Deterministic versus Adaptive Routing in Fat-Trees

C. Gómez, F. Gilabert, M.E. Gómez, P.López and J. Duato

Dep of Computer Engineering (DISCA)
Parallel Architectures Group
Universidad Politécnica de Valencia
(SPAIN)

Objective

- To propose a deterministic routing algorithm for fat-trees with a similar performance to the adaptive routing algorithm commonly-used in fat-trees
  - Implicit in-order delivery
  - More simple hardware implementation
- To provide a memory-efficient implementation for the routing algorithms presented in this work

Outline

- Introduction
- Fat-trees
- Proposed Deterministic Routing Algorithm
- Performance Evaluation
- Conclusions

Introduction

- Clusters of PCs have grown in popularity in the last years due to their excellent cost-performance ratio
- The interconnection network is important
- The chosen topology is usually regular:
  - Direct Networks
  - Indirect Networks

Introduction

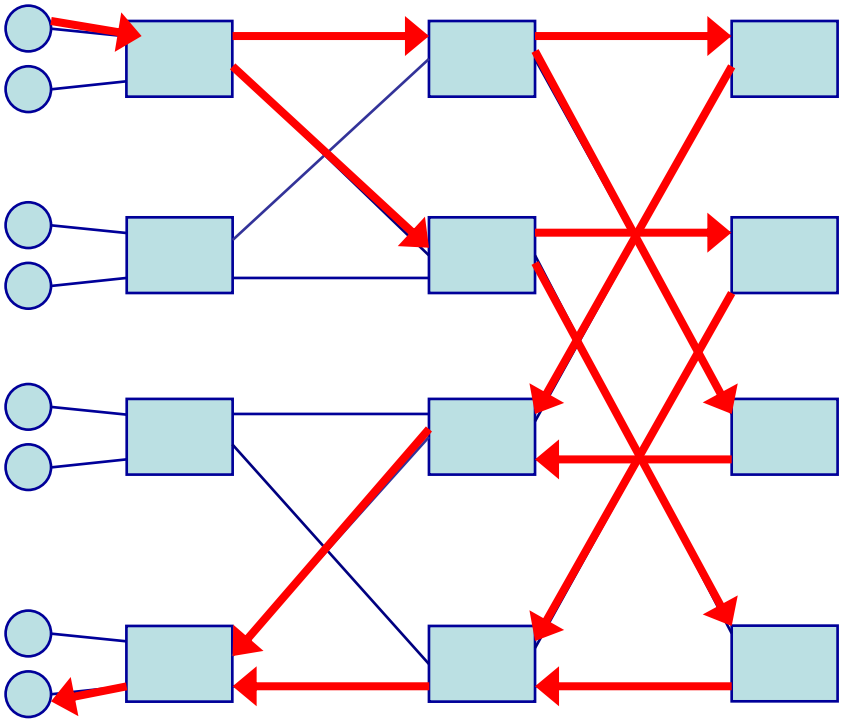
- Multistage interconnection networks (MIN) are a type of indirect regular topology
- All switches are identical and organized as a set of stages
- Each stage is only connected to the previous and the next stage using regular connection patterns
- As long as high degree switches are available, multistage networks (MINs) have become very popular
  - Among them, the fat-tree topology is the preferred choice

Introduction

	Mellanox InfiniBand	**Myricom Myrinet**	**Quadrics QsNet**	**Quadrics QsNetII**
Crossbar chip	24 port	16 port	8 port	8 port
Switch topology	**Fat tree**	**Fat tree**	**Fat tree**	**Fat tree**
Max size stand-alone switch	288	128 ports/9 RU	128 ports	128 ports
Host adapters	PCI-X	PCI-Xt	64 bit PCI 2.1	PCI-X
Port speeds	2.5, 10, 30 Gbps	2 Gbps	3.2 Gbps	10.6 Gbps
Throughput large messages >64B	6.6 Gbps (10 Gbps ports)	1.88 Gbps	2.5 Gbps	6.4 -7.2 Gbps
CPU utilization at max throughput	3%	6%	N/A	N/A
Send/Receive lat 16B message	5 µs	6.5 µs	2 µs	1.2 µs
Send/Receive lat 4B message	15 µs	32 µs	15 µs	N/A

Introduction

- Routing is one of the most important design issues of interconnection networks
- Deterministic vs Adaptive:

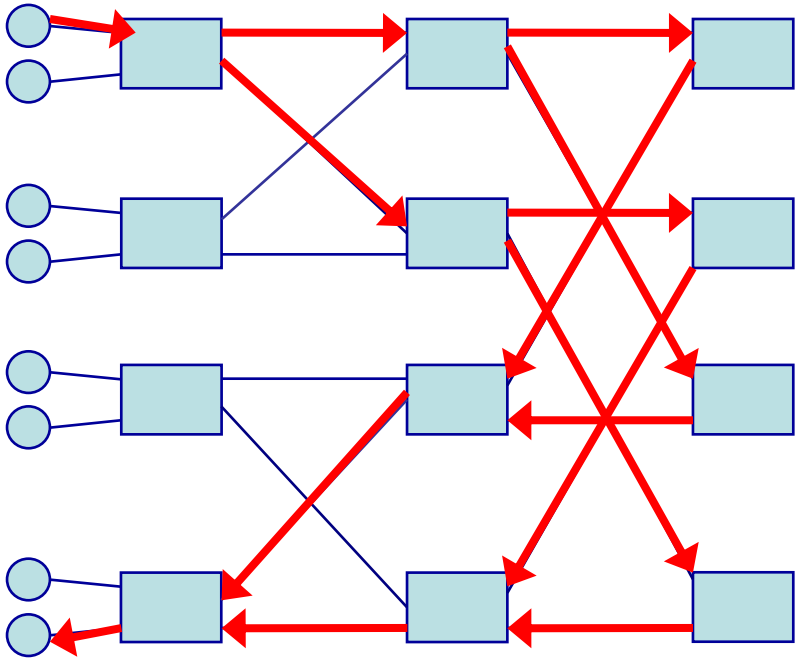


Deterministic path between two nodes

Adaptive paths between two nodes

Introduction

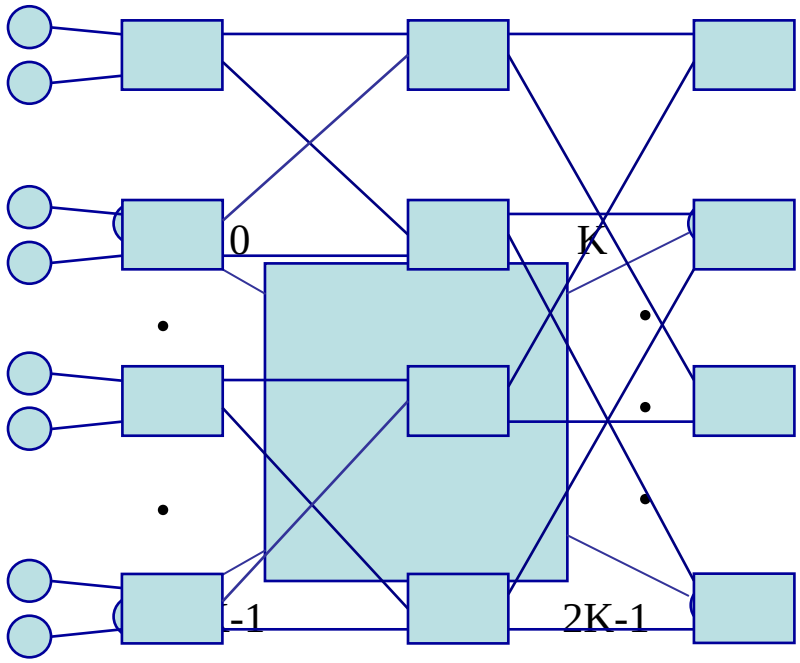
- Adaptive Routing
  - Multiple paths for any origin-destination pair
  - Need to select one path when a packet is routed
  - Selection Function



The selection function chooses one path for each packet transmitted.

