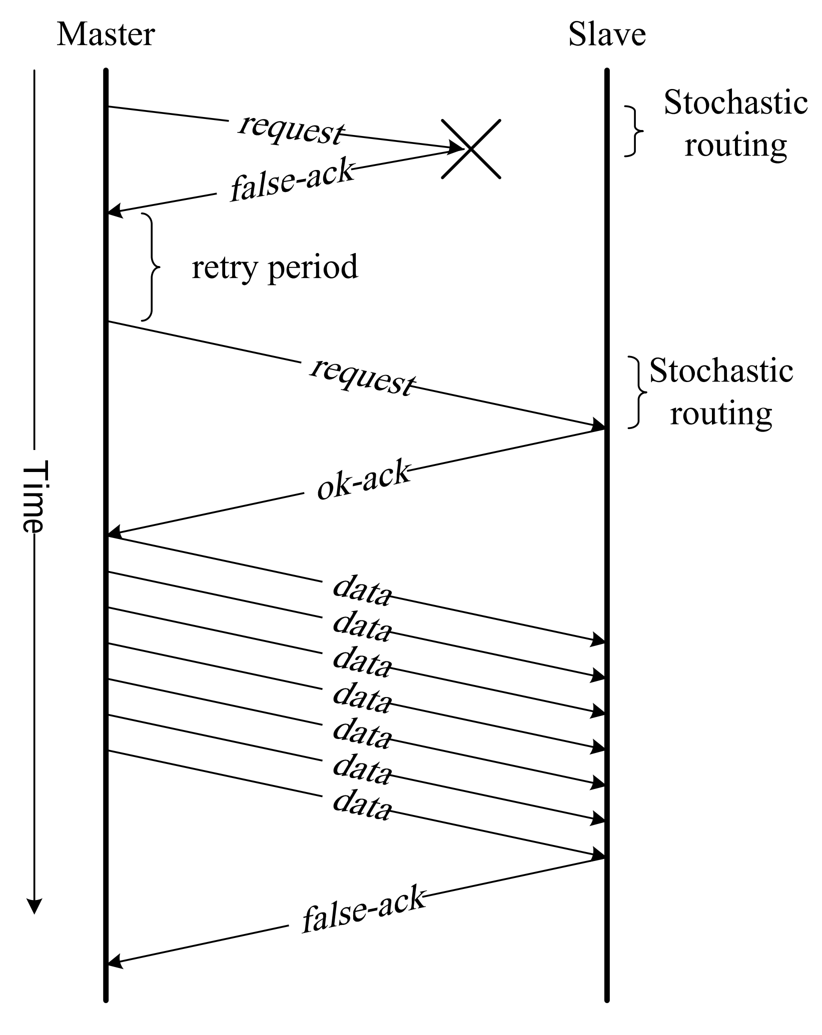
- The node reconfiguration



Router and Flit Types



- Simple router
- 4 types of flits
  - *request*
  - *false-ack*
  - *ok-ack*
  - *data*



