An Efficient Implementation of Distributed Routing Algorithms in NoCs

Authors: J. Flich, S. Rodrigo, and J. Duato

Parallel Architectures Group

Technical University of Valencia, Spain

Agenda

Introduction

System environment

Description

Evaluation

[Further evaluations]

Conclusions

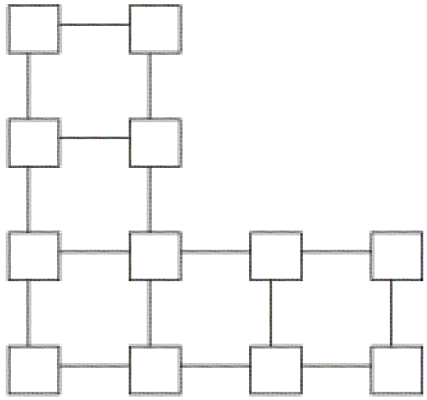
Introduction

MINISTERIO DE EDUCACIÓN Y CIENCIA
INFRAESTRUCTURA CIENTÍFICA COFINANCIADA POR EL FONDO EUROPEO DE DESARROLLO REGIONAL
Consolider

- Multi-core arquitectures are becoming mainstream for designing high performance processors
- Performance on single-core solutions is limited by power
- The trend is to integrate a large number of cores inside a chip
- Need for a high-performance on-chip interconnect (NoC) to communicate eficiently between all chip devices

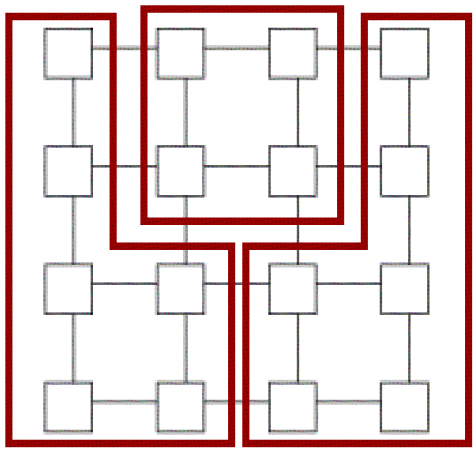
Introduction (2)

- Area, power and delay are the main constraints when designing a NoC
- Some problems arise:
  - High integration scale -> communication reliability issues
  - Fabrication faults
- Those problems lead to an irregular topology still functional

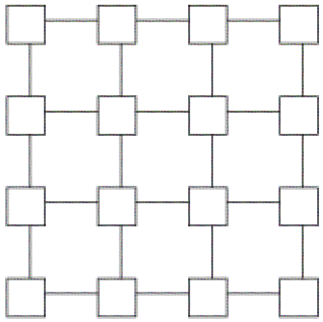


Introduction (3)

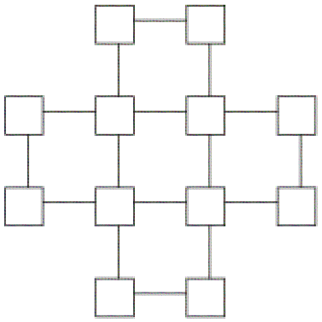
- Virtualization of the chip is also possible thanks to the increasing number of cores
  - Efficient use of resources
  - Distributing system resources among different tasks
- So, the original 2D mesh is partitioned into different irregular topologies.



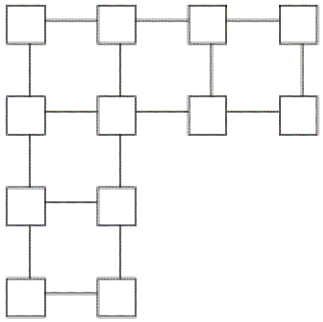
Introduction (4)