Fat-tree

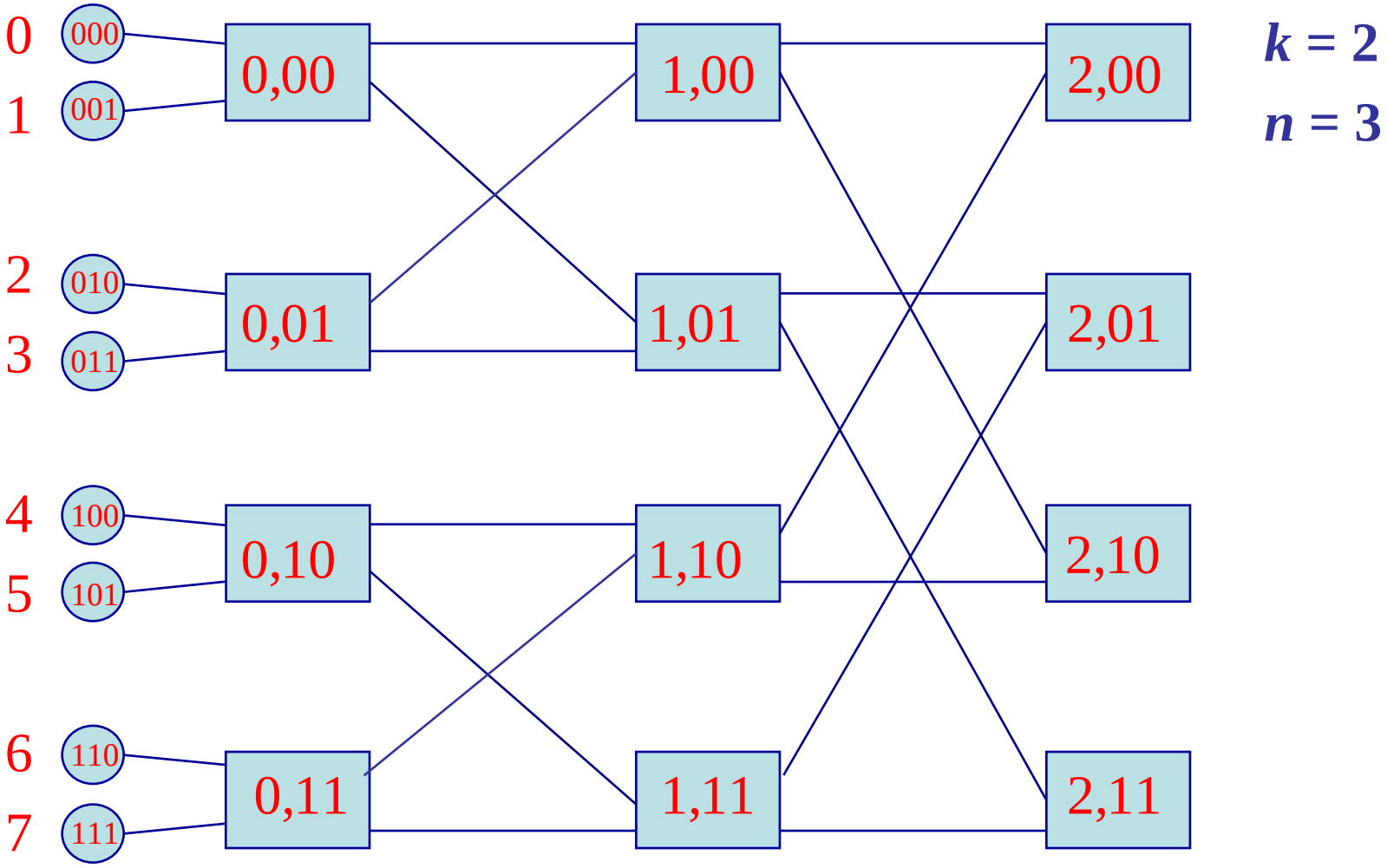
- A fat-tree topology is based on a complete tree
- Fat-trees get thicker near the root
- A set of processors is located at the leaves
- A *k*-ary *n*-tree:
  - switches with an arity of *k*
  - *n* stages
- Descending links will be labeled from *0* to *k-1*, and ascending links from *k* to *2k-1*



Fat-tree

- In a *k*-ary *n*-tree:
  - Each processing node is represented as a *n*-tuple $\{0,1,...,k-1\}^n$
  - Each switch is defined as a pair <*s*,*o*>:
    - ***S*** is the stage where the switch is located at. ***S*** in *{0..(n-1)}*
    - ***O*** is a *(n-1)*-tuple $\{0,1,...,k-1\}^{\{n-1\}}$

Fat-tree

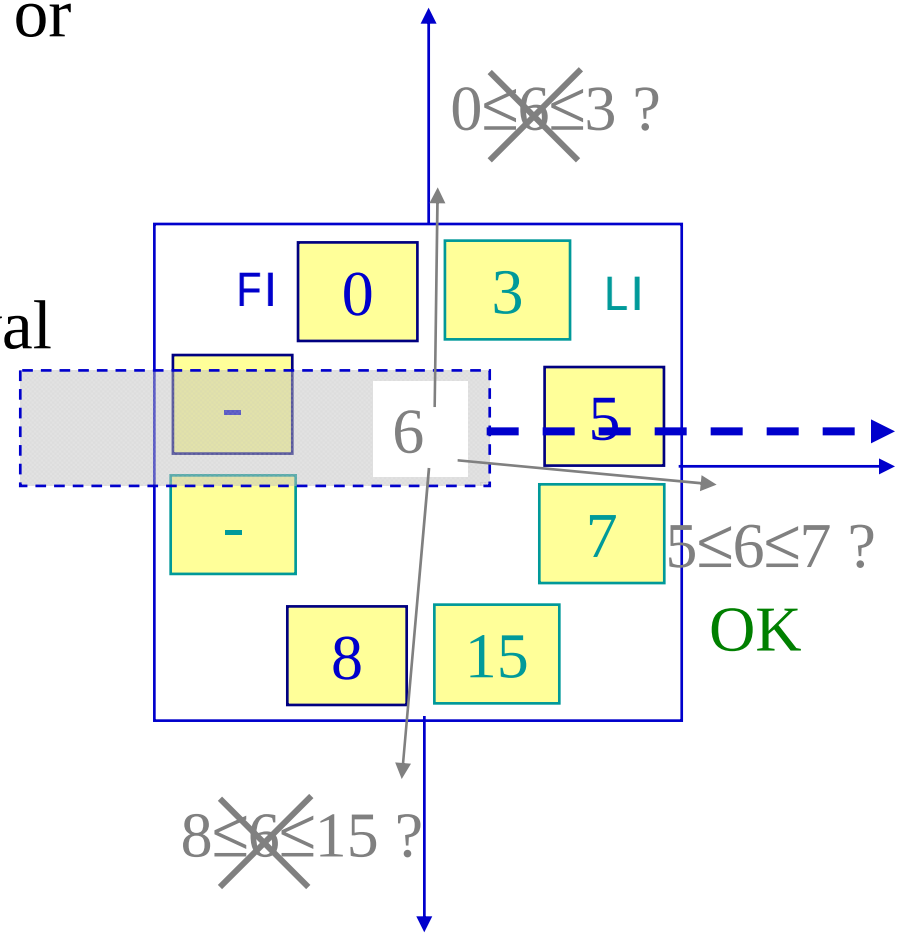

0
1
2
3
4
5
6
7
000
001
010
011
100
101
110
111
0,00
0,01
0,10
0,11
1,00
1,01
1,10
1,11
2,00
2,01
2,10
2,11
k = 2
n = 3

Fat-tree – Adaptive Routing

- The commonly-used adaptive algorithm in Fat-tree:
  - Accomplished in two phases:
    - Upwards phase fully adaptive.
    - Downwards phase is deterministic.
  - The unique downwards path to the destination depends on the switch that has been reached in the upwards phase.
  - The decisions made in the upwards phase are critical.