Analyze the COSR

P, ports point to shorst paths

$\overline{\mathrm{P}}$, ports point to non-shortest paths

L, the length of a path

$\hat{p}$, the first selected port

$P = P\{L \le 16\}$

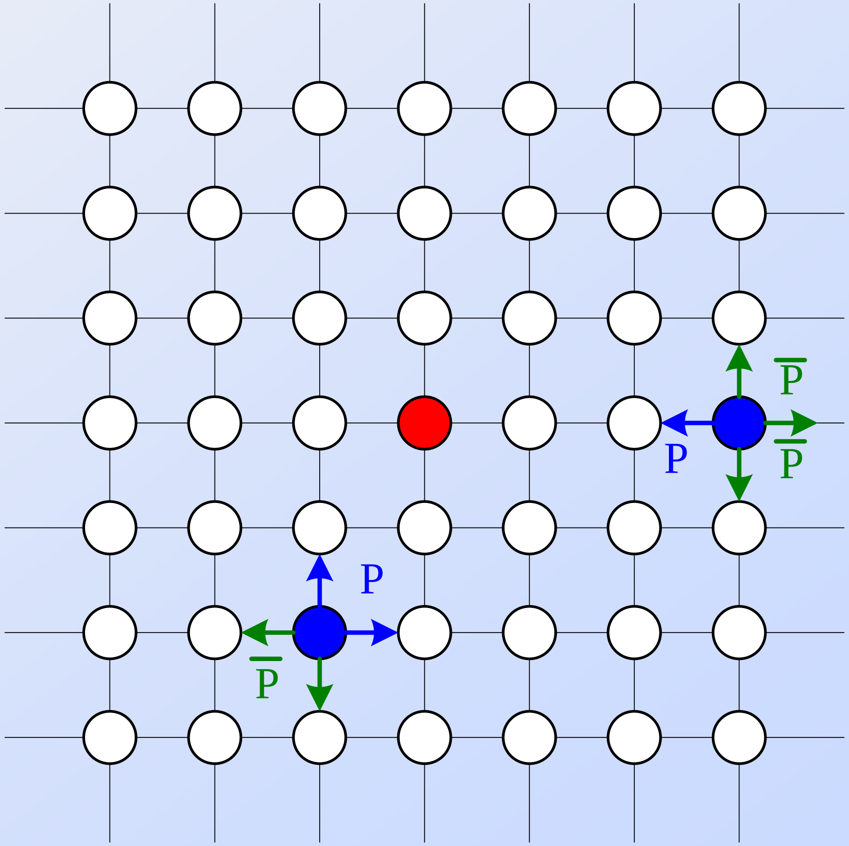
The probability to find the target in 16 hops