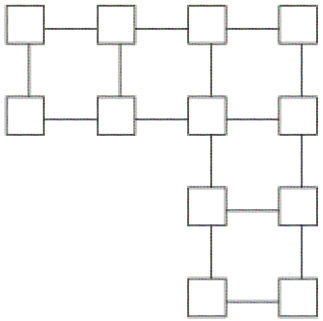
(a) 2D mesh



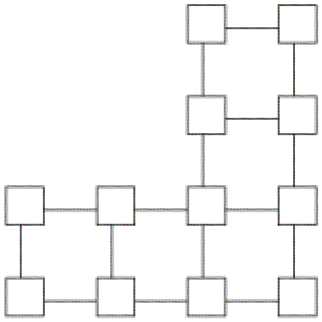
(b)"+" topology



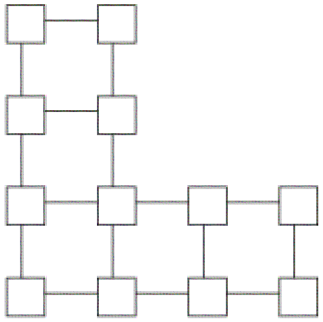
(c) p topology



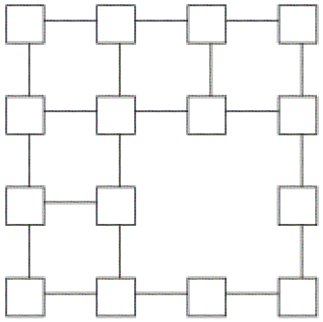
(d) q topology



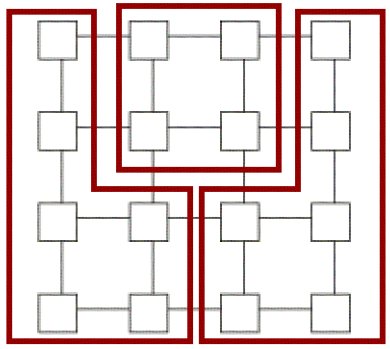
(e) d topology



(f) b topology



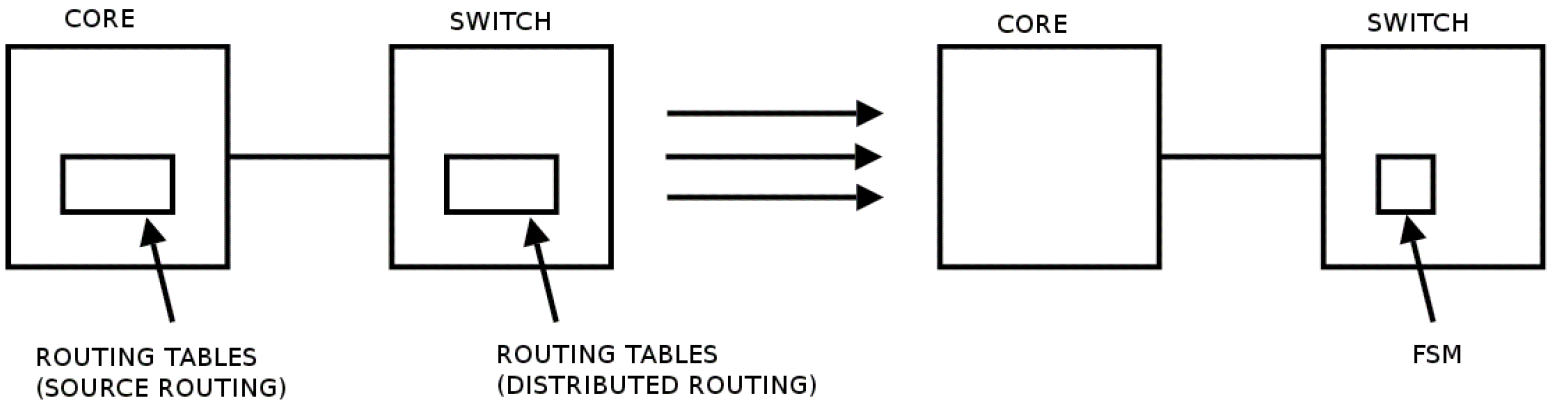
(g) non–minimal topology



(h) different regions

Introduction (5)

- To deal with irregular topologies, switches based on forwarding tables are preferred off-chip.
- However, on-chip, area, power and delay constraints are critical as memories do not scale in those terms.
- PROPOSAL: LBDR (Logic-Based Distributed Routing) is implemented to get rid of tables with a minimum logic to allow the use of any distributed routing algorithm.



Agenda

Introduction

System environment

Description

Evaluation

[Further evaluations]

Conclusions

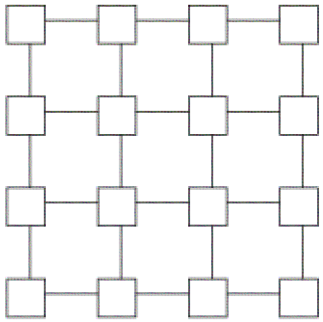
System environment

- For LBDR to be applied, some conditions must be fulfilled:
  - Messages routed with X and Y offsets, every switch must know its own coordinates
  - Every end node can communicate with other node through a minimal path
- LBDR, on the other hand:
  - There is no restriction to be applied in systems with or without virtual channel requeriments.
  - Supports both wormhole and virtual cut-through switching

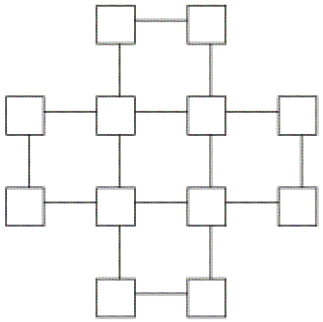
System environment (2)

- LBDR is applicable to any routing algorithm that enforces minimal paths for every source-destination pair:
  - A deterministic routing algorithm without cyclic dependencies can be represented by routing restrictions
  - A routing restriction forbids a packet to use two consecutive channels

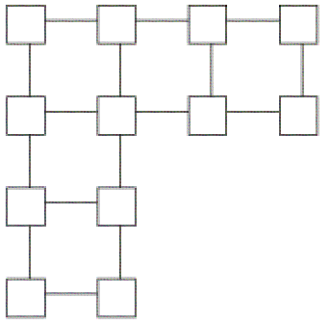
System environment (3)