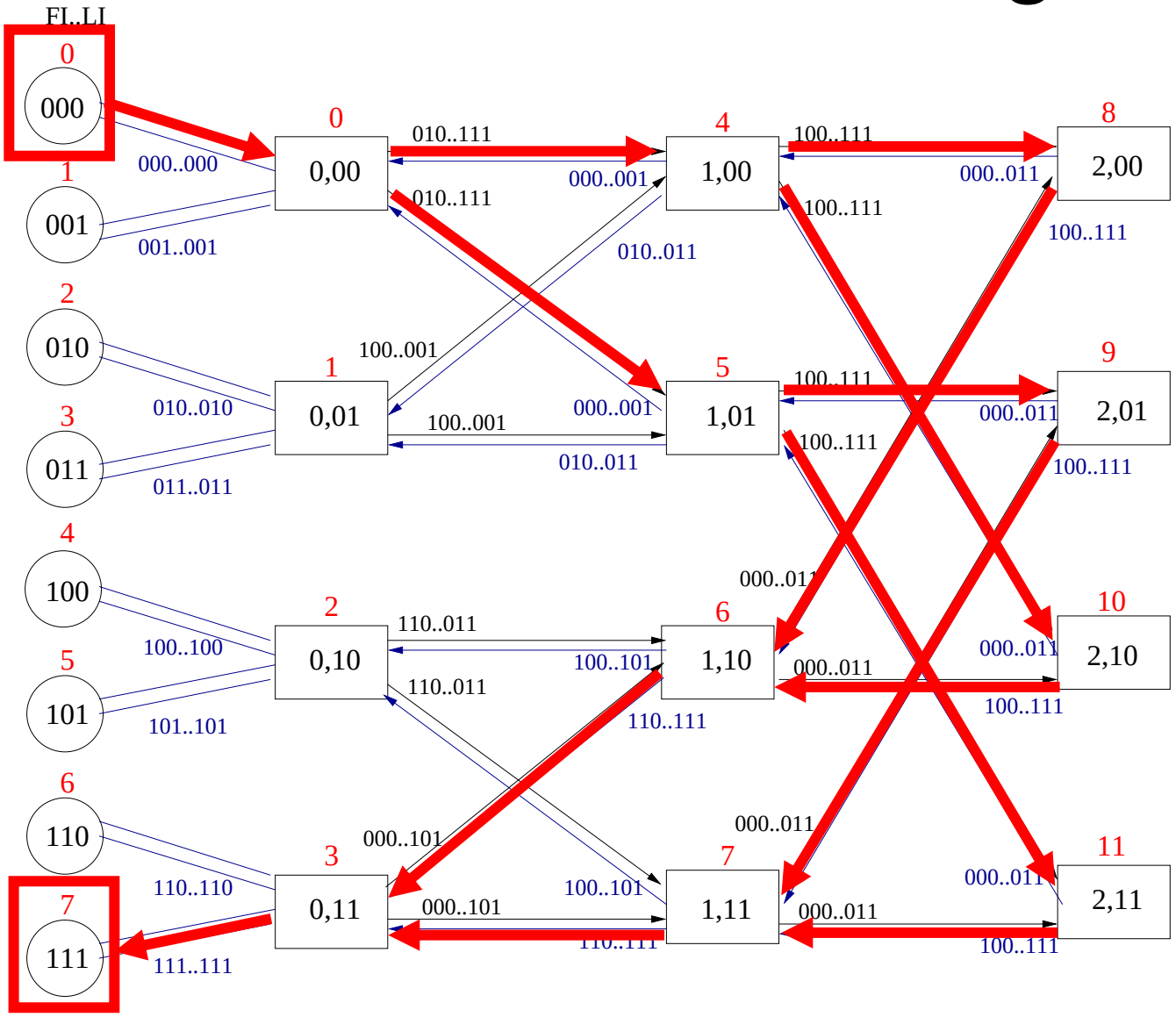
Fat-tree – Interval routing

- Each port has associated a range or *interval* of destinations
- A port can be used if packet destination id is inside the interval
- Implementation:
  - 2 registers (First Interval-FI, Last Interval-LI)
  - 2 comparators per port



