



**Università degli Studi di Catania**

Dipartimento di Ingegneria Informatica e delle Telecomunicazioni (DIIT)

---

# **Application Specific Routing Algorithms for Networks on Chip**

**Maurizio Palesi**

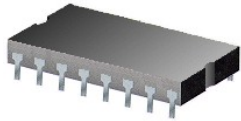
<http://www.diit.unict.it/~mpalesi>  
[mpalesi@diit.unict.it](mailto:mpalesi@diit.unict.it)

# Systems on Chip

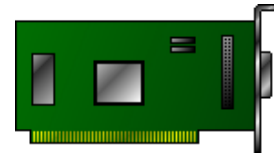
---

■ We are now in the era of block-based design

ASIC/ASSP  
Design

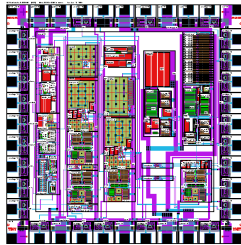


**Yesterday**  
Bus Standards,  
Predictable, Preverified

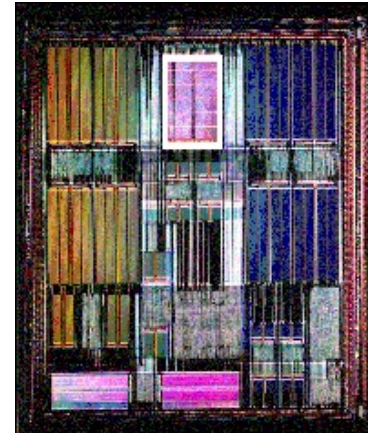


System-Board  
Integration

IP/Block  
Authoring



**Today**  
VSI Compatible Standards,  
Predictable, Preverified



System-Chip  
Integration

# Trends

---

- From *multi-core* to *many-core*

- From dual-, tri-, quad-, hexa-, octo-core chips to ones with **tens or even hundreds of core**

- Homogeneous vs. Heterogeneous diatribe

- Examples

- Teraflops Research Chip (Polaris), a 3.16 GHz, 80-core processor prototype

- NVIDIA's GPUs CUDA architecture (100s cores)

- Tiler TILE64, 64-core processor

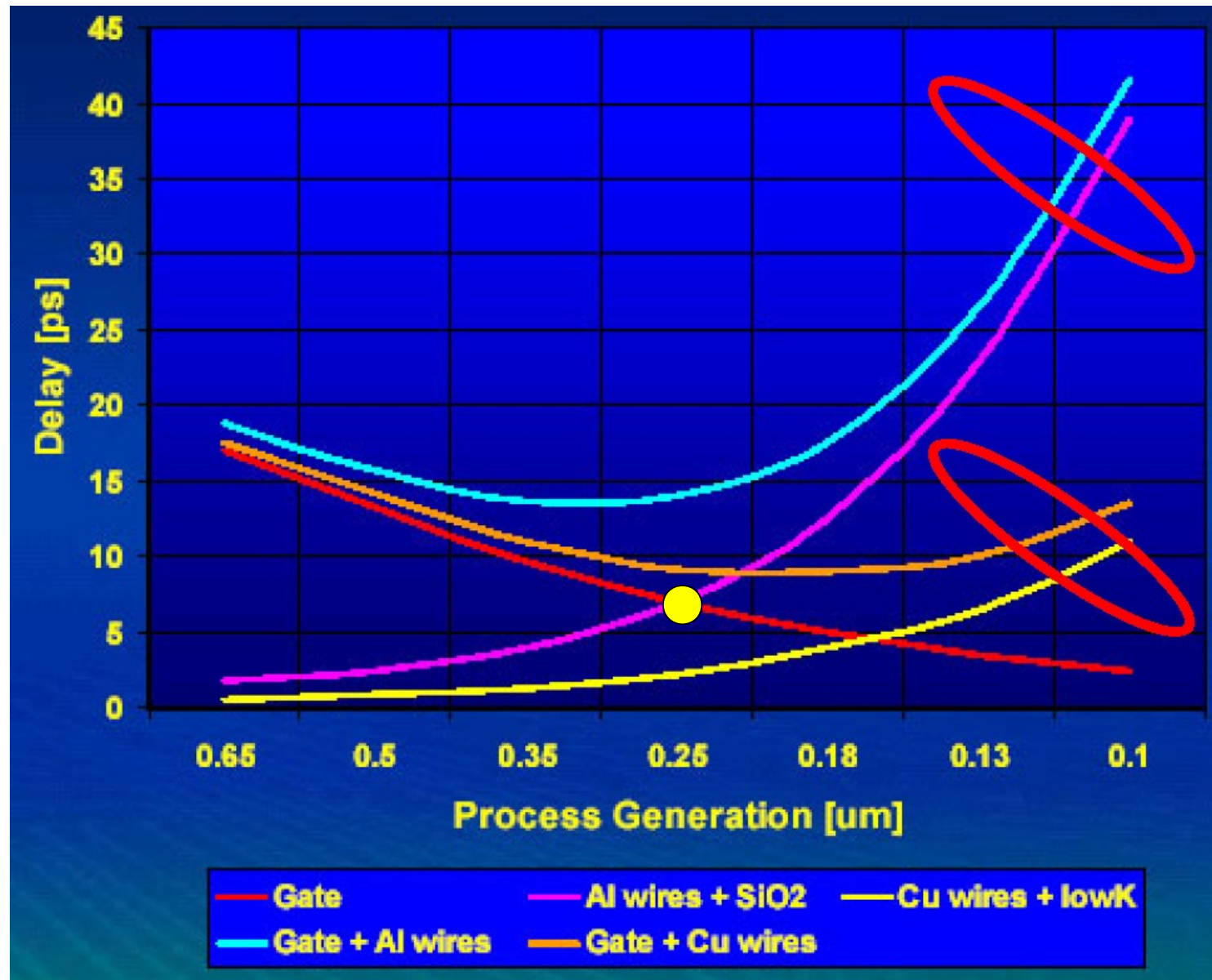
- picoChip PC200 series 200–300 cores per device for DSP & wireless

# Trends in SoCs

---

- Technology: no slow-down in sight!
  - Faster and smaller transistors: 90 → 65 → 45 → 32 nm
  - ...but slower wires, lower voltage, more noise!
    - ✓ 80% or more of the delay of critical paths will be due to interconnects
- Design complexity: from 2 to 10 to 100 cores!
  - Design reuse is essential
  - ...but differentiation/innovation is key for winning on the market!
- Performance and power: GOPS for MWs!
  - Performance requirements keep going up
  - ...but power budgets do not!