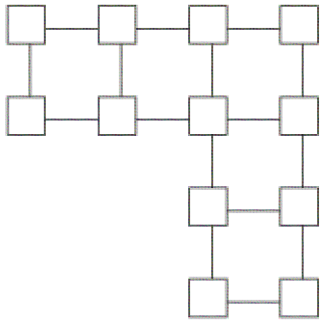
(a) 2D mesh



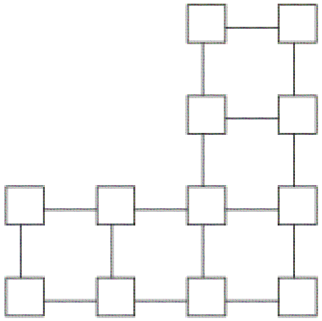
(b)"+" topology



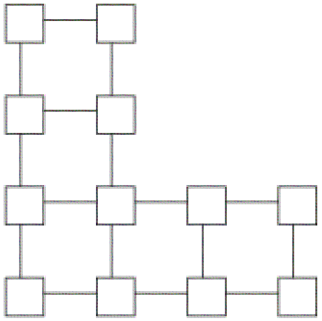
(c) p topology



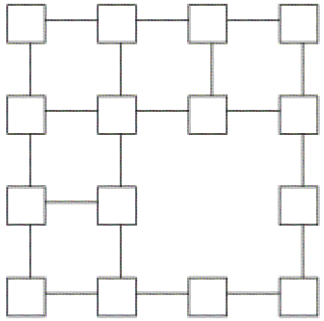
(d) q topology



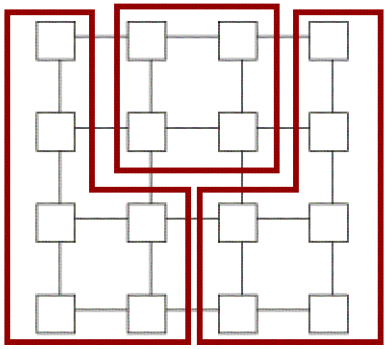
(e) d topology



(f) b topology

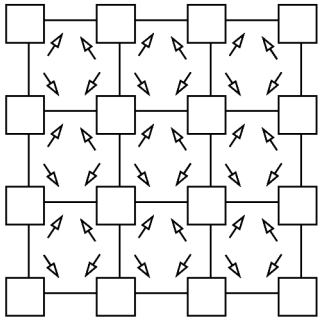


(g) non–minimal topology

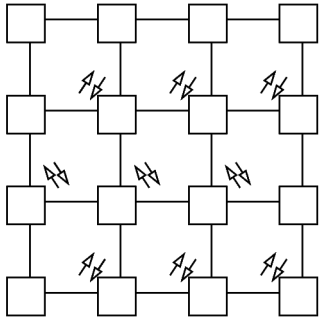


(h) different regions

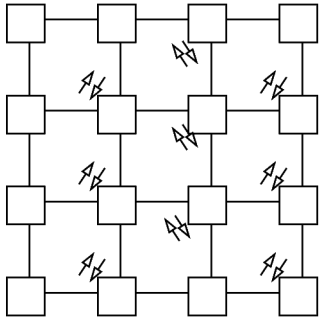
System environment (4)



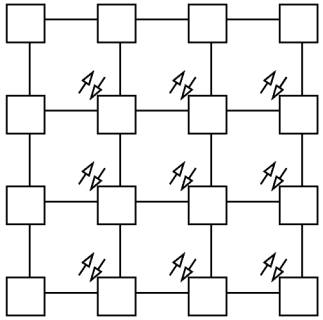
(a) XY in 2D mesh



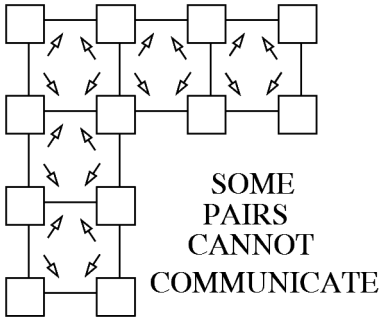
(b) SRh in 2D mesh



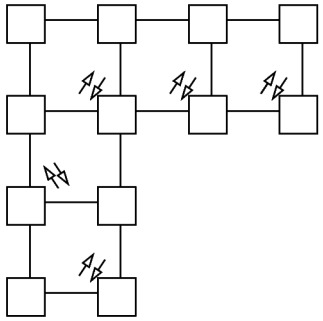
(c) SRv in 2D mesh



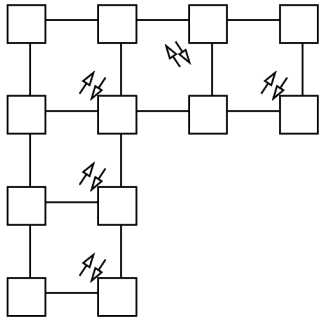
(d) UD in 2D mesh



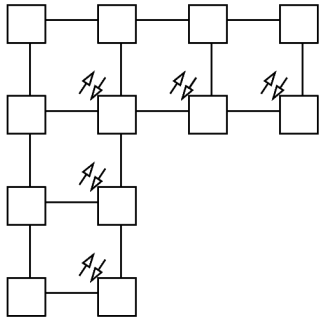
(e) XY in p topology



(f) SRh in p topology



(g) SRv in p topology



(h) UD in p topology

Agenda

Introduction

System environment

Description

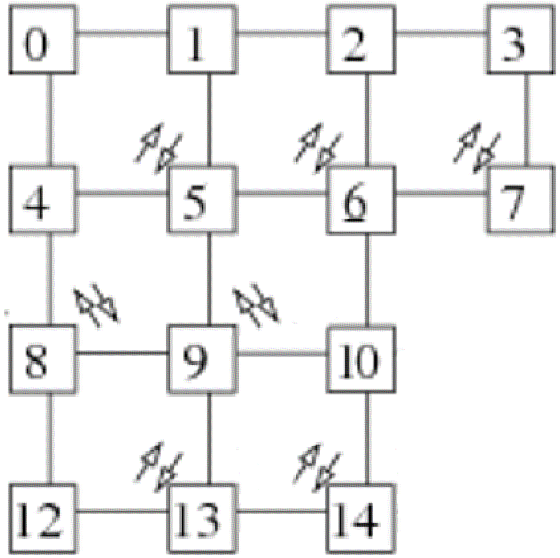
Evaluation

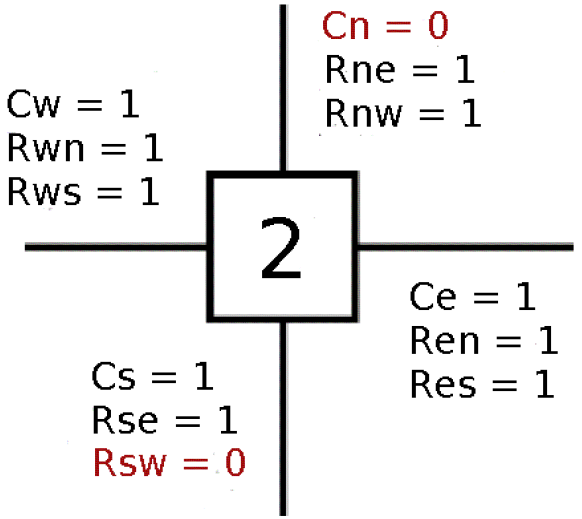
[Further evaluations]