Fat-tree – Interval routing

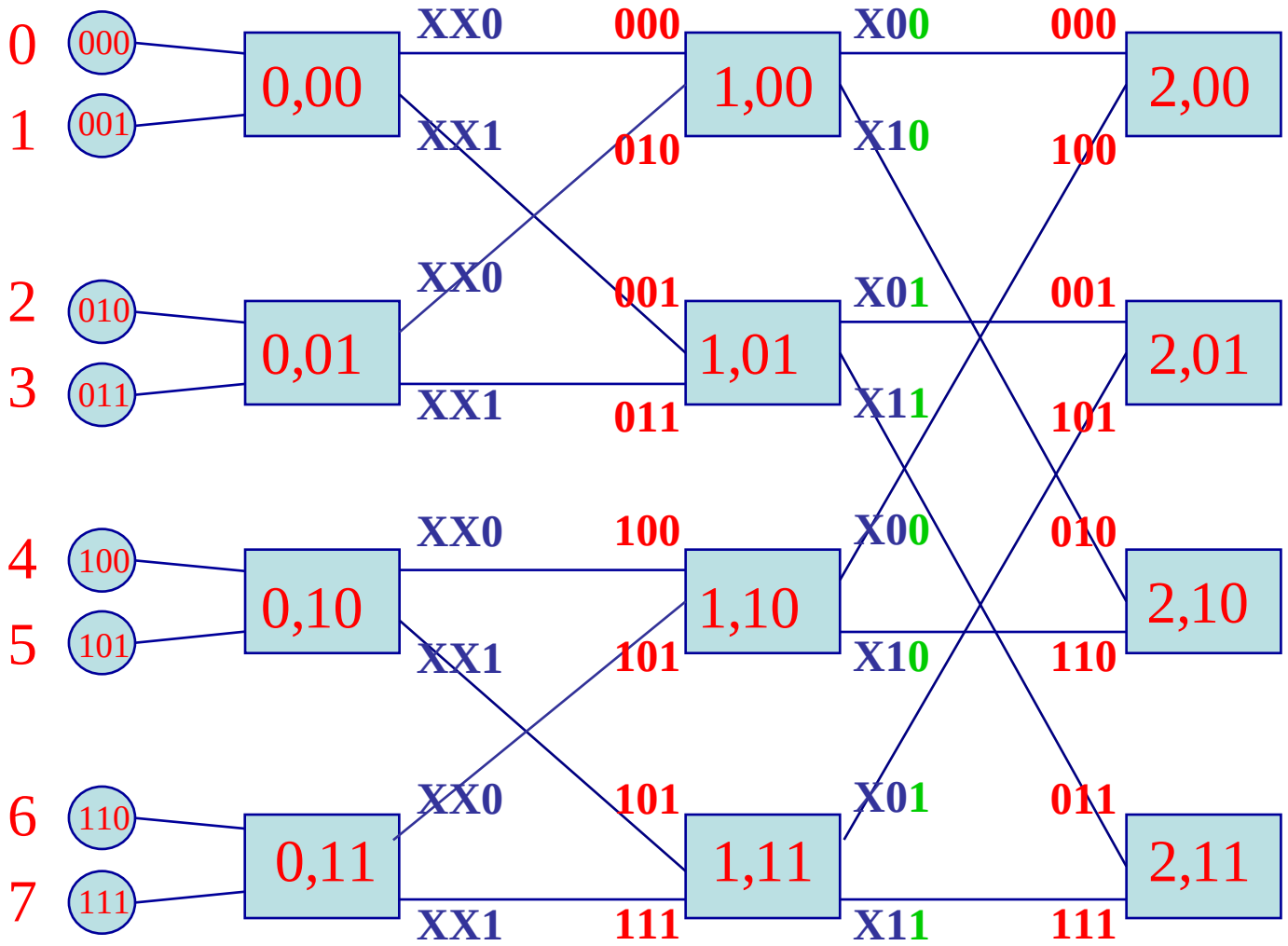
Message:
From **0**
To **7**



Deterministic Routing

- Reduce the multiple ascending paths to one
  - Trying to balance the traffic in the network
- At the switch $<s,o_{\{n-2\}},\ldots,o_1,o_0>$, the chosen output port for a packet with destination $p_{\{n-1\}},..p_s.,p_1,p_o$ will be $k+p_{s.}$