$P_{\mathrm{P}} = P\{L \le 16 \mid \hat{p} \in \mathrm{P}\}$

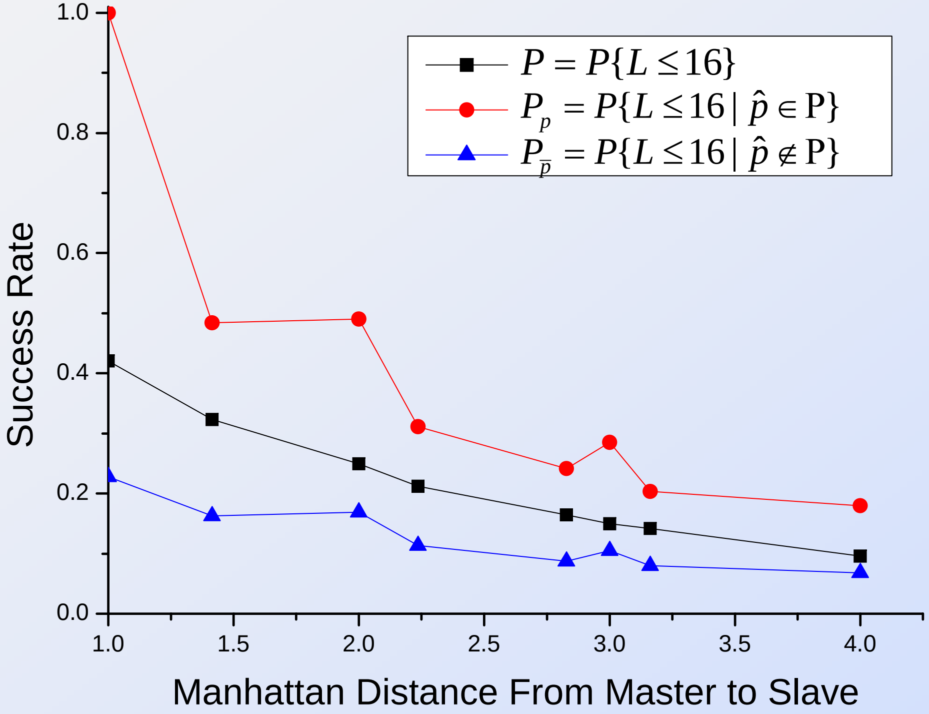
The probability when using set P

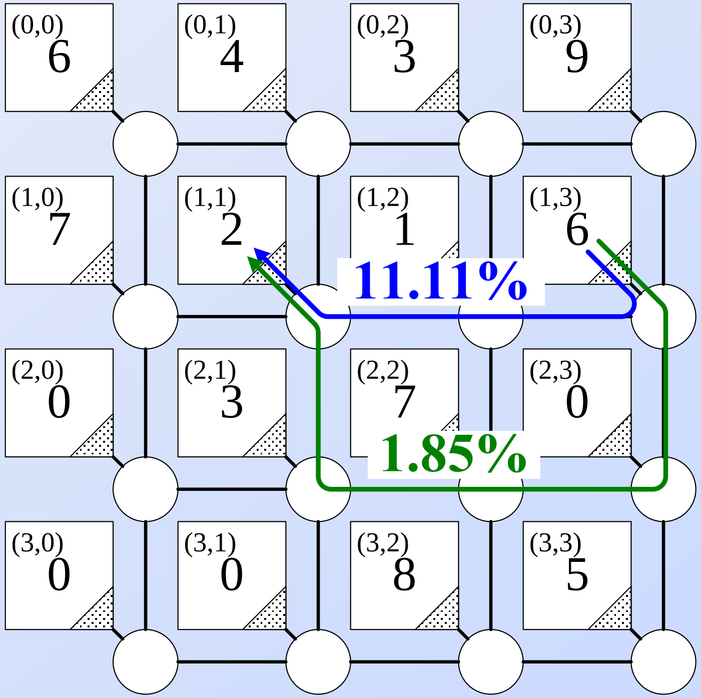
$P_{\overline{\mathrm{P}}} = P\{L \le 16 \mid \hat{p} \in \overline{\mathrm{P}}\}$

The probability when using set $\overline{\mathrm{P}}$



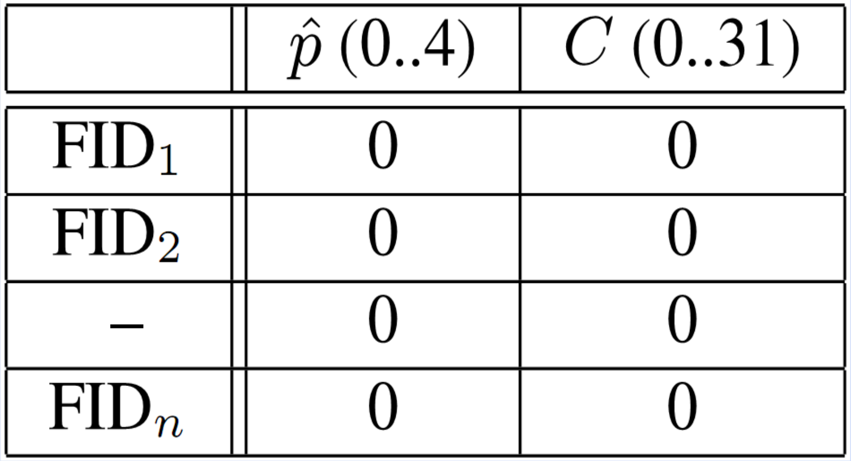
The Success Rate of COSR





Adaptive Stochastic Routing (ASR) Algorithm

Try to remember the last successful output port.