Conclusions

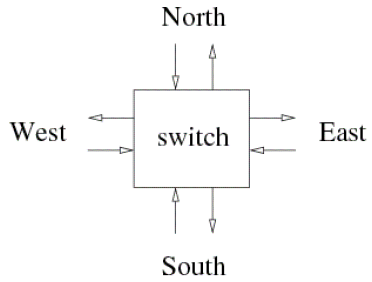
Description

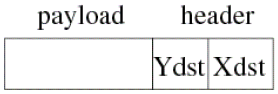
- LBDR uses two sets of bits:
  - Routing bits (Rxy), 2 per each output port
  - Connectivity bits (Cx), 1 per each output port
- The four output ports are labeled as N, E, W and S





Description (2)

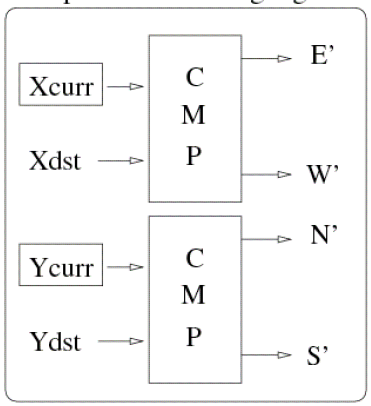




Routing and connectivity bits required per switch (12 bits, 3 per output port)

bits at the North port	Rne, Rnw	bits at the West port	Rwn, Rws		Cn, Ce
bits at the East port	Ren, Res	bits at the South port	Rse, Rsw		Cw, Cs

First part of the routing logic



Second part of the routing logic

$N'' = N' \cdot \overline{E'} \cdot \overline{W'} + N' \cdot E' \cdot Rne + N' \cdot W' \cdot Rnw$

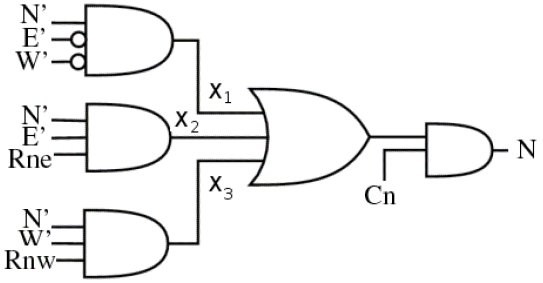
$E'' = E' \cdot \overline{N'} \cdot \overline{S'} + E' \cdot N' \cdot Ren + E' \cdot S' \cdot Res$

$W'' = W' \cdot \overline{N'} \cdot \overline{S'} + W' \cdot N' \cdot Rwn + W' \cdot S' \cdot Rws$

$S'' = S' \cdot \overline{E'} \cdot \overline{W'} + S' \cdot E' \cdot Rse + S' \cdot W' \cdot Rsw$

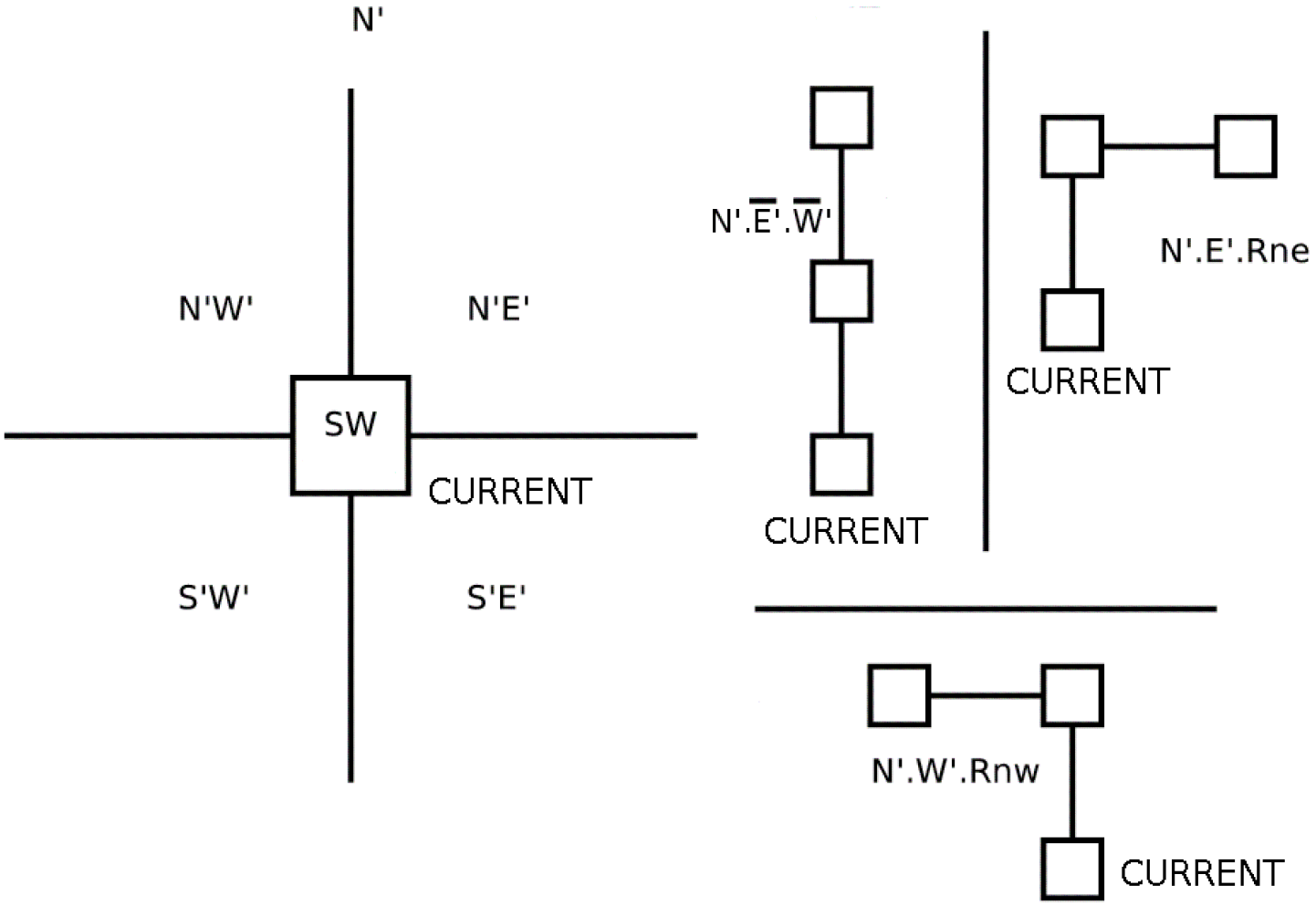
$N = N'' \cdot Cn$ $\quad W = W'' \cdot Cw$