Deterministic Routing

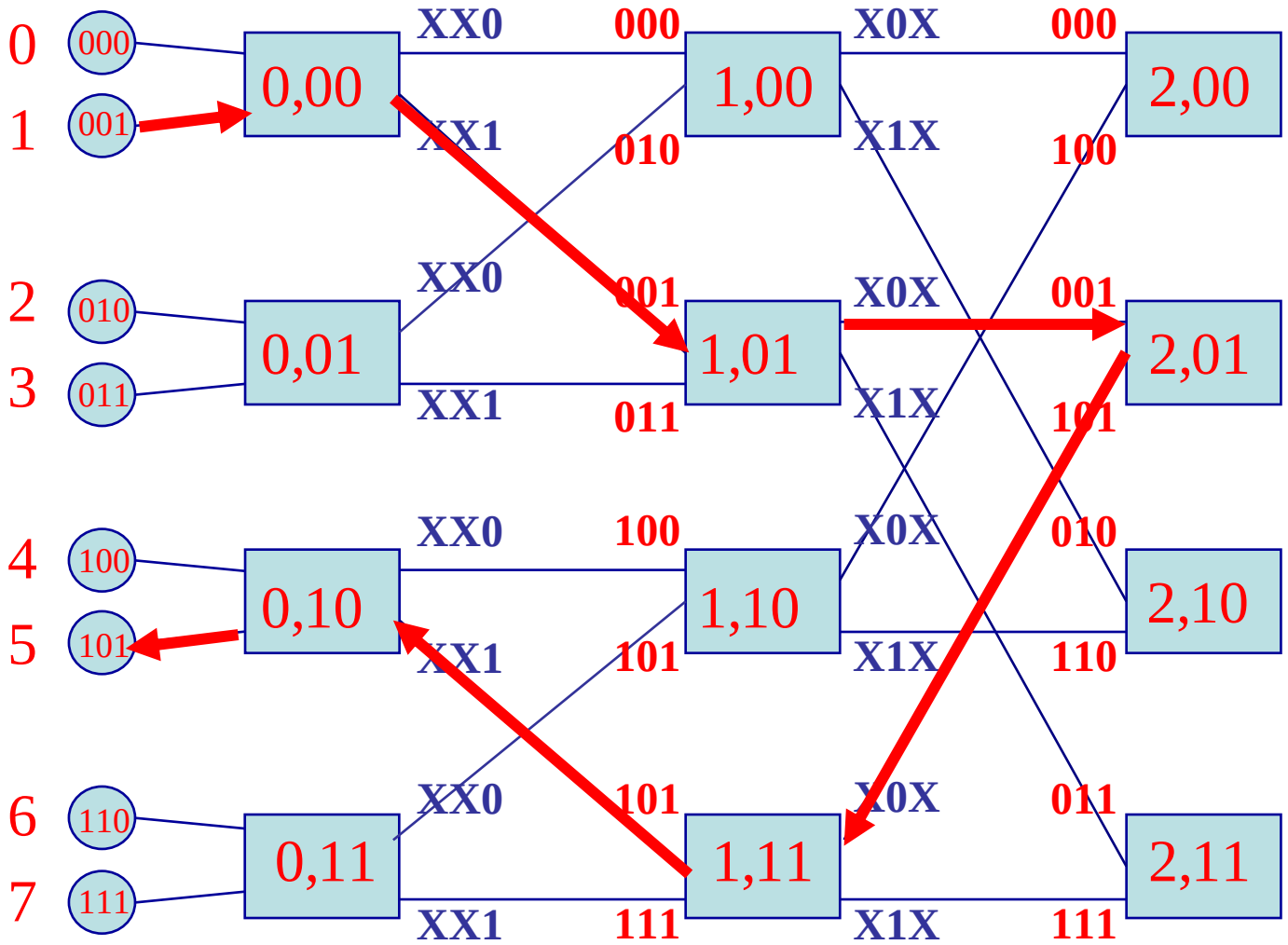


According to destination components

Blue: **Deterministic** ascending link

Red: Deterministic descending link

Deterministic Routing



Message

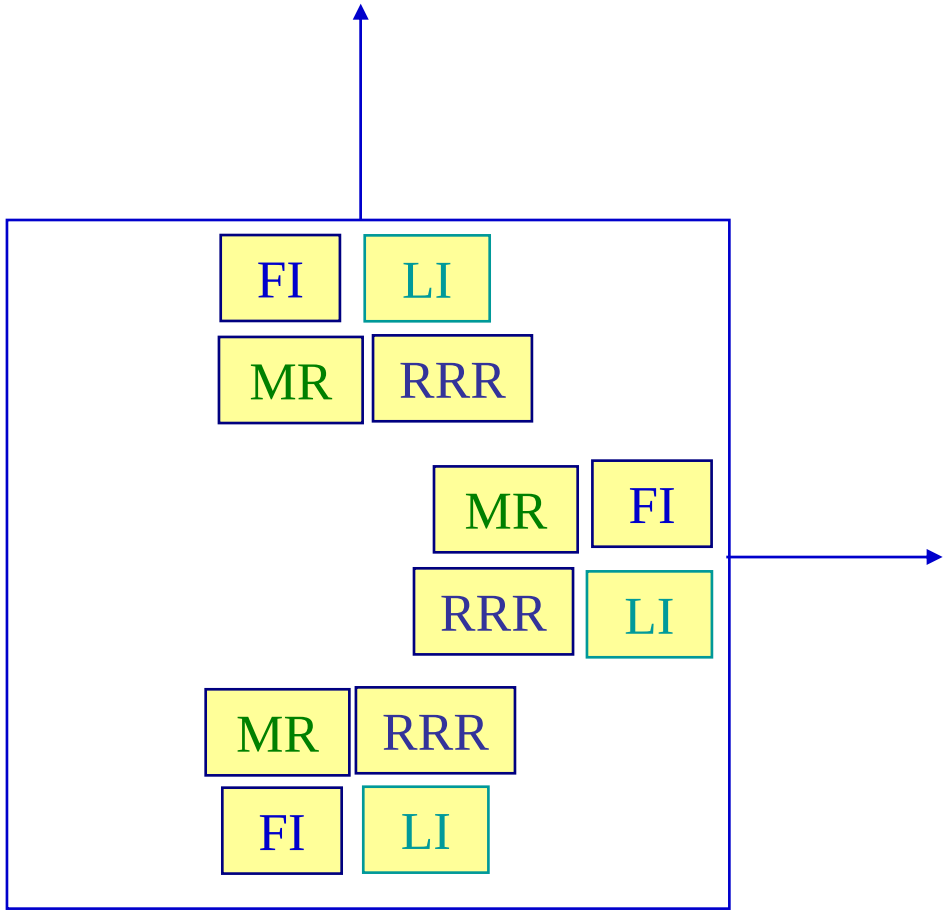
From 1 to 5

- Destination:

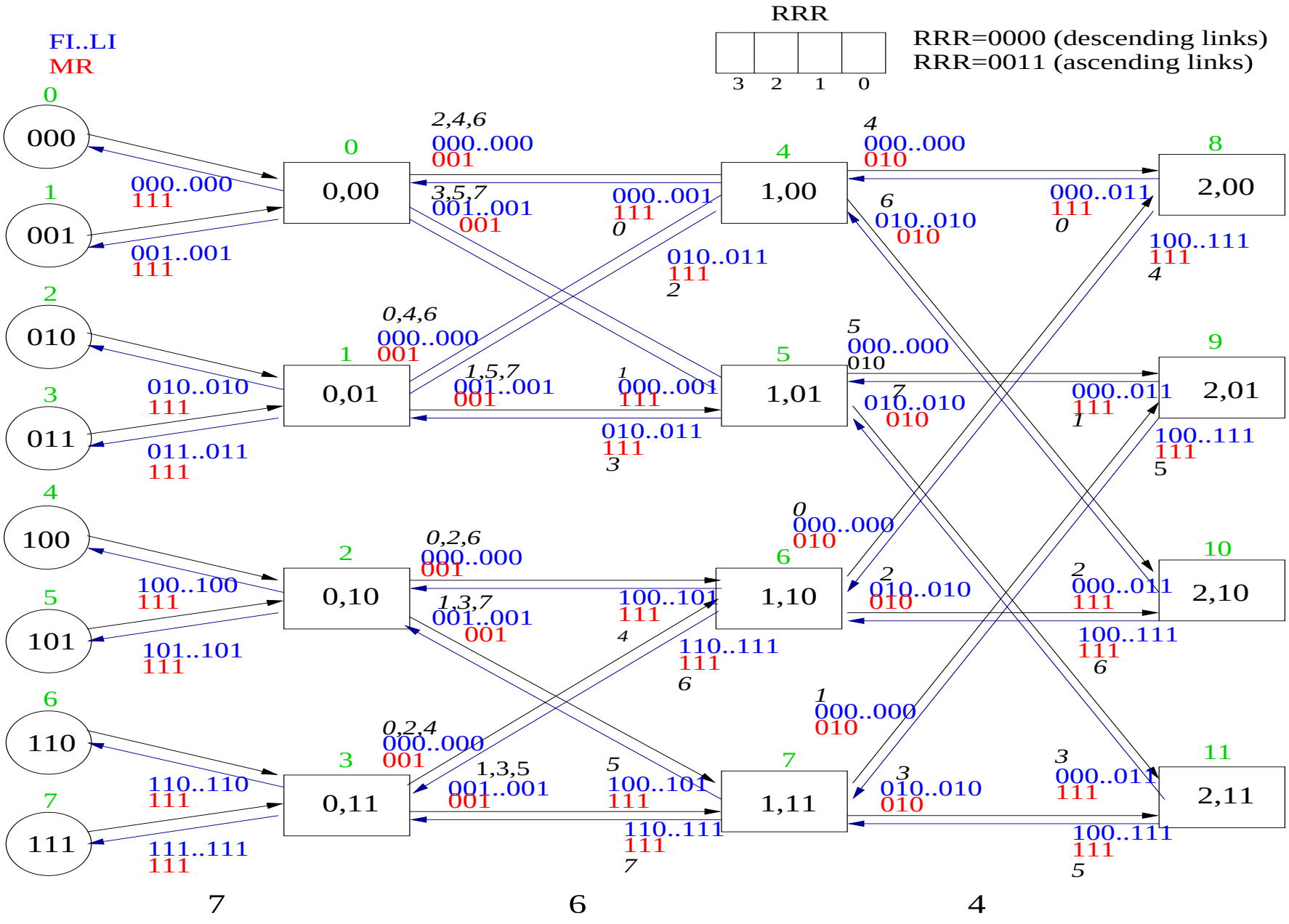
< 101 >

Deterministic Routing – Flexible Interval Routing

- Configuration registers, per port
  - FI: Lower bound of the interval.
  - LI: Upper bound of the interval.
  - MR: Selects the bits of the destination that are compared with FI,LI
  - RRR: Defines the priority of the output ports (1 bit per port)



Deterministic Routing – Flexible Interval Routing



Performance Evaluation

- In order to evaluate our deterministic algorithm, we will compare it with the commonly-used adaptive routing algorithm, using two different selection functions[1]:
  - First Free
  - Stage and Destination Priority

[1] F. Gilabert, M.E. Gómez, P. López, J. Duato. On the Influence of the Selection Function on the Performance of Fat-trees. *European Conf. on Parallel Computing*, Aug. 2006.

Performance Evaluation

- **First Free (FF):** The FF selection function selects the first physical link which is available
- This selection function achieves the worst performance results, but it is also the most simple one

F. Gilabert, M.E. Gómez, P. López, J. Duato. On the Influence of the Selection Function on the Performance of Fat-trees. *European Conf. on Parallel Computing*, Aug. 2006.

Performance Evaluation

- **Stage And Destination Priority (SADP):** The SADP selection function selects the physical link which will be chosen by our deterministic proposal
- If this link is not available, any other link is selected
- This selection function achieved the best performance results

F. Gilabert, M.E. Gómez, P. López, J. Duato. On the Influence of the Selection Function on the Performance of Fat-trees. *European Conf. on Parallel Computing*, Aug. 2006.