	$\hat{p}$ (0..4)	C (0..31)
FID_1	0	0
FID_2	0	0
–	0	0
FID_n	0	0

The update algorithm

- *request*

  Select a random number *v*

  Choose $\hat{p}$ in RT if *v*<*C*

  Choose a random port if $v \geq C$

- *false-ack*

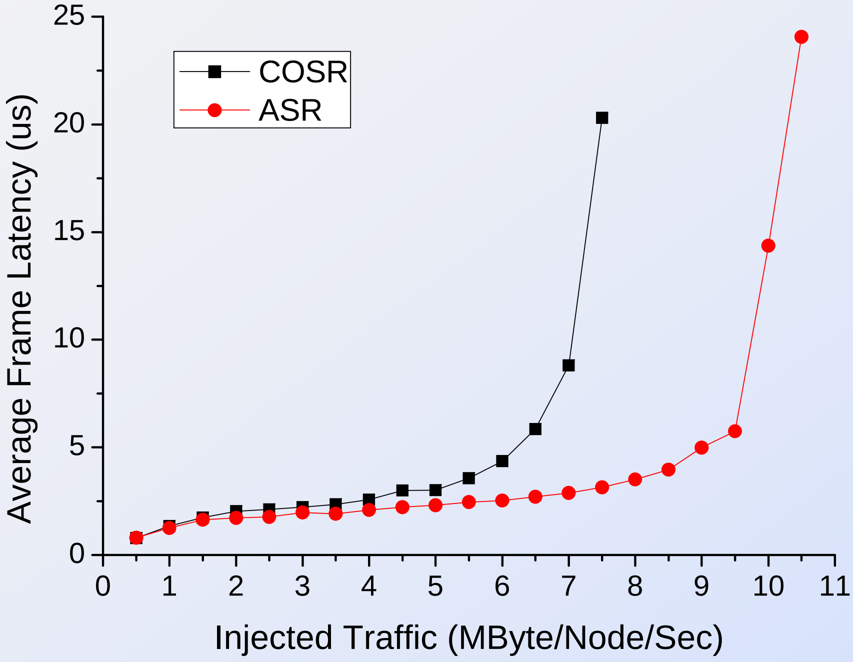
  C=C-1 if come from $\hat{p}$

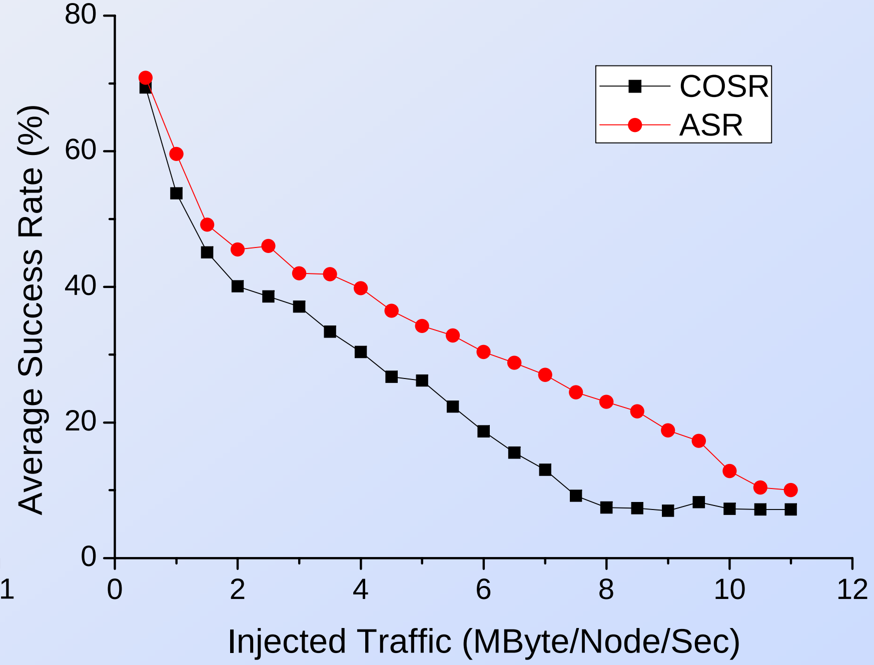
- *ok-ack*

  C=C+2 if come from $\hat{p}$

  Update $\hat{p}$ if C=0

Performance on an Error-free NoC

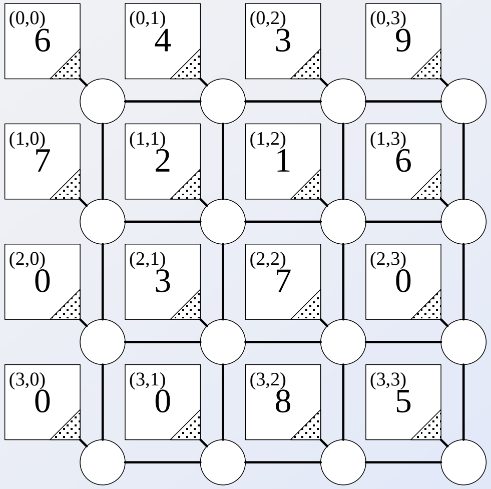




40% enlargement of the maximal accepted traffic.