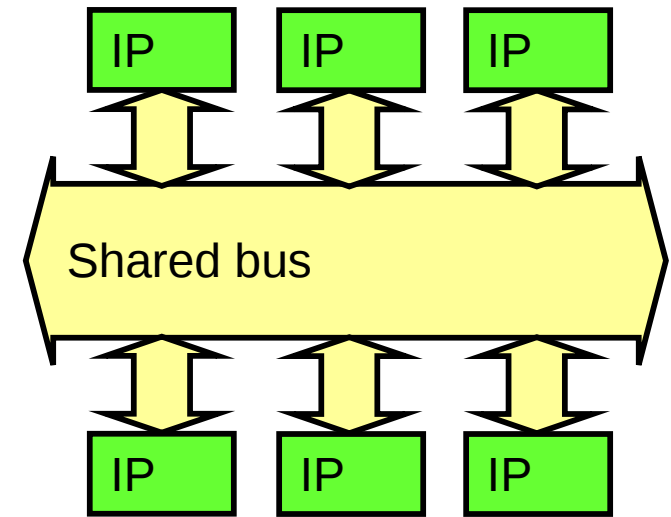
# The Deep Submicron Effects



# Communication Architectures

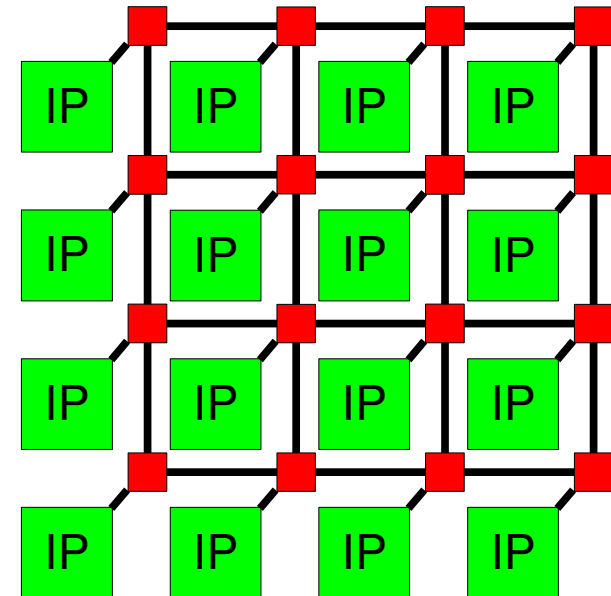
## ■ Shared bus

- Low area
- Poor scalability
- High energy consumption

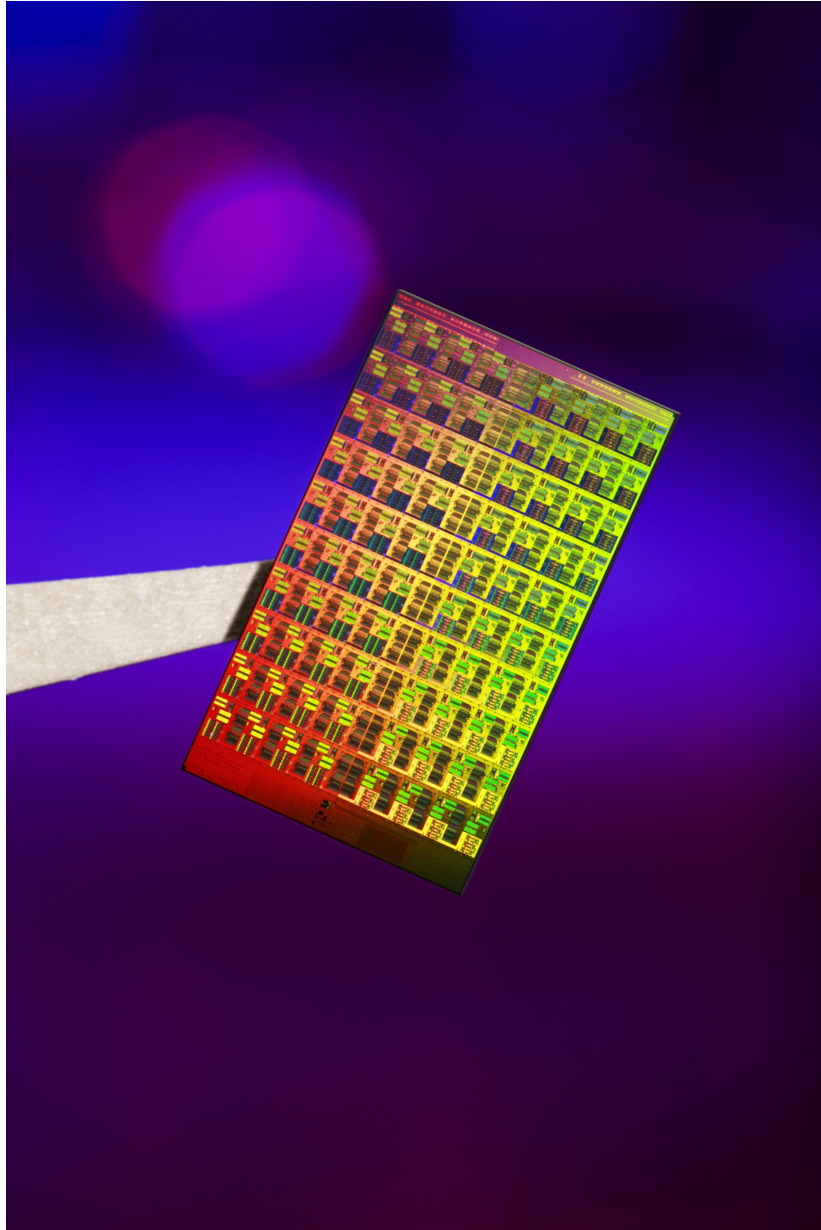


## ■ Network-on-Chip

- Scalability and modularity
- Low energy consumption
- Increase of design complexity

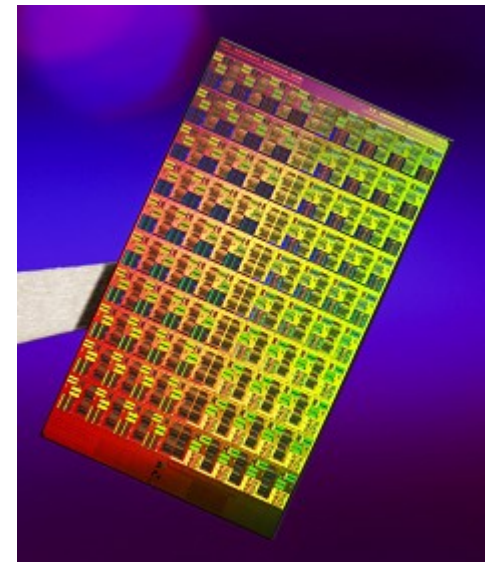
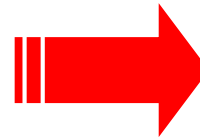


# Intel's Teraflops



- 100 Million transistors
- 80 cores, 160 FP engines
- Teraflops perf. @ 62 Watts
- On-die mesh network
- Power aware design

# Terascale vs. ASCI Red



- **ASCI Red** the first computer to reach a Teraflops of processing (1996)
  - 10,000 Pentium @ 200 MHz
  - 500kW of power (+500kW to keep the room cool)
- **Terascale** the first on-chip solution reaching Teraflop of processing (2006)
  - 80 core chip (VLIW based)
  - 3.16-5.7 GHz
  - 62W-256W

# Outline

---

- Application Specific Routing Algorithms
- Concurrent Mapping and Routing
- Dealing with Manufacturing Defects
- Encoding Scheme for Low Power

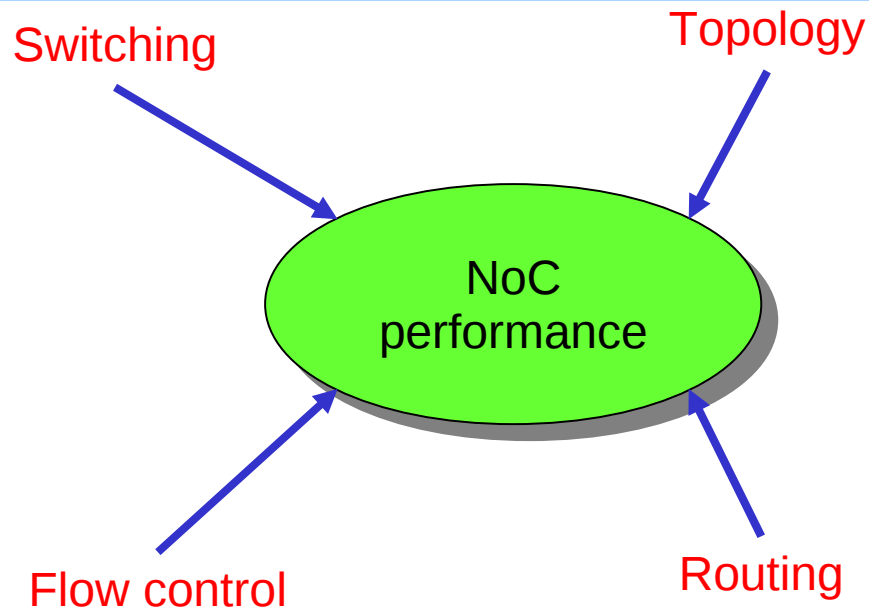
# Outline

---

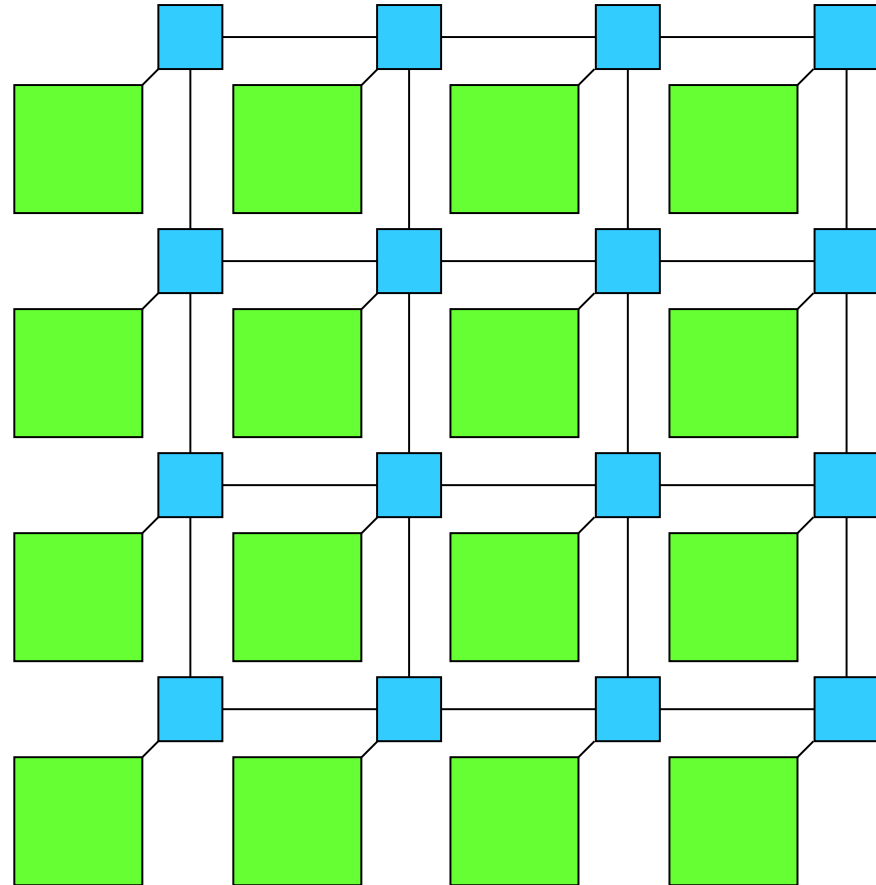
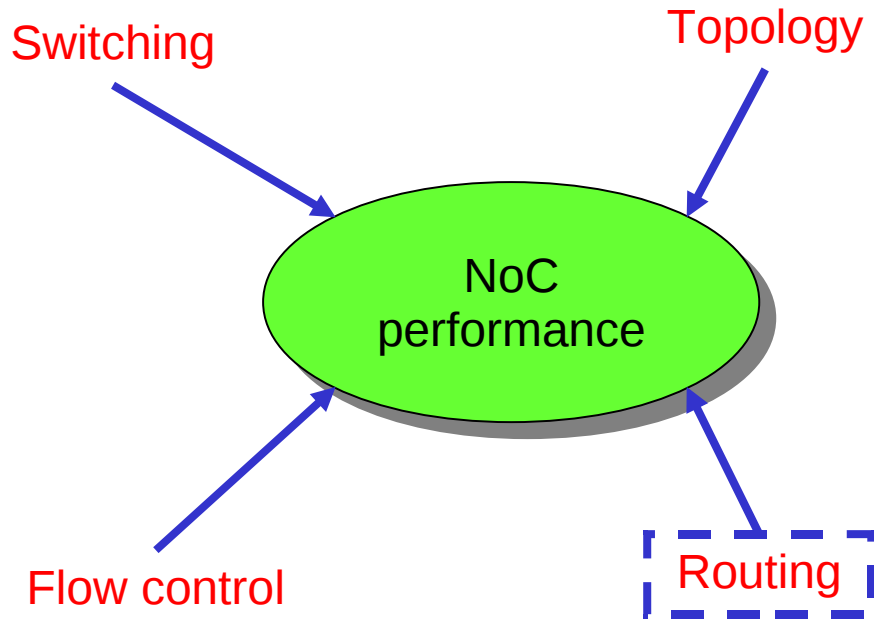
- Application Specific Routing Algorithms
- Concurrent Mapping and Routing
- Dealing with Manufacturing Defects
- Encoding Scheme for Low Power