Performance Evaluation – Network Model

- To compare the deterministic algorithm proposed previously, a detailed event-driven simulator has been implemented
- The simulator models a *k*-ary *n*-tree virtual cut-through switching
- Each switch has a full crossbar with queues both at the input and output ports
- It takes 20 clock cycles to apply the routing algorithm and the selection function

Performance Evaluation – Network Model

- In-order delivery:
  - Deterministic: Implicitly provided
  - Adaptive: An ideal mechanism has been designed
    - Sequence number in the packet header
    - One buffer at each destination to store out-of-order packets
    - No packet is delivered to the processing node if the previous packets has not arrived
    - No additional delay
    - No retransmissions

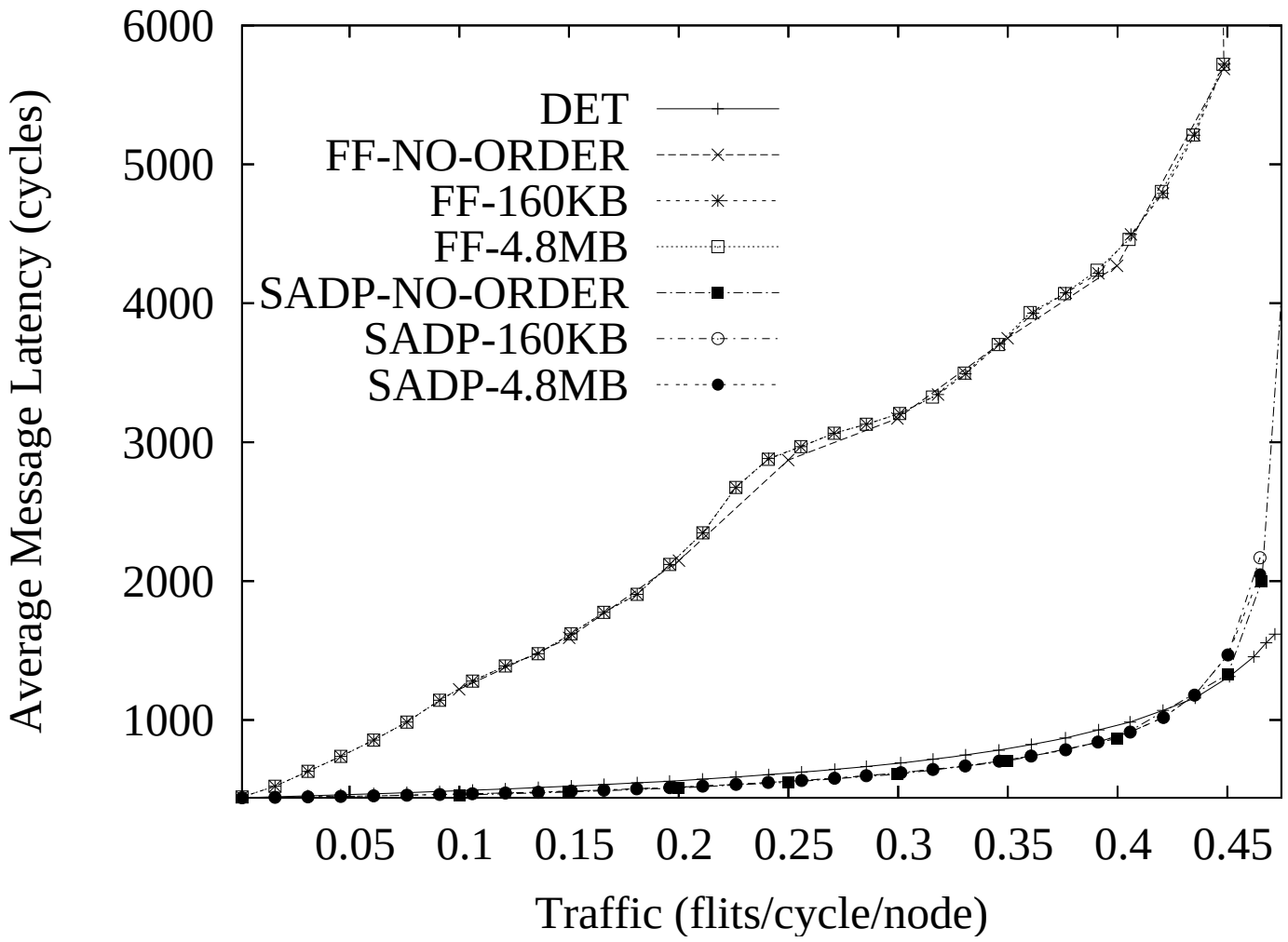
Performance Evaluation – Network Model

- Two traffic types: Synthetic and Traces
  - Synthetic traffic: Uniform, Bit-Reversal, Hot-Spot, Butterfly and Complement traffic pattern
    - All the traffic patterns, but the Bit-Reversal, present similar results
      - Bit-Reversal is the worst case of our strategy
        - » Only one up link of every switch is used in the deterministic case
  - Traces where provided by *HP* from the *cello* system
    - *Cello* system is a timesharing system with a storage subsystem
      - Traces from the I/O *cello* subsystem

Performance Evaluation

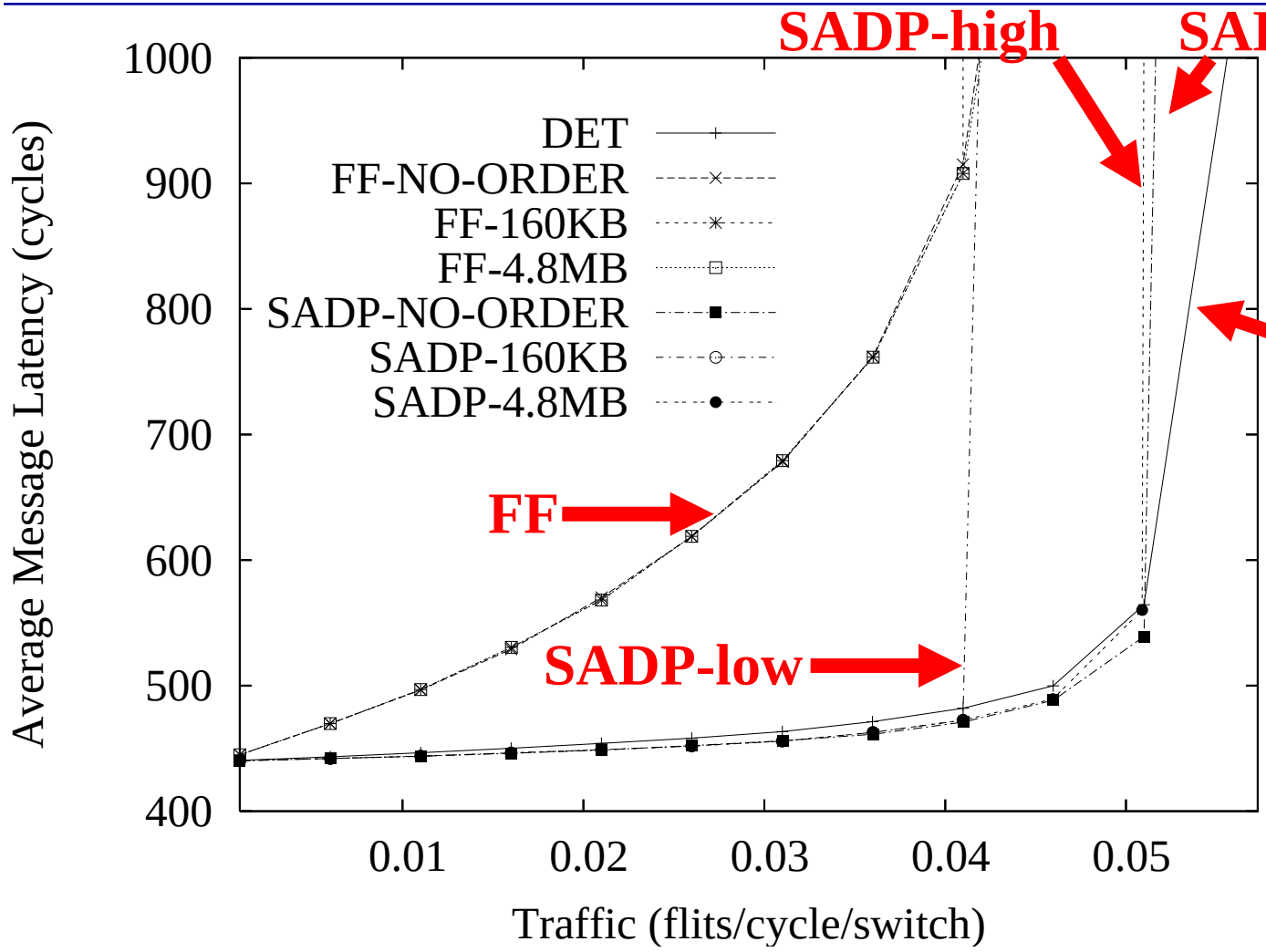
- We have evaluated a wide range of *k*-ary *n*-tree topologies.
  - From *2*-ary *2*-tree (4 nodes) to *2*-ary *8*-tree (256 nodes)
  - From *4*-ary *2*-tree (16 nodes) to *4*-ary *6*-tree (4096 nodes)
  - From *8*-ary *2*-tree (64 nodes) to *8*-ary *4*-tree (4096 nodes)
  - From *16*-ary *2*-tree (256 nodes) to *16*-ary *3*-tree (4096 nodes)
  - *32*-ary *2*-tree (1024 nodes).
- Due to space limitations, we show here only a subset of the most representative simulations.