$E = E'' \cdot Ce$ $\quad S = S'' \cdot Cs$



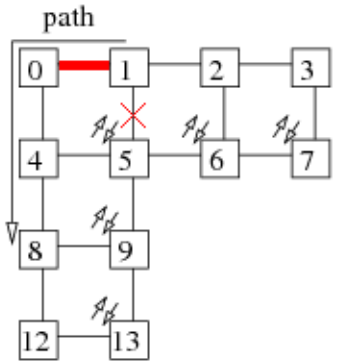
Logic block for N port

Description (3)



Description (4)

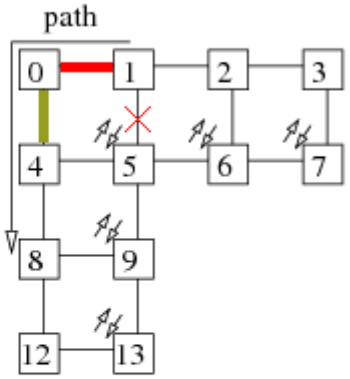
Switch	Rne	Rnw	Ren	Res	Rwn	Rws	Rse	Rsw	Cn	Ce	Cw	Cs
0	1	1	1	1	1	1	1	1	0	1	0	1
1	1	1	1	1	1	1	1	0	0	1	1	1
2	1	1	1	1	1	1	1	0	0	1	1	1
3	1	1	1	1	1	1	1	0	0	0	1	1
4	1	1	0	1	1	1	1	1	1	1	0	1
5	1	1	0	1	1	1	1	0	1	1	1	1
6	1	1	0	1	1	1	1	1	1	1	1	0
7	1	1	1	1	1	1	1	1	1	0	1	0
8	1	1	0	1	1	1	1	1	1	1	0	1
9	1	1	1	1	1	1	1	0	1	0	1	1
10												
11												
12	1	1	0	1	1	1	1	1	1	1	0	0
13	1	1	1	1	1	1	1	1	1	0	1	0
14												
15												



- 1st part of logic:
  - S'=1, W'=1
  - N'=0, E'=0
- 2nd part of logic
  - S''=0 (Rsw=0)
  - W''=1 (Rws=1, W'=1, S'=1)
- Final
  - W=1 (Cw=1) -> TO ARBITER

Description (5)

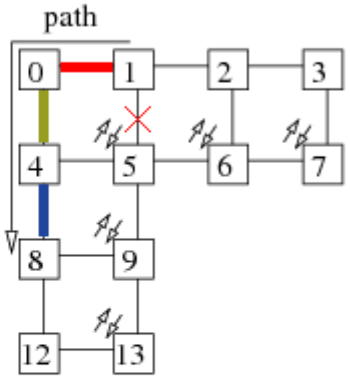
Switch	Rne	Rnw	Ren	Res	Rwn	Rws	Rse	Rsw	Cn	Ce	Cw	Cs
0	1	1	1	1	1	1	1	1	0	1	0	1
1	1	1	1	1	1	1	1	0	0	1	1	1
2	1	1	1	1	1	1	1	0	0	1	1	1
3	1	1	1	1	1	1	1	0	0	0	1	1
4	1	1	0	1	1	1	1	1	1	1	0	1
5	1	1	0	1	1	1	1	0	1	1	1	1
6	1	1	0	1	1	1	1	1	1	1	1	0
7	1	1	1	1	1	1	1	1	1	0	1	0
8	1	1	0	1	1	1	1	1	1	1	0	1
9	1	1	1	1	1	1	1	0	1	0	1	1
10												
11												
12	1	1	0	1	1	1	1	1	1	1	0	0
13	1	1	1	1	1	1	1	1	1	0	1	0
14												
15												



- 1st part of logic:
  - S'=1
  - W'=0, N'=0, E'=0
- 2nd part of logic
  - S''=1 (S'=1, E'=0, W'=0)
- Final
  - S=1 (Cs=1) -> TO ARBITER

Description (6)

Switch	Rne	Rnw	Ren	Res	Rwn	Rws	Rse	Rsw	Cn	Ce	Cw	Cs
0	1	1	1	1	1	1	1	1	0	1	0	1
1	1	1	1	1	1	1	1	0	0	1	1	1
2	1	1	1	1	1	1	1	0	0	1	1	1
3	1	1	1	1	1	1	1	0	0	0	1	1
4	1	1	0	1	1	1	1	1	1	1	0	1
5	1	1	0	1	1	1	1	0	1	1	1	1
6	1	1	0	1	1	1	1	1	1	1	1	0
7	1	1	1	1	1	1	1	1	1	0	1	0
8	1	1	0	1	1	1	1	1	1	1	0	1
9	1	1	1	1	1	1	1	0	1	0	1	1
10												
11												
12	1	1	0	1	1	1	1	1	1	1	0	0
13	1	1	1	1	1	1	1	1	1	0	1	0
14												
15												