# Routing Algorithms

---



# Routing Algorithms

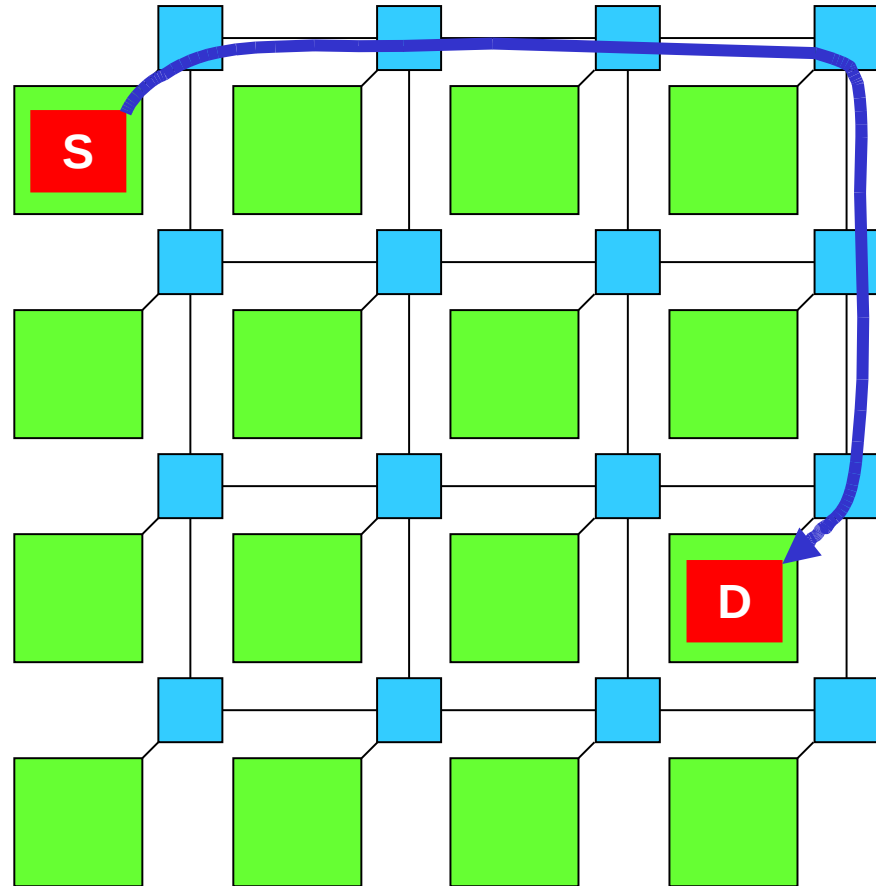
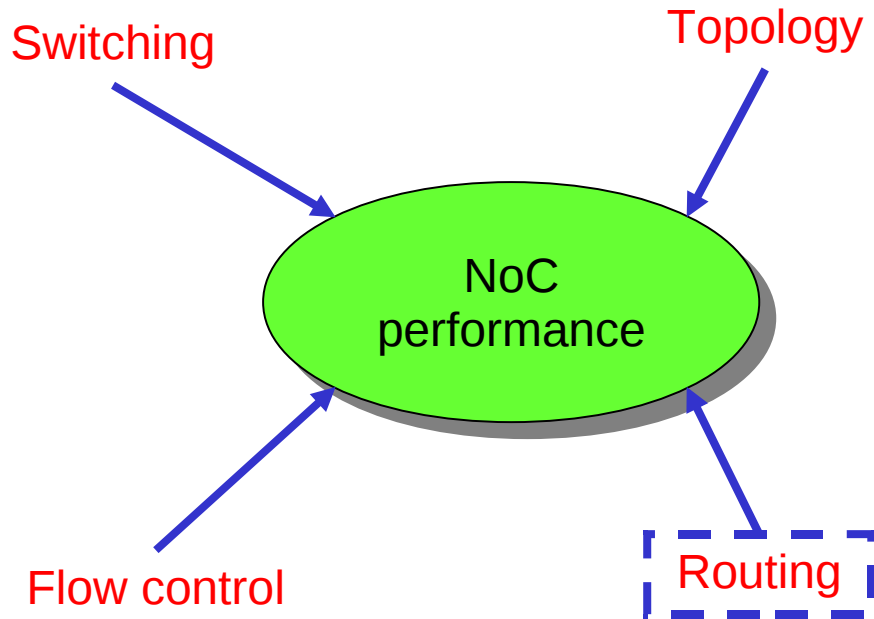


- Routing determines the path selected by a packet to reach its destination

→ Deterministic

→ Adaptive

# Routing Algorithms



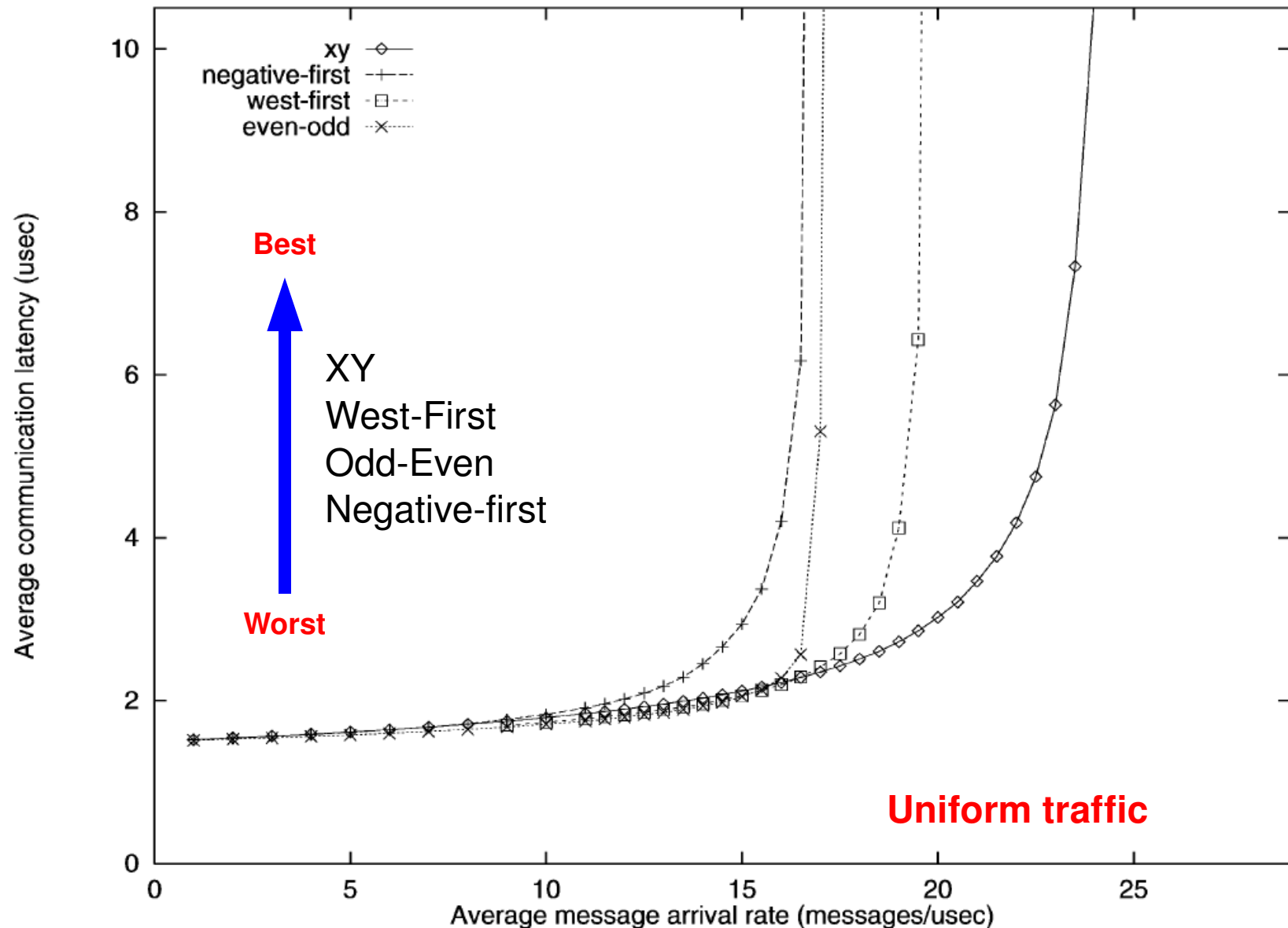
- Routing determines the path selected by a packet to reach its destination