Performance Evaluation - Uniform



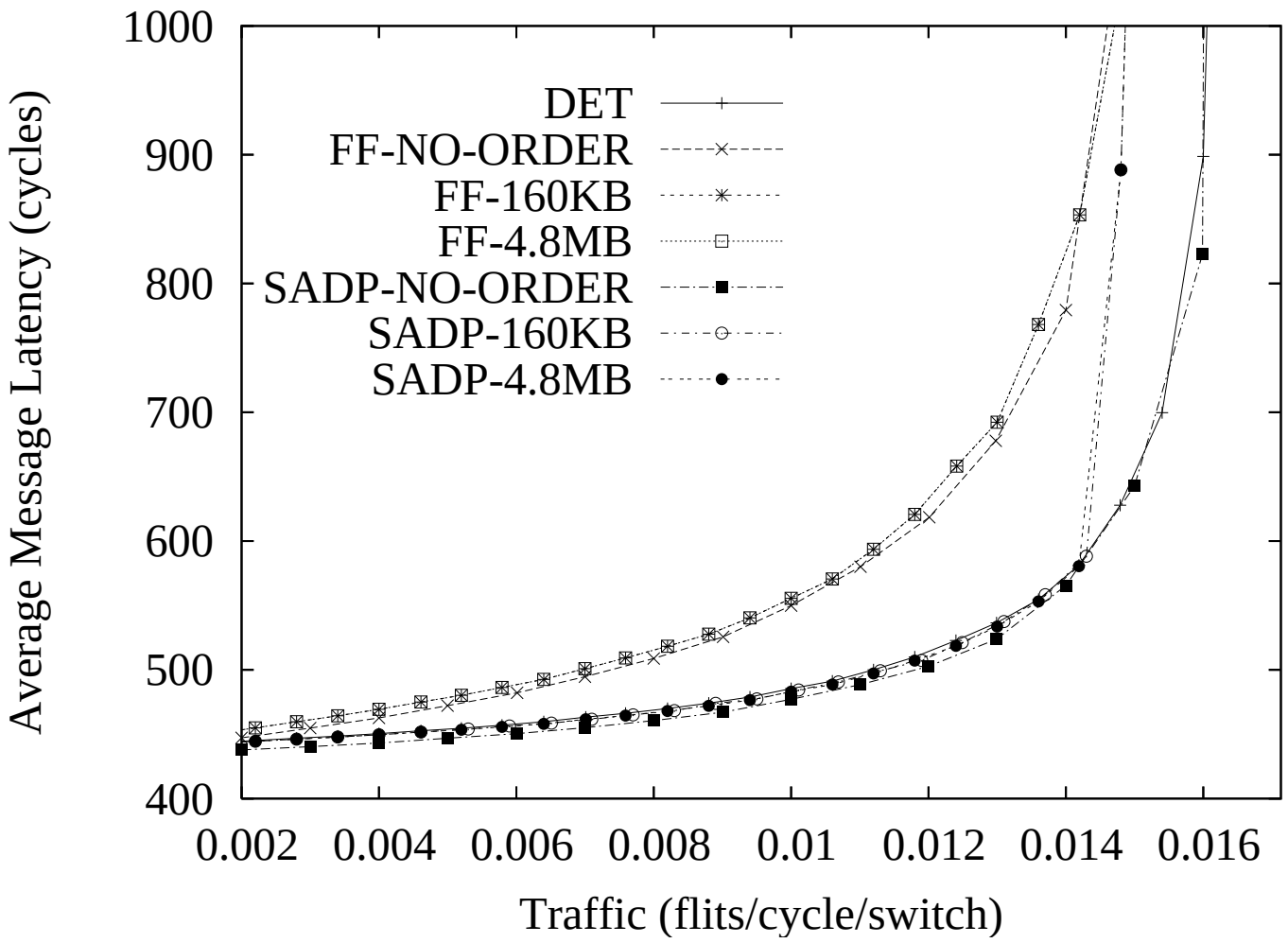
- 4-ary 4-tree
- 256 nodes

Performance Evaluation – Hot-Spot



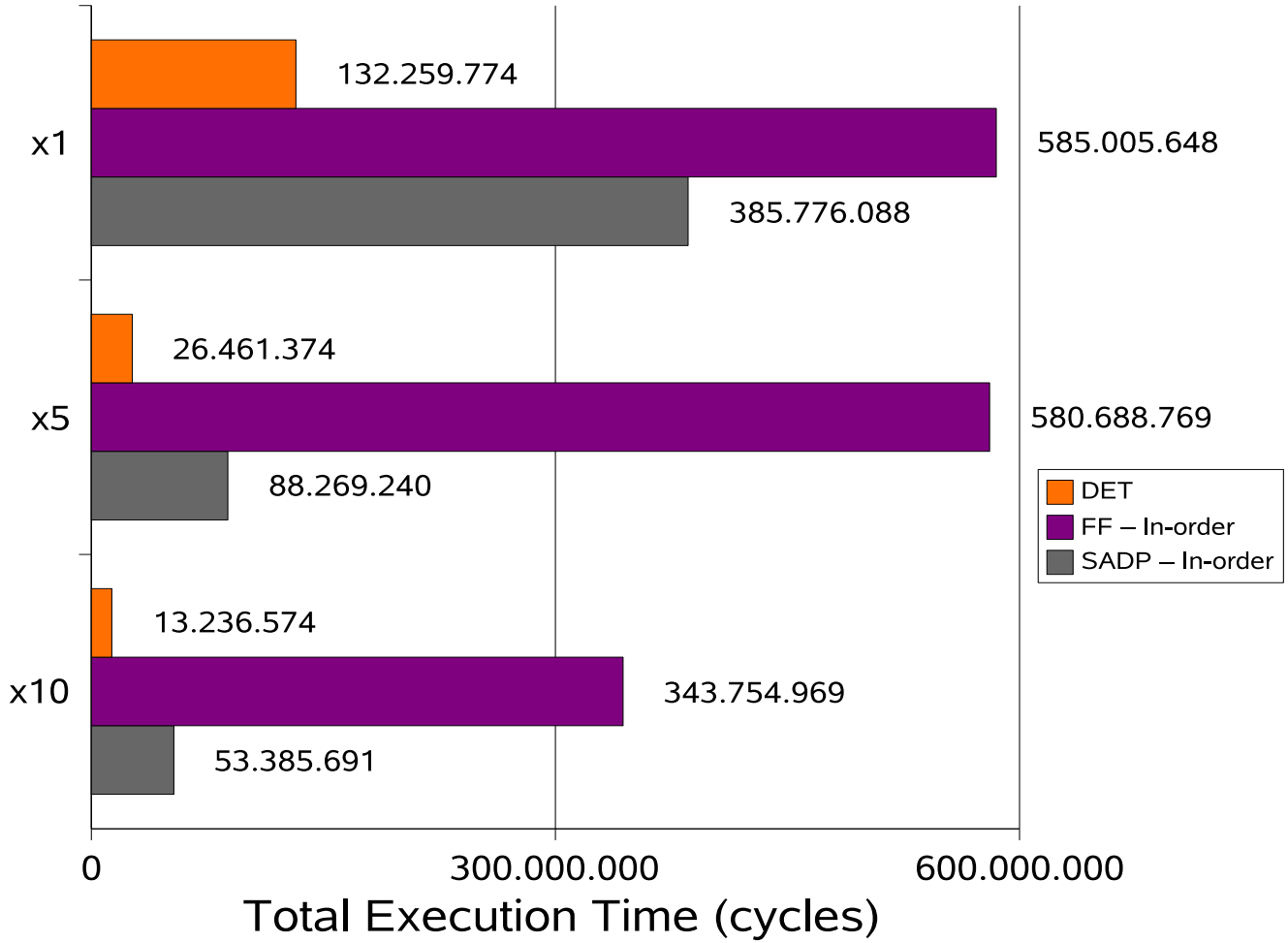
- 4-ary 4-tree
- 256 nodes
- Hot-Spot: 5%