15.59% higher success rate at the maximal 8MByte/Node/Sec injected traffic.

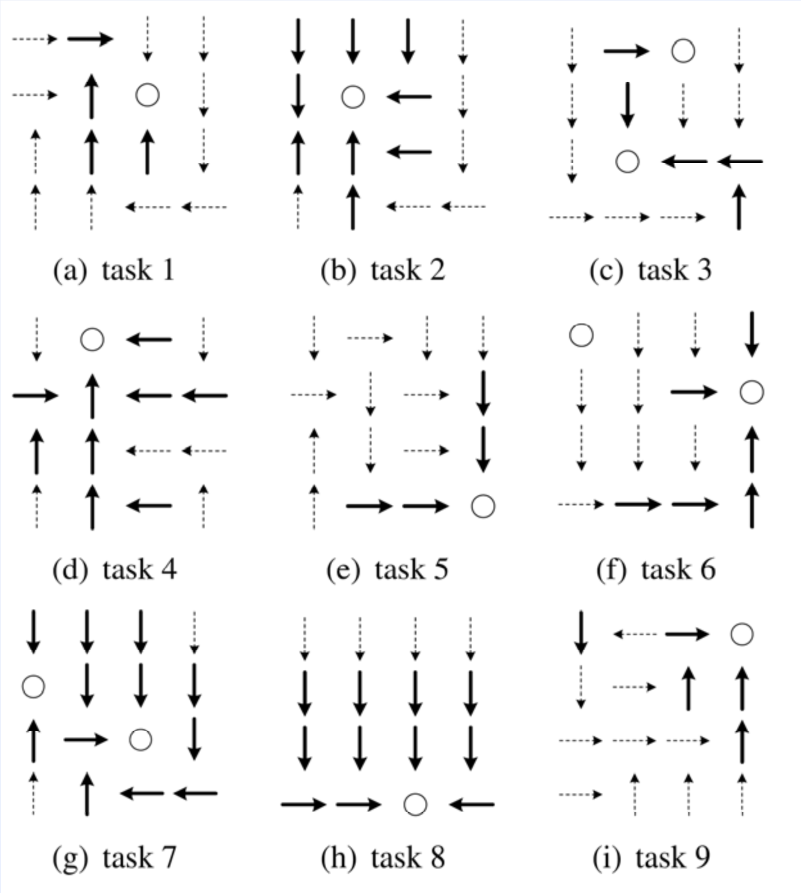
Result of the Learning Procedure



Injected traffic: 1MByte/Node/Sec

➞ Strong estimation, C >= 16

- -▸ Weak estimation, C < 16



(a) task 1 (b) task 2 (c) task 3

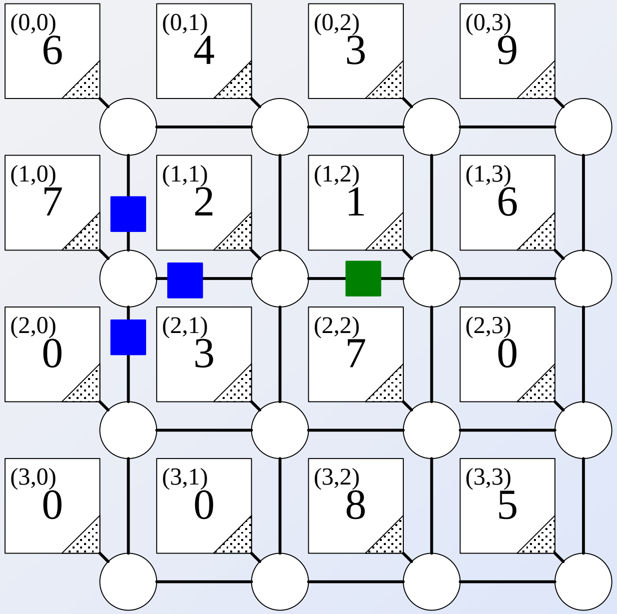
(d) task 4 (e) task 5 (f) task 6

(g) task 7 (h) task 8 (i) task 9

78.8% ports estimations are correct.