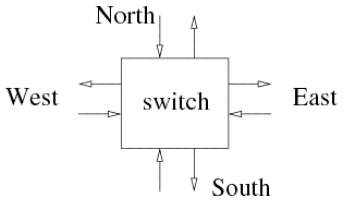


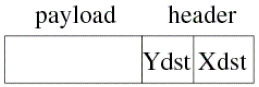
- 1st part of logic:
  - S'=1
  - W'=0, N'=0, E'=0
- 2nd part of logic
  - S''=1 (S'=1, E'=0, W'=0)
- Final
  - S=1 (Cs=1) -> TO ARBITER

Description (7)

- LBDR has visibility of one hop away -> LBDRe expands visibility to two hops away
- LBDRe adds four more bits per ouput port. It is a second set of routing bits (R2xy), meaning that y direction can be taken two hops away through the x direction

Description (8)

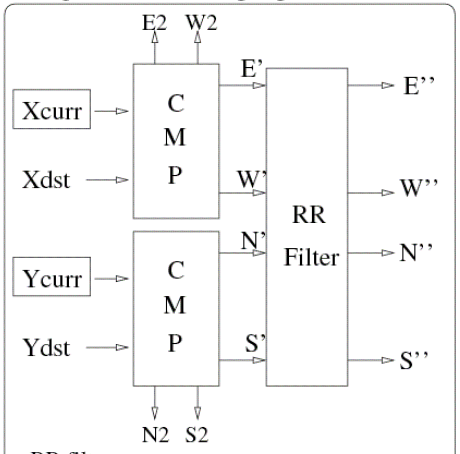




Routing and connectivity bits required per switch (24 bits, 6 per output port)

bits at the North port	Rne	R2ne	RRen	bits at the West port	Rwn	R2wn	RRnw	Cn			
	Rnw	R2nw	RRwn		Rws	R2ws	RRsw	Ce			
bits at the East port	Ren	R2en	RRne	bits at the South port	Rse	R2se	RRes	Cw			
	Res	R2es	RRse		Rsw	R2sw	RRws	Cs			

First part of the routing logic



RR filter

$N'' = N' \cdot (ipE \cdot \overline{RRen} + ipW \cdot \overline{RRwn} + ipS + ipL)$
$E'' = E' \cdot (ipN \cdot \overline{RRne} + ipS \cdot \overline{RRse} + ipW + ipL)$
$W'' = W' \cdot (ipN \cdot \overline{RRnw} + ipS \cdot \overline{RRsw} + ipE + ipL)$
$S'' = S' \cdot (ipE \cdot \overline{RRes} + ipW \cdot \overline{RRws} + ipN + ipL)$

Second part of the routing logic

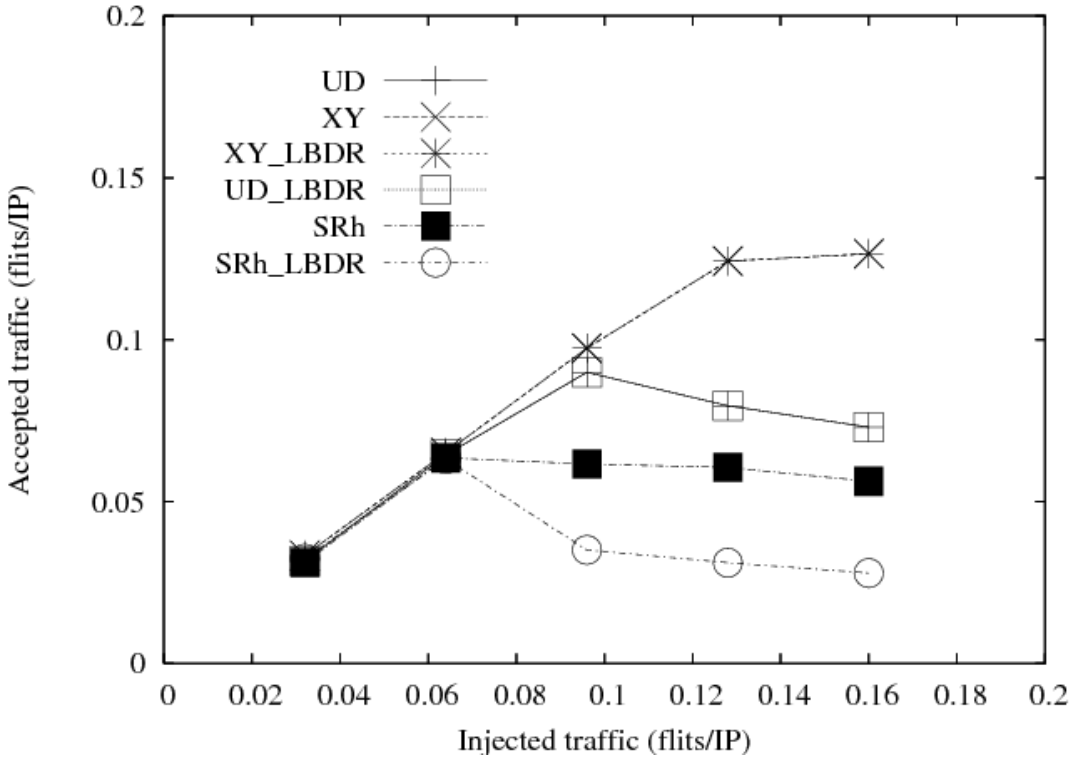
$N''' = N' \cdot \overline{E'} \cdot \overline{W'} + N' \cdot E' \cdot Rne + N' \cdot W' \cdot Rnw + N2 \cdot E' \cdot R2ne + N2 \cdot W' \cdot R2nw$
$E''' = E' \cdot \overline{N'} \cdot \overline{S'} + E' \cdot N' \cdot Ren + E' \cdot S' \cdot Res + E2 \cdot N' \cdot R2en + E2 \cdot S' \cdot R2es$
$W''' = W' \cdot \overline{N'} \cdot \overline{S'} + W' \cdot N' \cdot Rwn + W' \cdot S' \cdot Rws + W2 \cdot N' \cdot R2wn + W2 \cdot S' \cdot R2ws$
$S''' = S' \cdot \overline{E'} \cdot \overline{W'} + S' \cdot E' \cdot Rse + S' \cdot W' \cdot Rsw + S2 \cdot E' \cdot R2se + S2 \cdot W' \cdot R2sw$