→ **Deterministic**

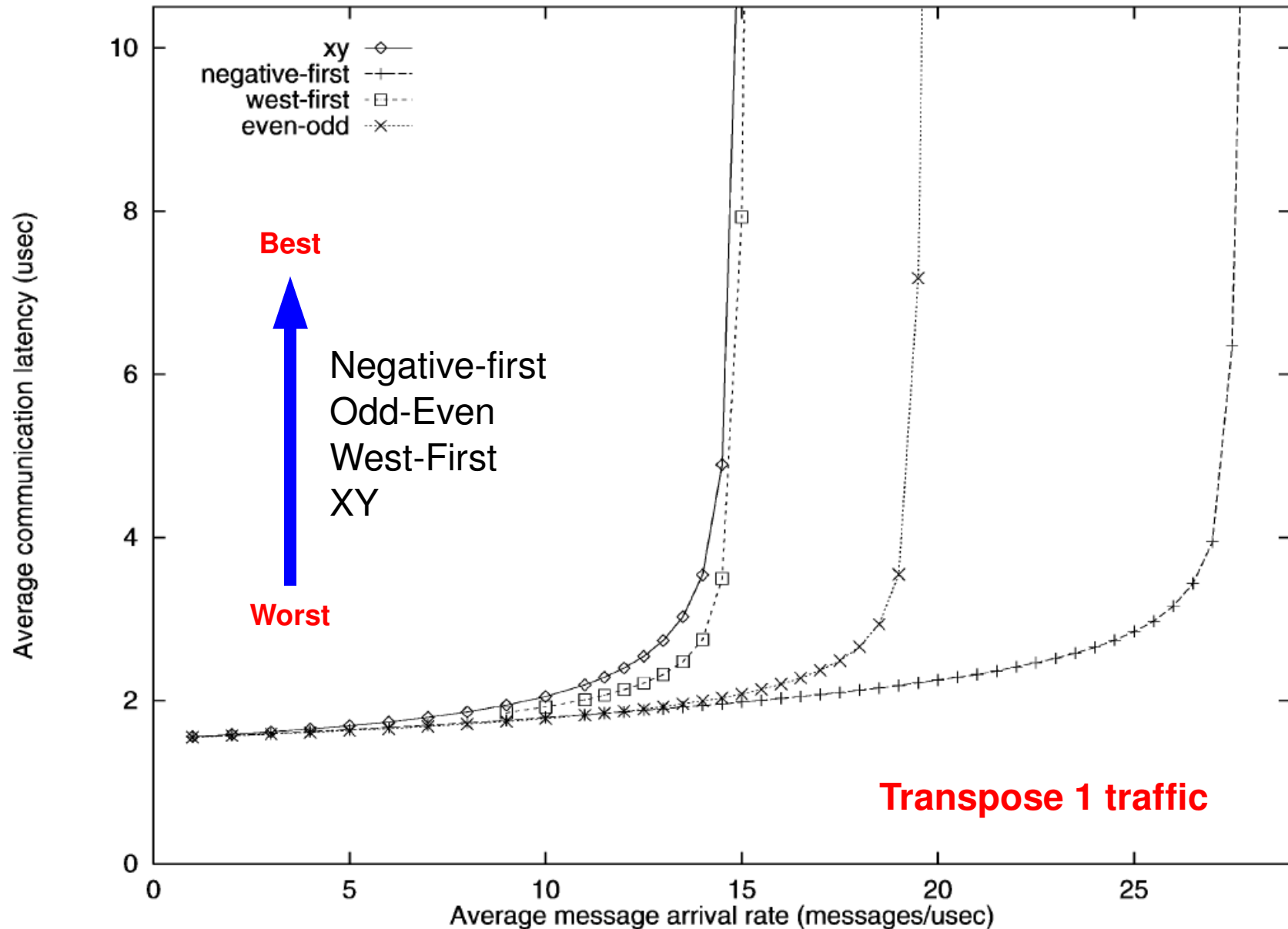
→ Adaptive



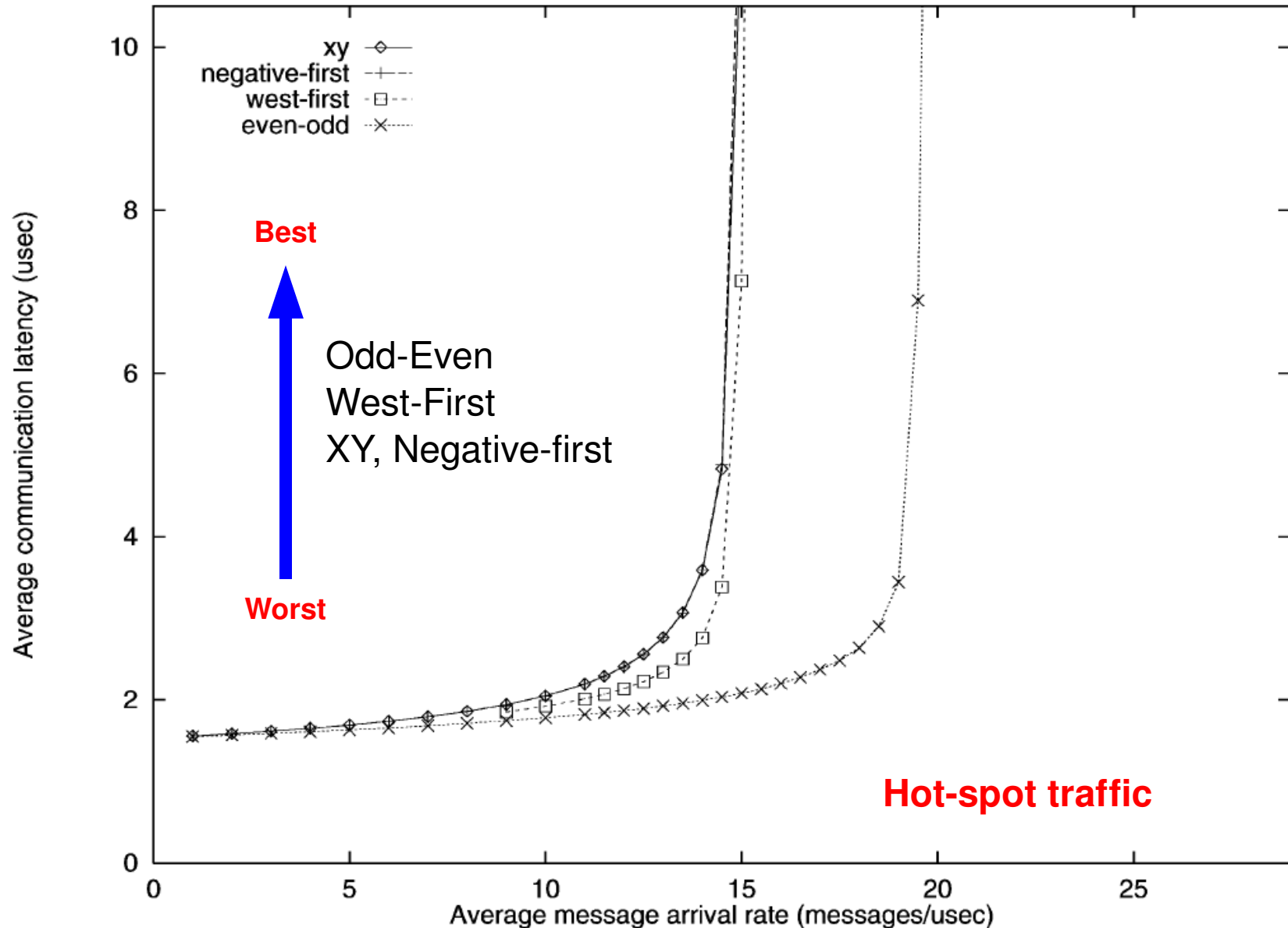
# Performance of Routing Algorithms



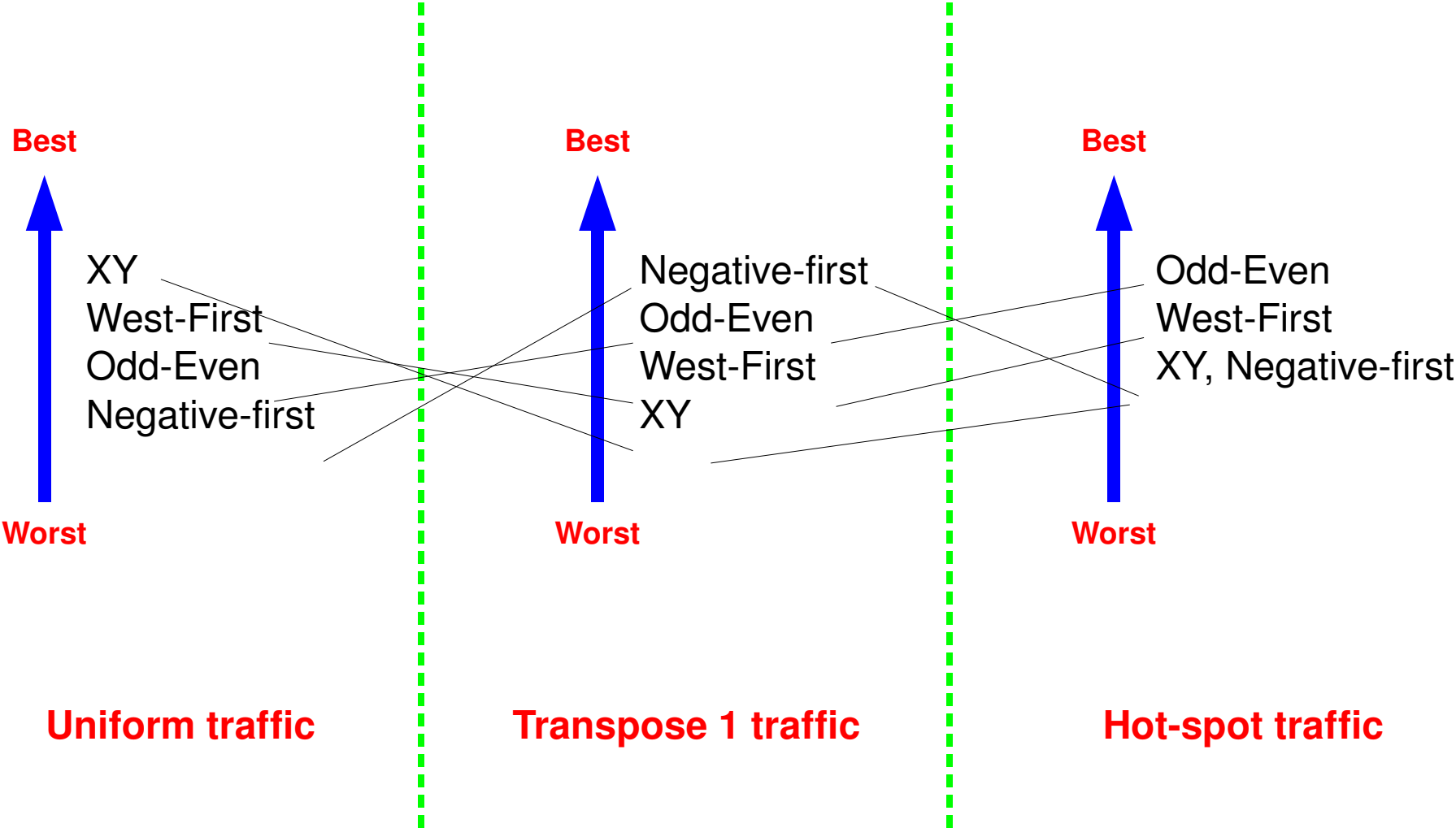
# Performance of Routing Algorithms



# Performance of Routing Algorithms



# No Winner Routing Algorithm



# Application Specific Routing Algorithm

## ■ Information about

→ Tasks which communicate and tasks which do never communicate