Performance Evaluation – Hot-Spot



- 4-ary 4-tree
- 256 nodes
- Hot-Spot: 20%

Performance Evaluation – I/O Traces



- 2-ary 7-tree
- 128 nodes

Conclusions

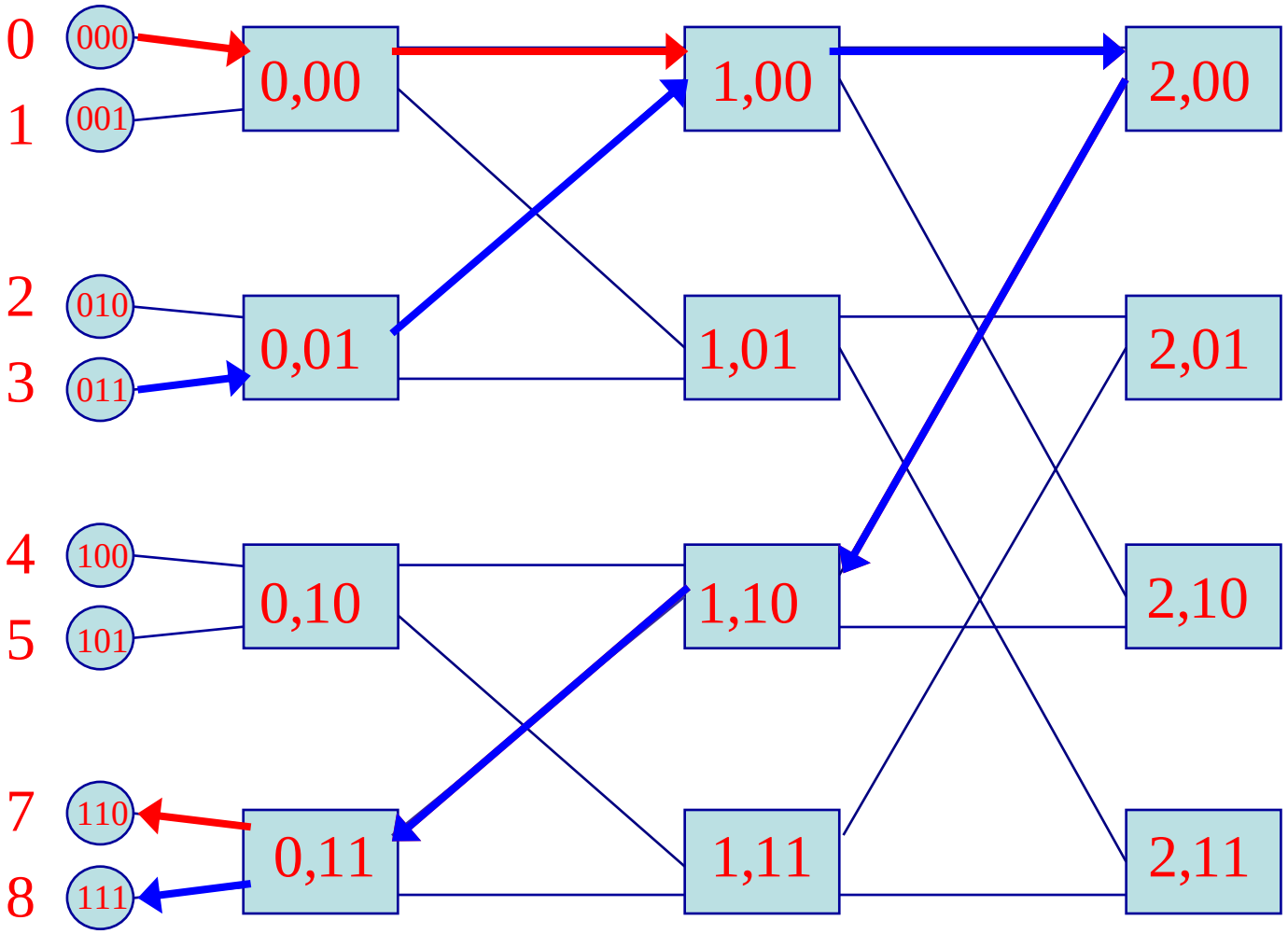
- Our deterministic routing algorithm provide similar results to the adaptive one when in-order delivery is not guaranteed
- When in-order delivery is guaranteed and traces are used, deterministic algorithm outperforms the adaptive ones
  - i.e. deterministic improves performance in a factor of 3 over SADP
- Deterministic algorithm is more simple:
  - No selection function
  - No additional hardware and protocols for in-order delivery

Deterministic versus Adaptive Routing in Fat-Trees

C. Gómez, F. Gilabert, M.E. Gómez, P.López and J. Duato

Dep of Computer Engineering (DISCA)
Parallel Architectures Group
Universidad Politécnica de Valencia
(SPAIN)

Selection Functions – FF



Assuming that the network is empty

One message sent from node 0 to node 7

One message sent from node 3 to node 8

Network Model

- Packet size is 8 Kb and packet generation rate is constant and the same for all the processors in the network.
- Two different traffic patterns: uniform and complement.
  - Uniform traffic pattern: message destination is randomly chosen among all the processors in the network
  - Complement traffic pattern: each processor sends all its messages to the opposite node(I,e, node 000 send messages to 111).
    - All the packets have to reach the upper stage in order to arrive to their destination.
    - The second one is that each processor node only sends messages to one destination.

Network Model

- Switch and link bandwidth has been assumed to be one flit per clock cycle
- Fly time through the link has been assumed to be 8 clock cycles
- Credits are used to implement the flow control mechanism