93.4% strong estimations are correct. (Strong estimations affect the routing procedure more significantly than weak ones.)

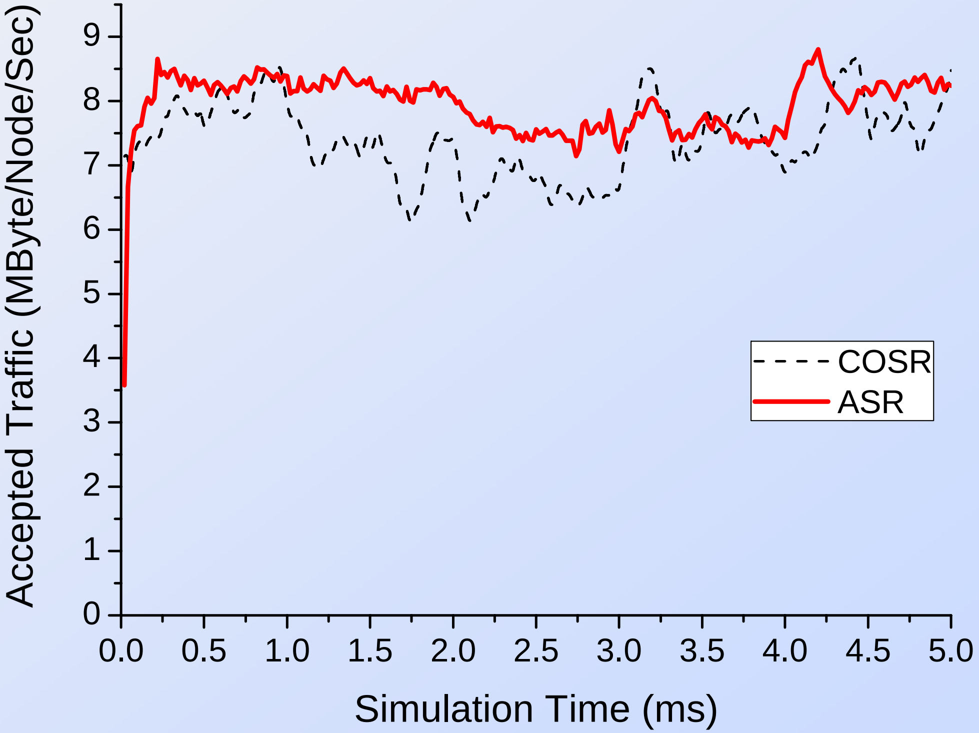
Performance on a Faulty NoC
Intermittent Errors



🟩 From 1ms to 3ms

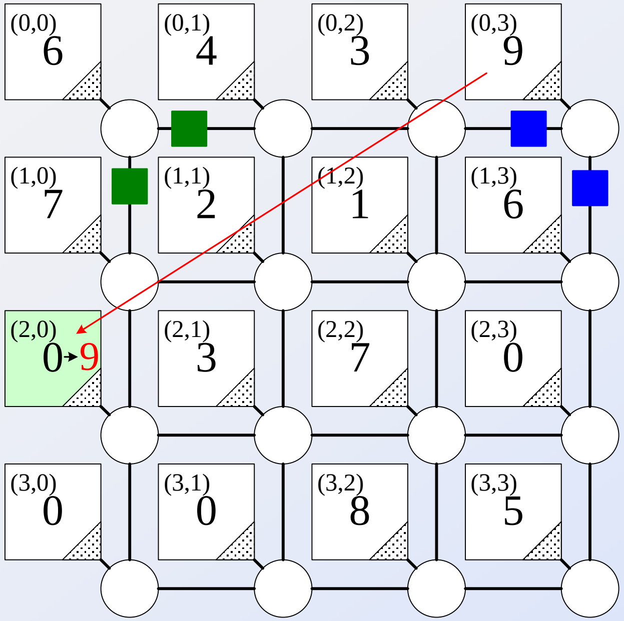
🟦 From 2ms to 4ms

Injected traffic: 8MByte/Node/Sec



Intermittent errors does not affect the traffic significantly.