✓ After task mapping → Information about network nodes which communicate

→ Concurrent/non concurrent communications

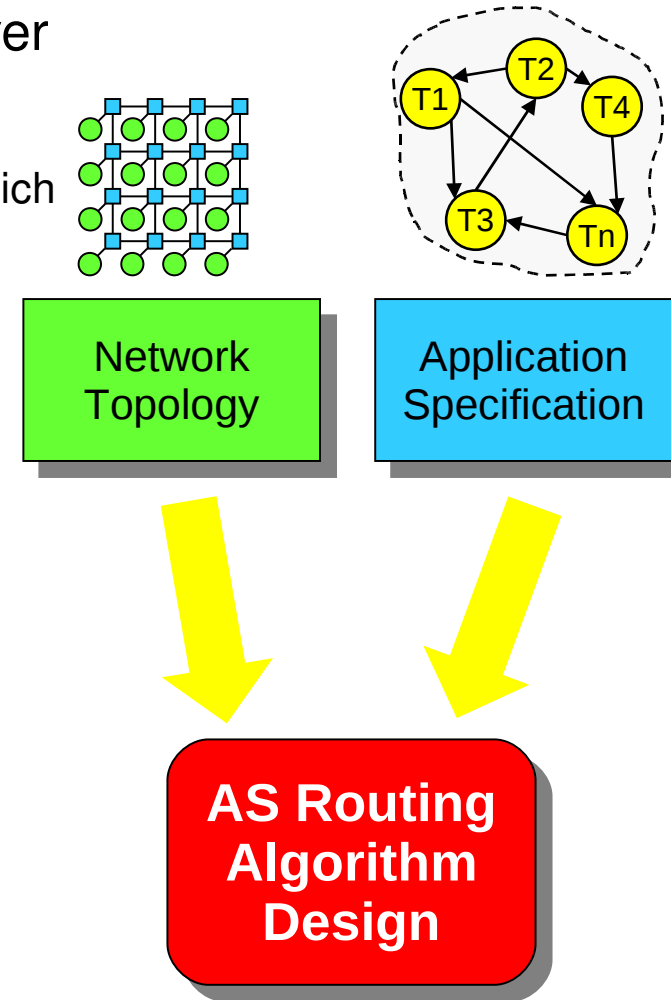
→ Communications bandwidth requirements

## ■ Many opportunities

→ Improving performance (e.g., maximize routing adaptivity)

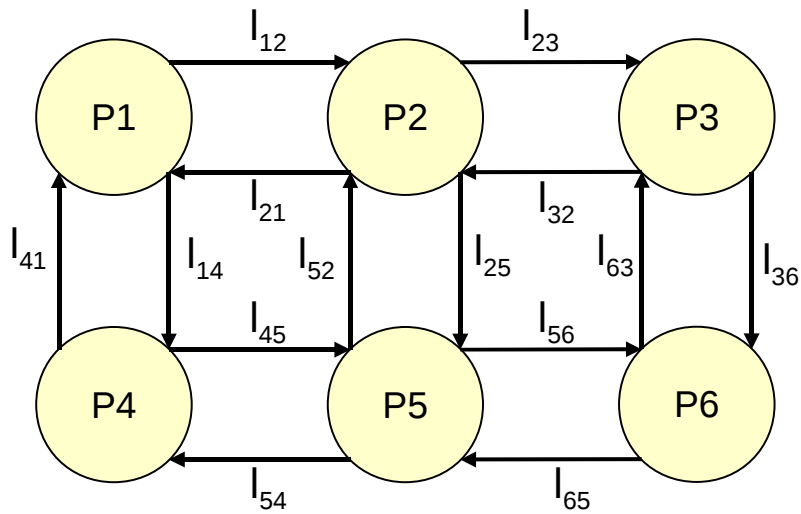
→ Simplify the estimation/control of congestion