$N = N'' \cdot Cn \cdot N'''$ $\quad W = W'' \cdot Cw \cdot W'''$
$E = E'' \cdot Ce \cdot E'''$ $\quad S = S'' \cdot Cs \cdot S'''$

(*) For further details of the full logic, please refer to the paper

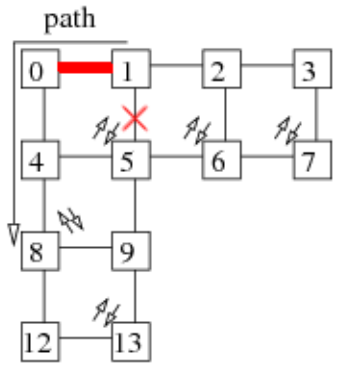
Description (9)

- Why LBDRe?



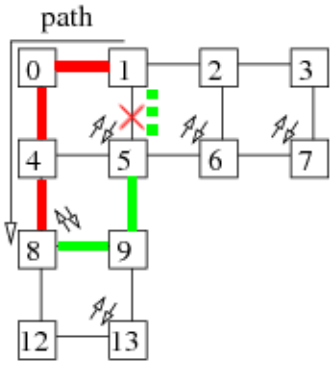
Description (10)

Switch	Rne	Rnw	Ren	Res	Rwn	Rws	Rse	Rsw	Cn	Ce	Cw	Cs
0	1	1	1	1	1	1	1	1	0	1	0	1
1	1	1	1	1	1	1	1	0	0	1	1	1
2	1	1	1	1	1	1	1	0	0	1	1	1
3	1	1	1	1	1	1	1	0	0	0	1	1
4	1	1	0	1	1	1	0	1	1	1	0	1
5	1	1	0	1	1	1	1	1	1	1	1	1
6	1	1	0	1	1	1	1	1	1	1	1	0
7	1	1	1	1	1	1	1	1	1	0	1	0
8	1	1	1	1	1	1	1	1	1	1	0	1
9	1	1	1	1	0	1	1	0	1	0	1	1
10												
11												
12	1	1	0	1	1	1	1	1	1	1	0	0
13	1	1	1	1	1	1	1	1	1	0	1	0
14												
15												



Description (11)

Switch	Rne	Rnw	Ren	Res	Rwn	Rws	Rse	Rsw	Cn	Ce	Cw	Cs
0	1	1	1	1	1	1	1	1	0	1	0	1
1	1	1	1	1	1	1	1	0	0	1	1	1
2	1	1	1	1	1	1	1	0	0	1	1	1
3	1	1	1	1	1	1	1	0	0	0	1	1
4	1	1	0	1	1	1	0	1	1	1	0	1
5	1	1	0	1	1	1	1	1	1	1	1	1
6	1	1	0	1	1	1	1	1	1	1	1	0
7	1	1	1	1	1	1	1	1	1	0	1	0
8	1	1	1	1	1	1	1	1	1	1	0	1
9	1	1	1	1	0	1	1	0	1	0	1	1
10												
11												
12	1	1	0	1	1	1	1	1	1	1	0	0
13	1	1	1	1	1	1	1	1	1	0	1	0
14												
15												



Agenda

Introduction

System environment

Description

Evaluation

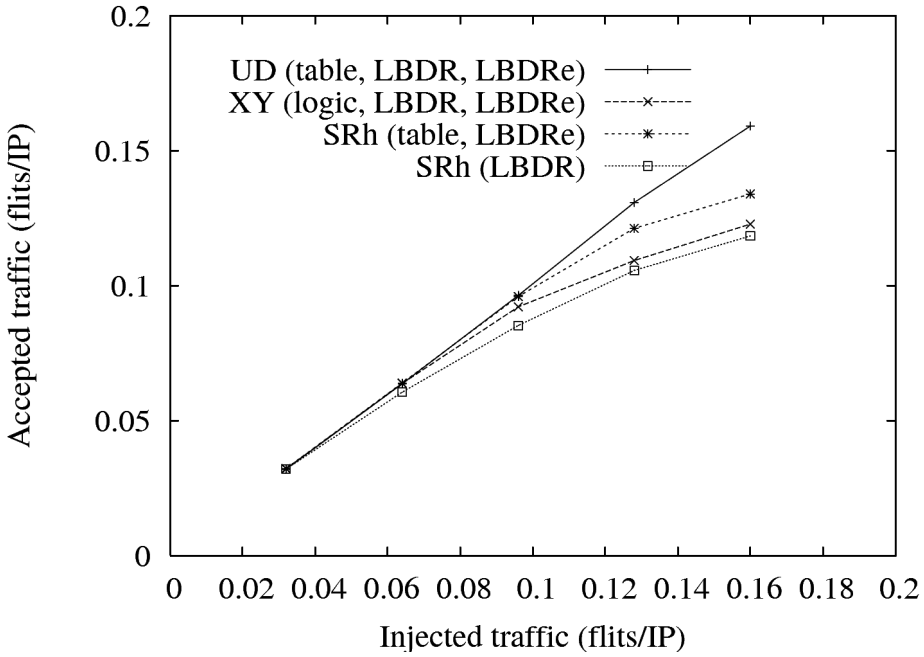
[Further evaluations]