Performance on a Faulty NoC Permanent Errors

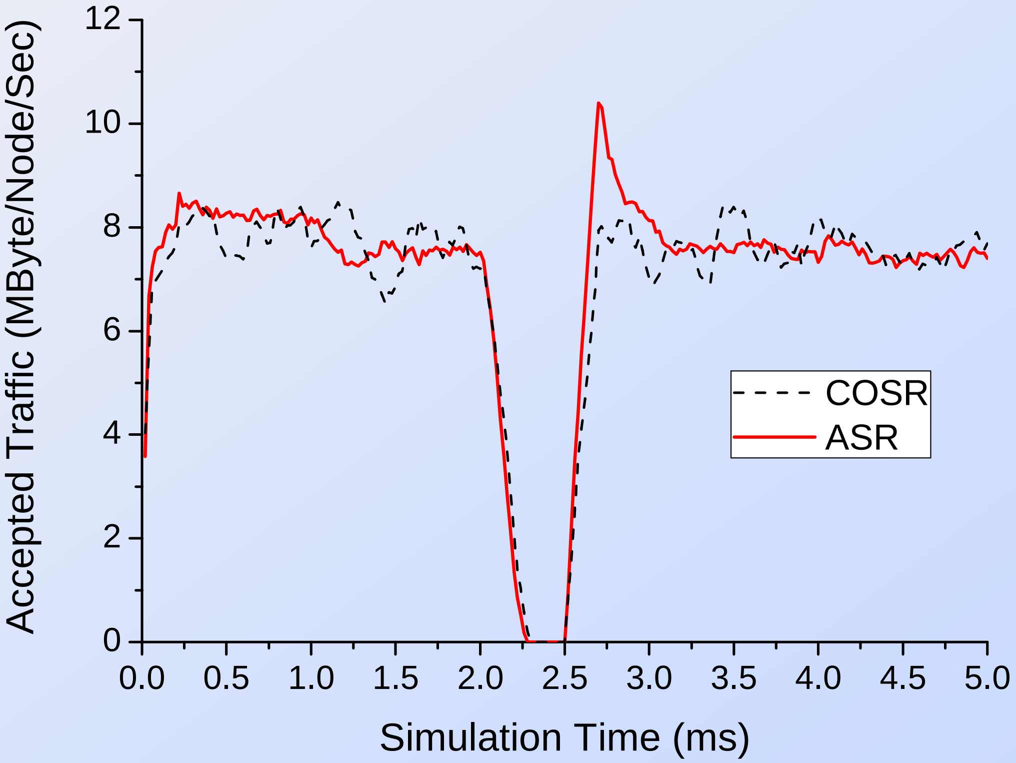


🟩 After 1ms

🟦 After 2ms

Node (2,0) is reconfigured to FID 9 after 2.5ms

Injected traffic: 8MByte/Node/Sec



If a node fails without an alternative node, the network can recover by reconfiguring it to another node.

Conclusions and Future Work

- Conclusions
  - NoRC is a fault-tolerant and dynamically reconfigurable NoC platform.
  - The COSR algorithm can search for any existing functions (slave candidates) but it is inefficient.
  - By learning from the results of COSR, ASR could improve the sub-optimized performance.
  - Both Algorithms are fault-tolerant to permanent/intermittent errors.

- Future Work
  - Implement them into an asynchronous router.
  - Design an error detection scheme in routers.
  - Combine the deterministic routing algorithms and the stochastic routing algorithms to further improve the network performance.

Thanks for your listening!
And thank Simon for his presentation!

Any further questions,

songw@cs.man.ac.uk