→ Design more effective selection policies

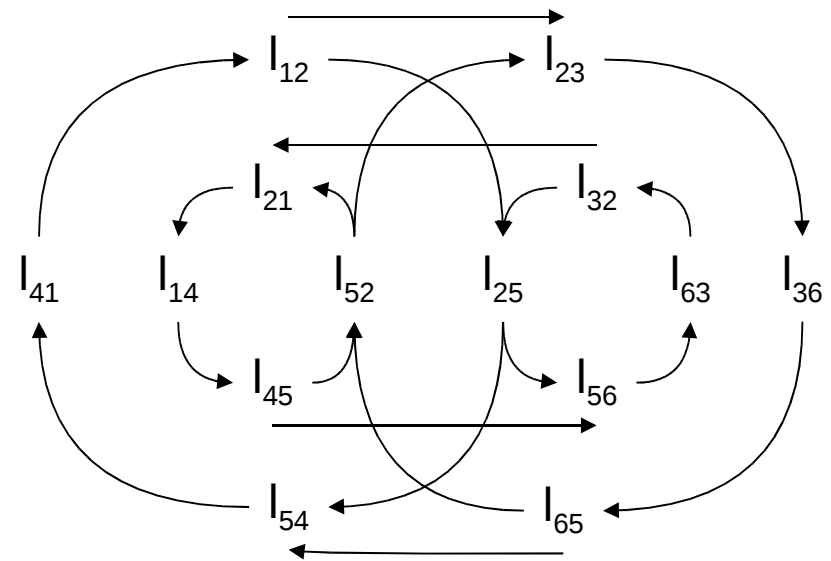


# APSRA Example

Topology Graph

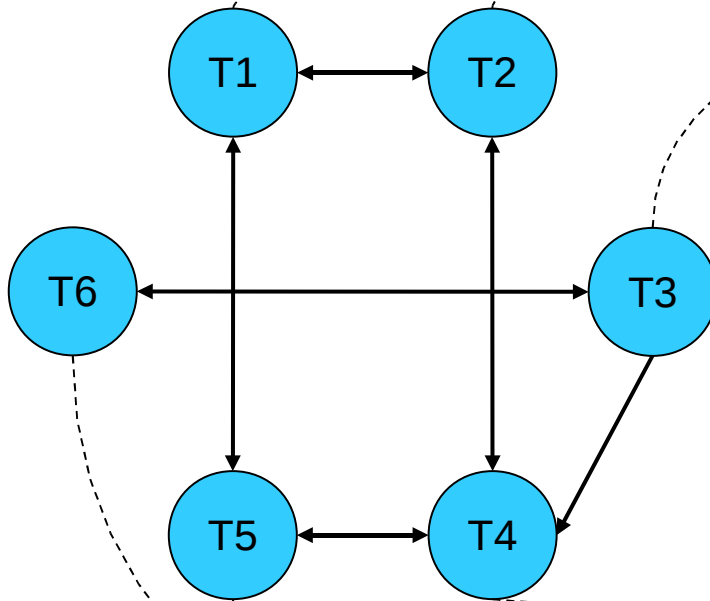


Channel Dependency Graph

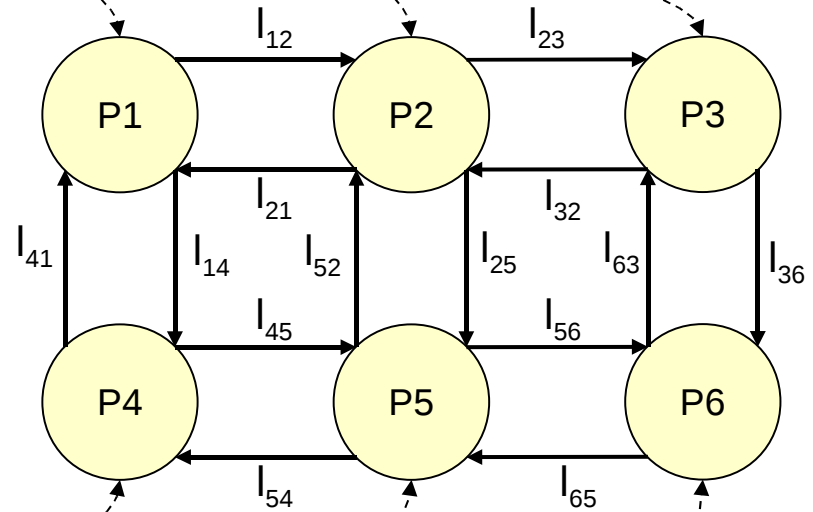


# APSRA Example (cnt'd)

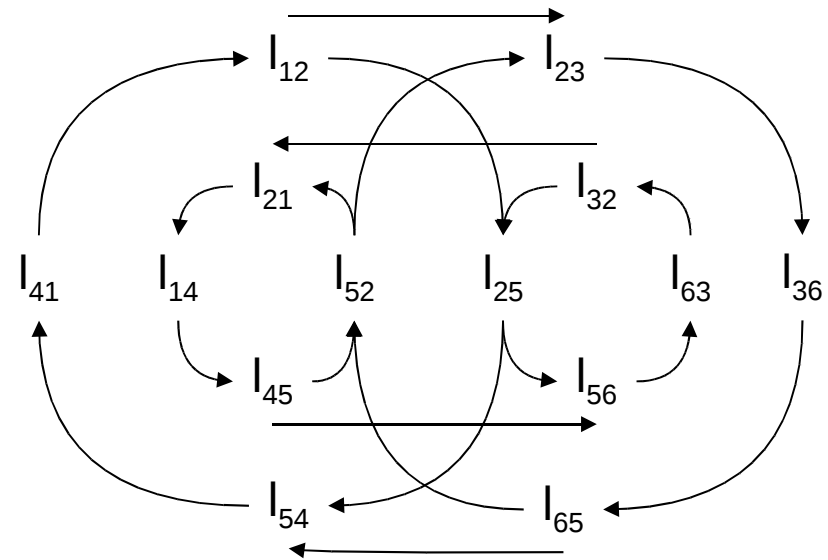
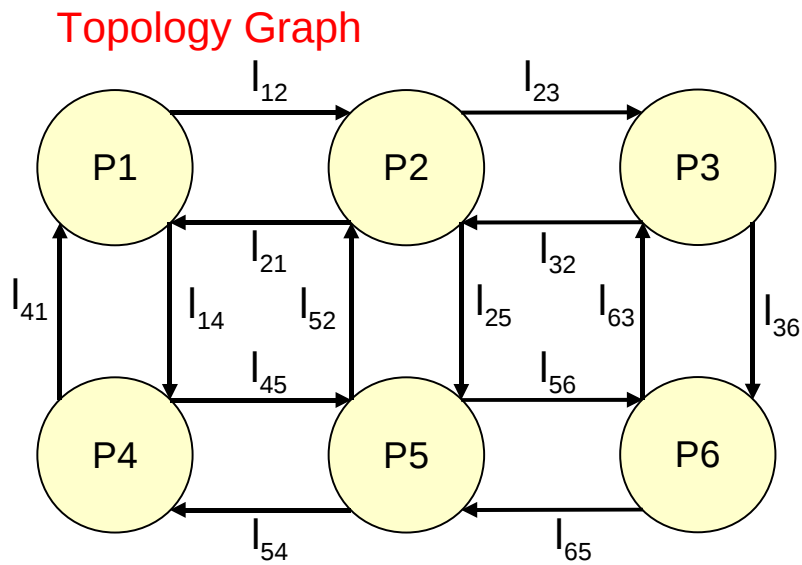
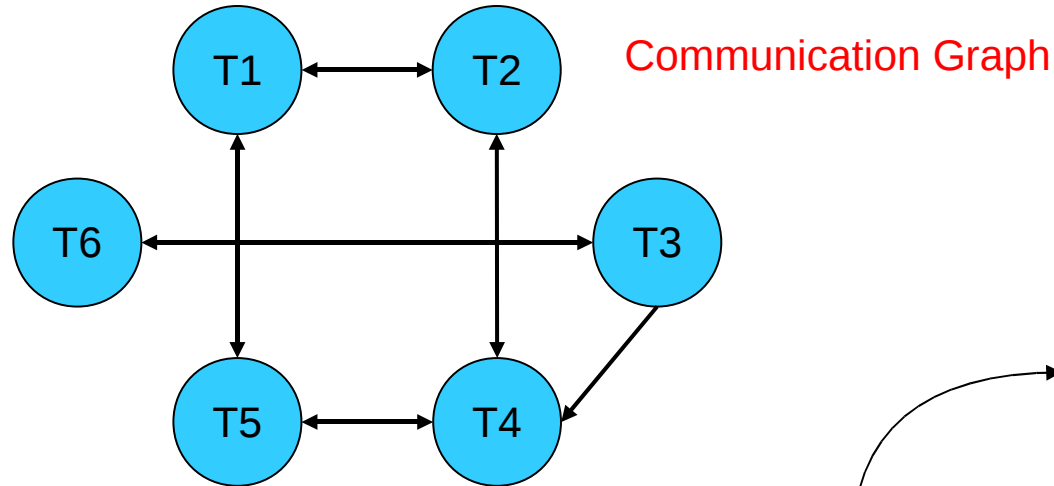
Communication Graph



Topology Graph

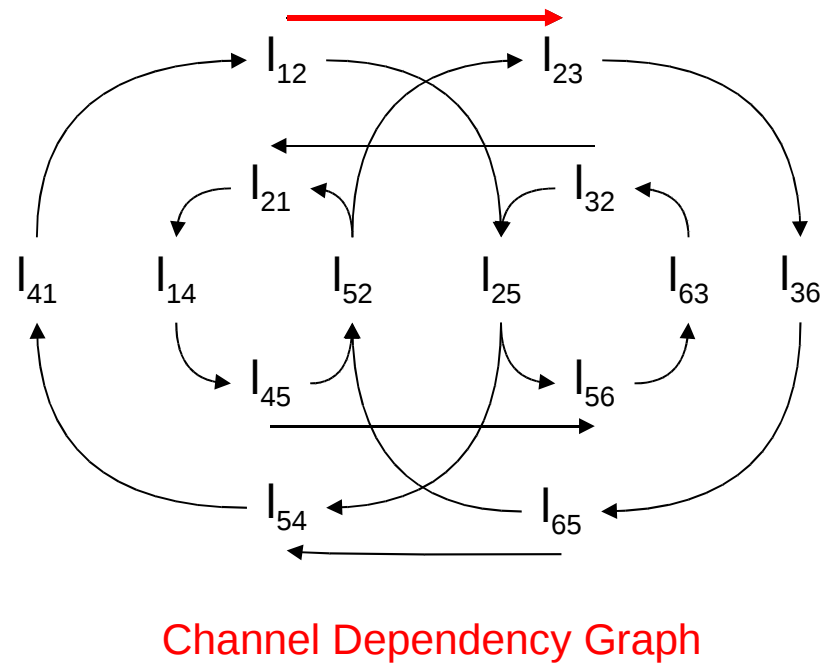
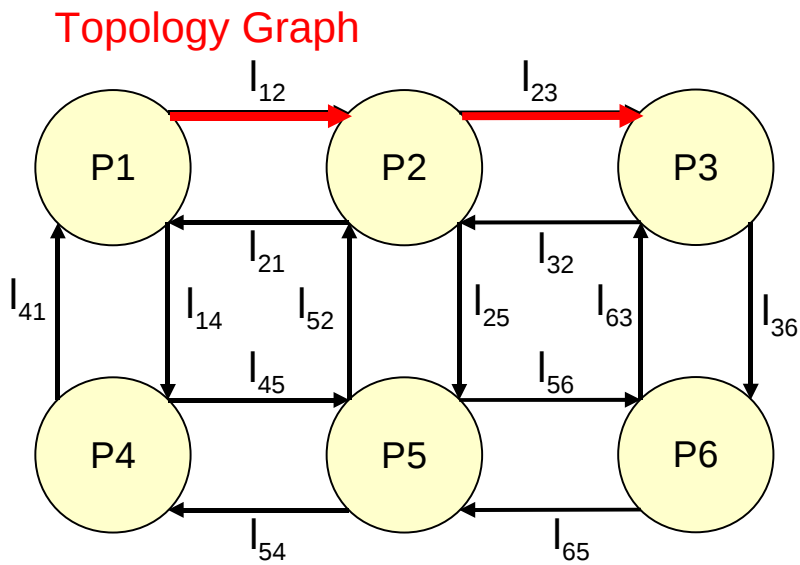
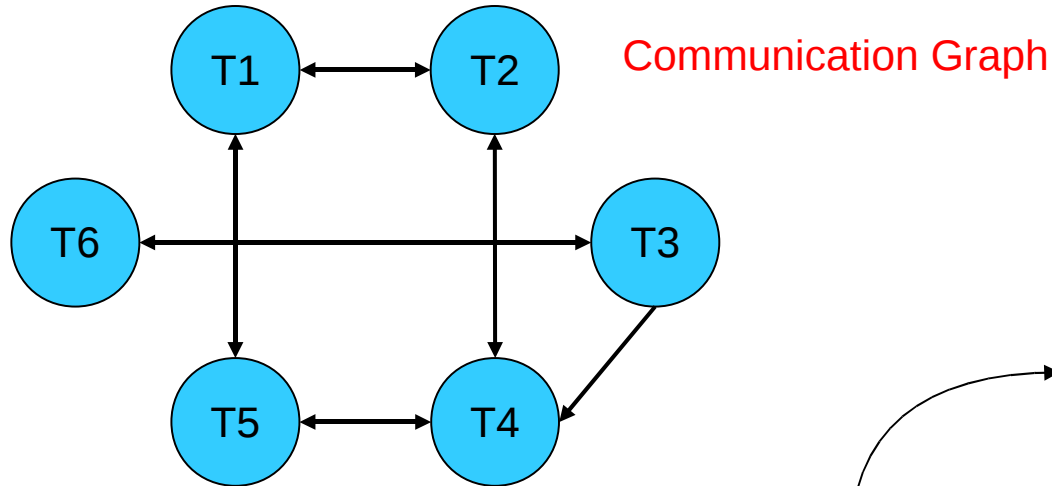