Conclusions

Evaluation

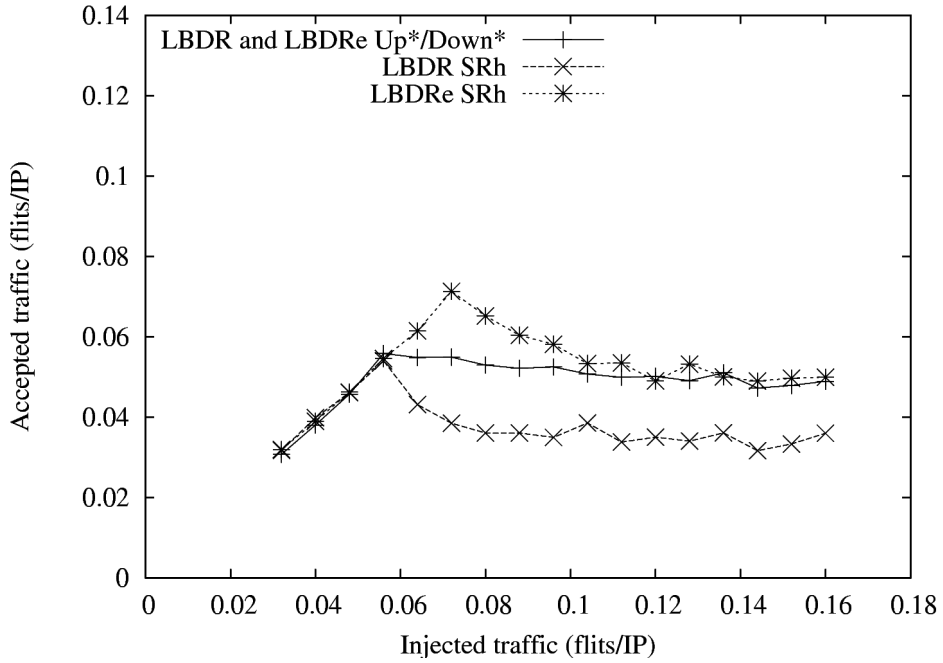
- NOXIM Simulator
  - Wormhole switching
  - Input port buffer 4-flit long
  - Packets 32-flit long
  - 8x8 mesh with different irregular topologies
  - XY, UD and SRh routing algorithms

Evaluation (2)



- Performance achieved for different routing algorithms on a 2D mesh

Evaluation (3)



- Comparison of performance for LBDR and LBDRe

Agenda

Introduction

System environment

Description

Evaluation

[Further evaluations]

Conclusions

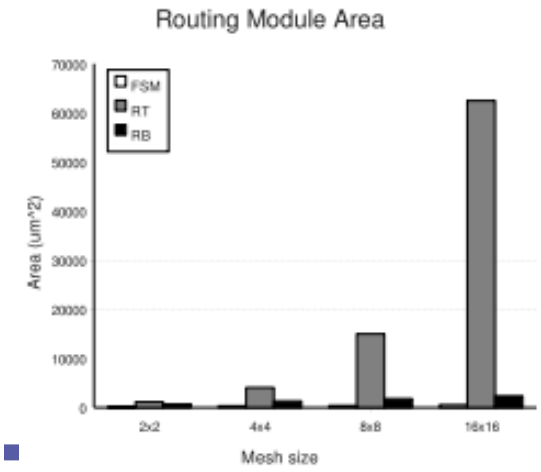
Further evaluations

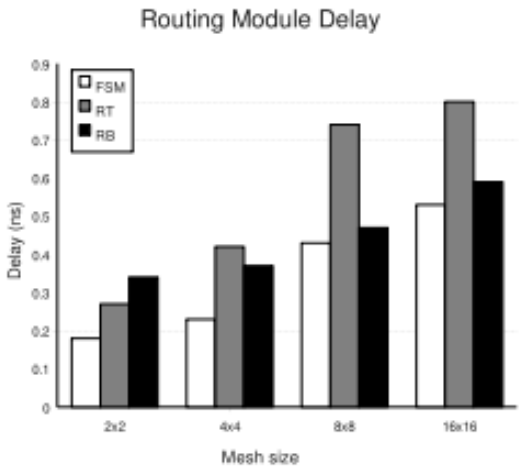
- Study on impact on area, power and delay constraints
- Evaluations achieved with much more detail using Synopsys Design Compiler and 90nm technology library from TSMC
- Good expectations. Region-Based Routing(*), with much more logic implied than LBDR, gets better results than implemented tables

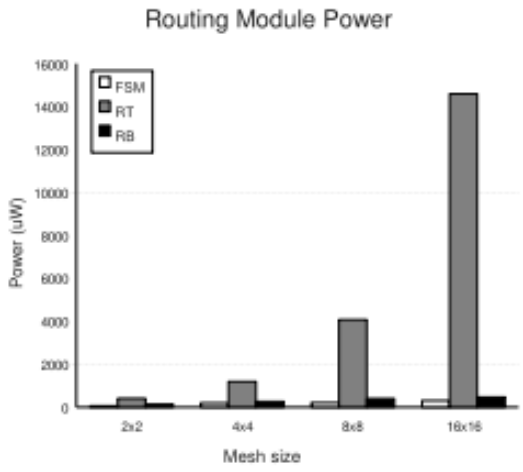
(*) Region-Based Routing: An Efficient Routing Mechanism to Tackle Unreliable Hardware in Network on Chips, NoCs 2007

Further evaluations (2)

- Minimum logic (*n* x *n* 2D mesh, d ports):
  - Table-based: *n x n* x d x d bits
  - RBR: 4 comparators, 4 registers $\log_2(N)/2$ bits, 1 register d+1 bits, 1 register d bits
  - LBDR: 12 bits per switch (3 per output port), 2 comparators, 2 inverters and 5 gates







Agenda

Introduction

System environment

Description

Evaluation

[Further evaluations]

Conclusions

Conclusions

- LBDR (and LBDRe) allows for implementing most of the distributed routing algorithms in suitable topologies for NoCs.
- Future work:
  - Applicability on system/chip virtualization
  - Support non-minimal paths
  - Broadcast


01000001010000110
01000001110000110
ADVANCED COMMUNICATION & COMPUTER ARCHITECTURE


MINISTERIO
DE EDUCACIÓN
Y CIENCIA


Consolider

Thank you.