# APSRA Example (cnt'd)

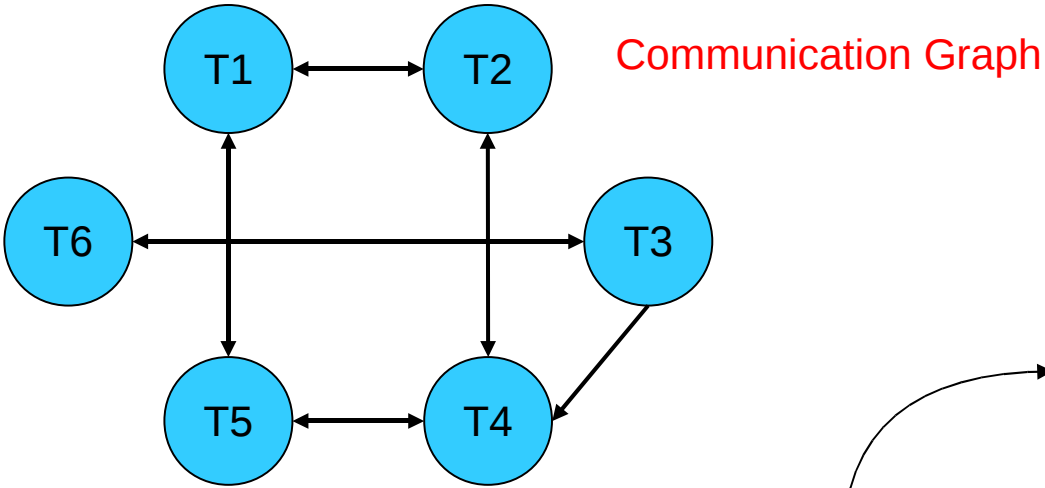


Channel Dependency Graph

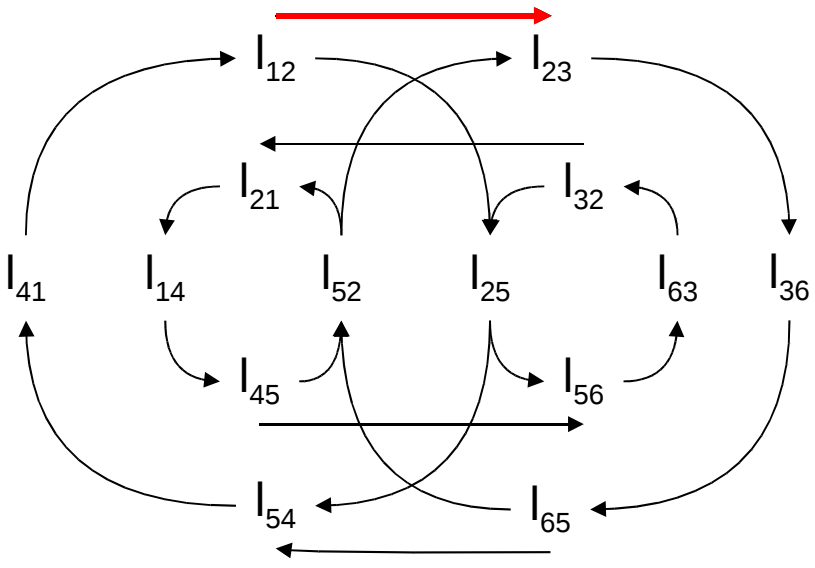
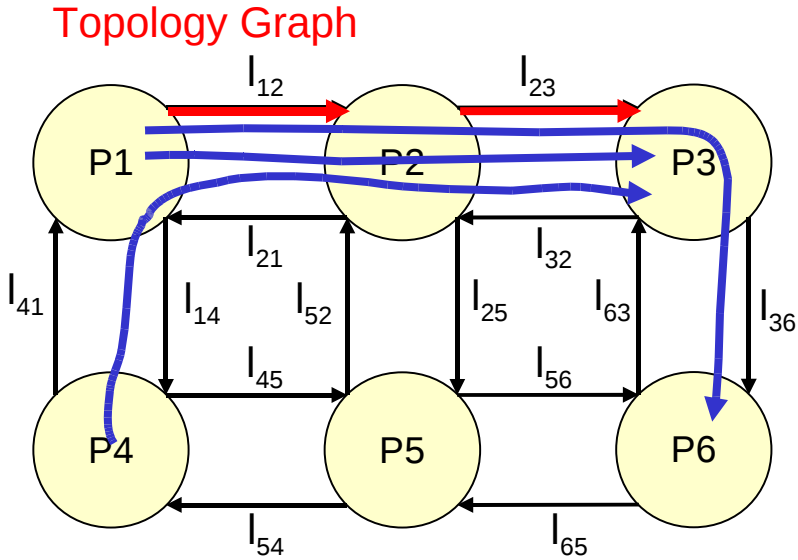
# APSRA Example (cnt'd)



# APSRA Example (cnt'd)

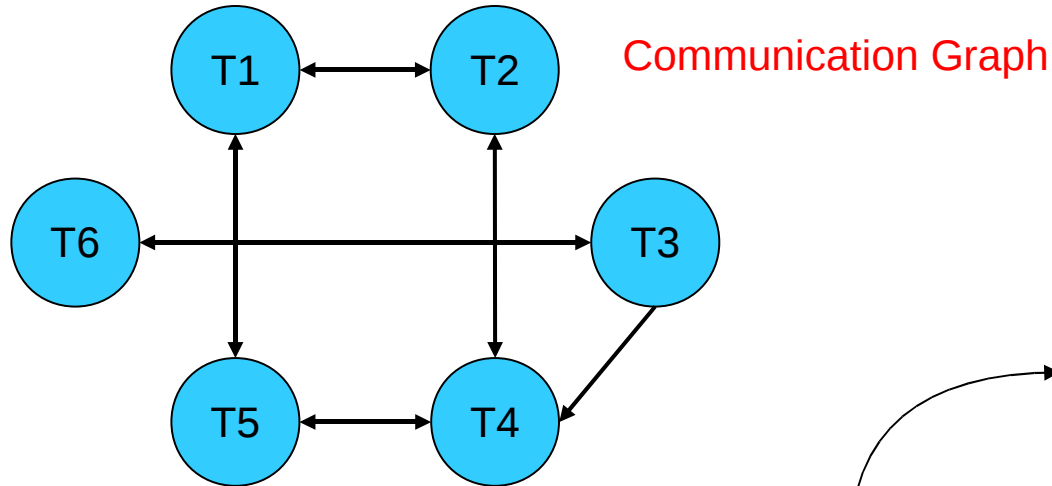


- T1 → T3
- T4 → T3
- T1 → T6

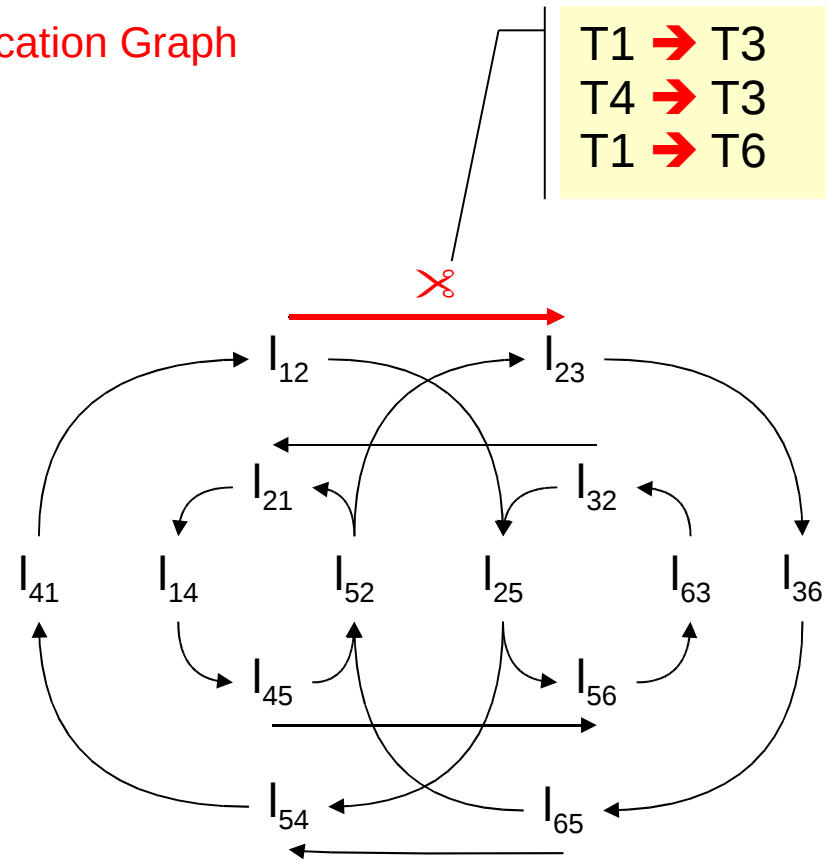
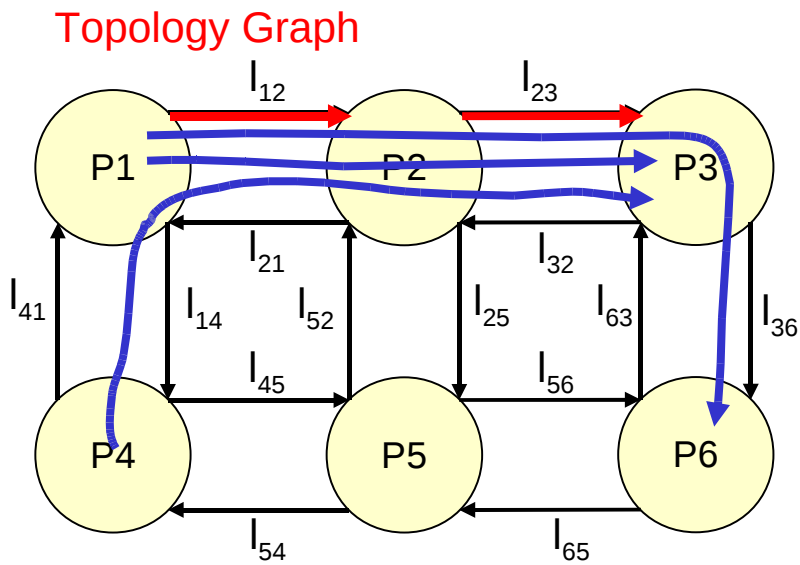


Channel Dependency Graph

# APSRA Example (cnt'd)

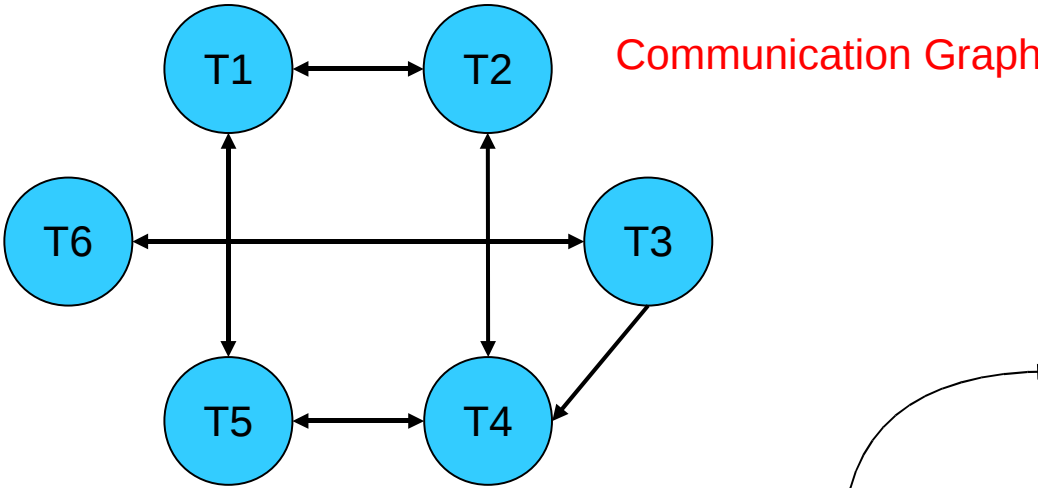


T1 → T3  
T4 → T3  
T1 → T6

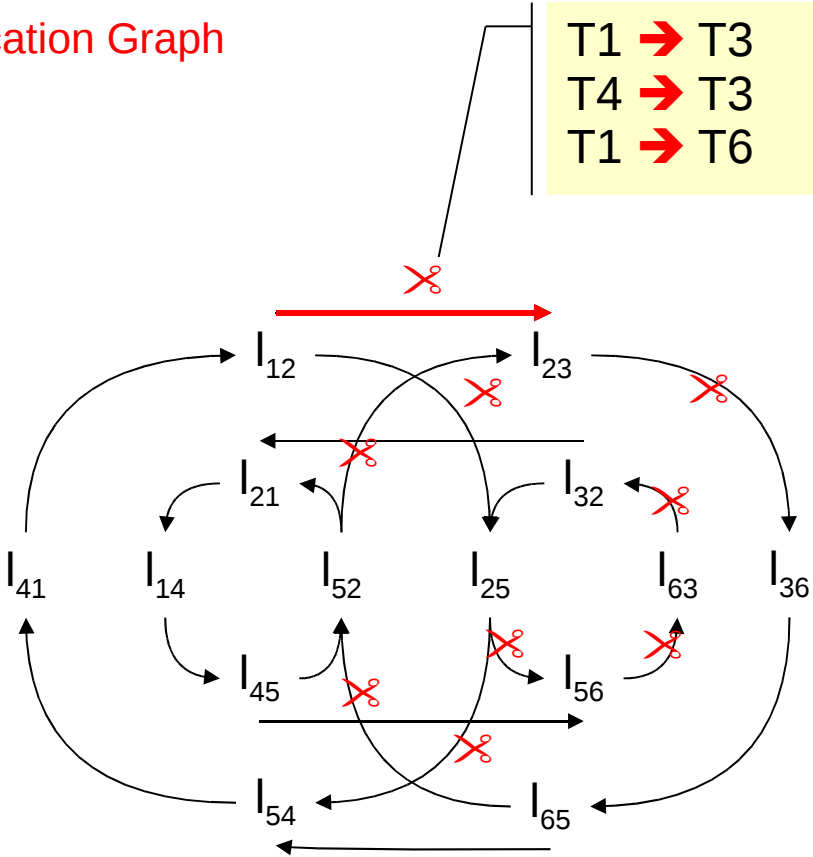
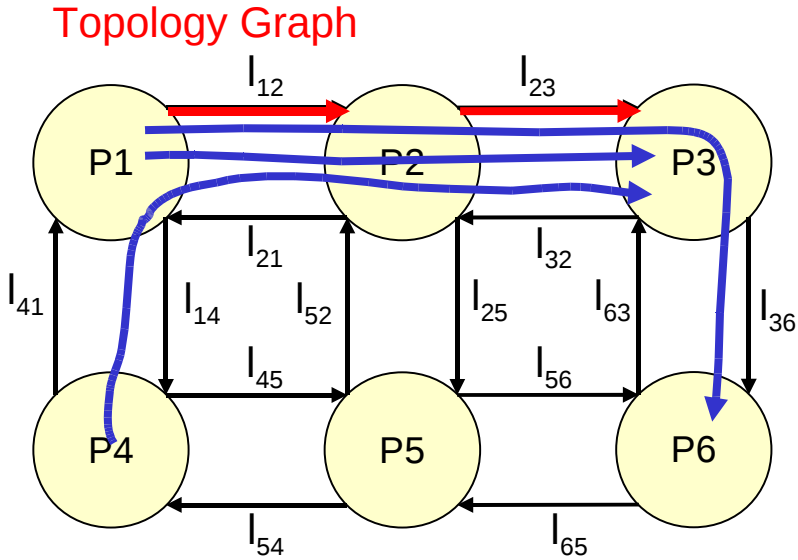


Channel Dependency Graph

# APSRA Example (cnt'd)

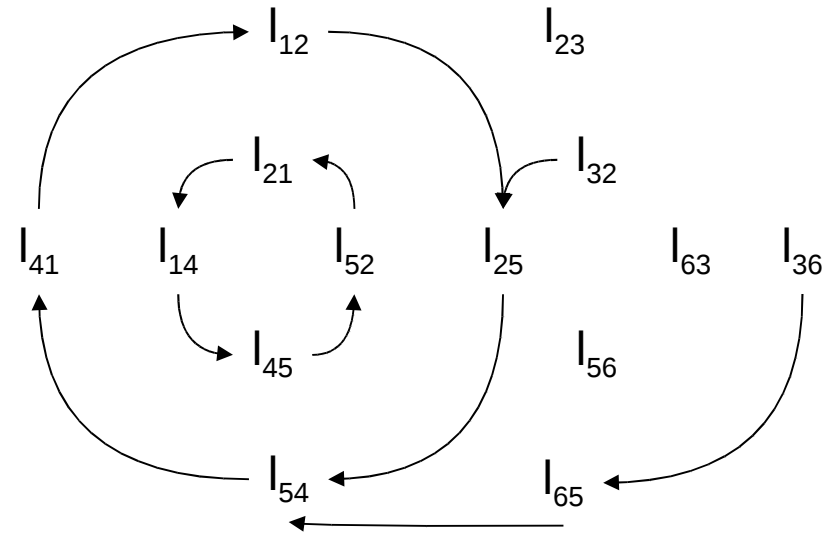
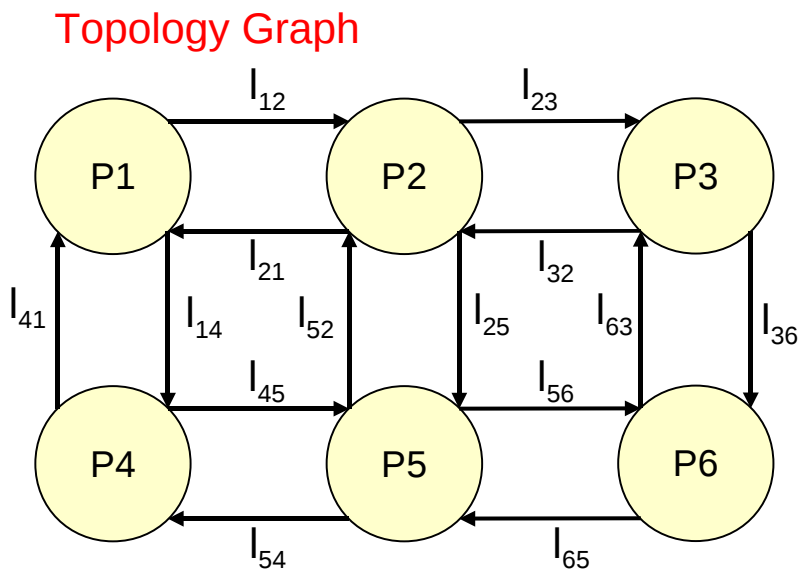
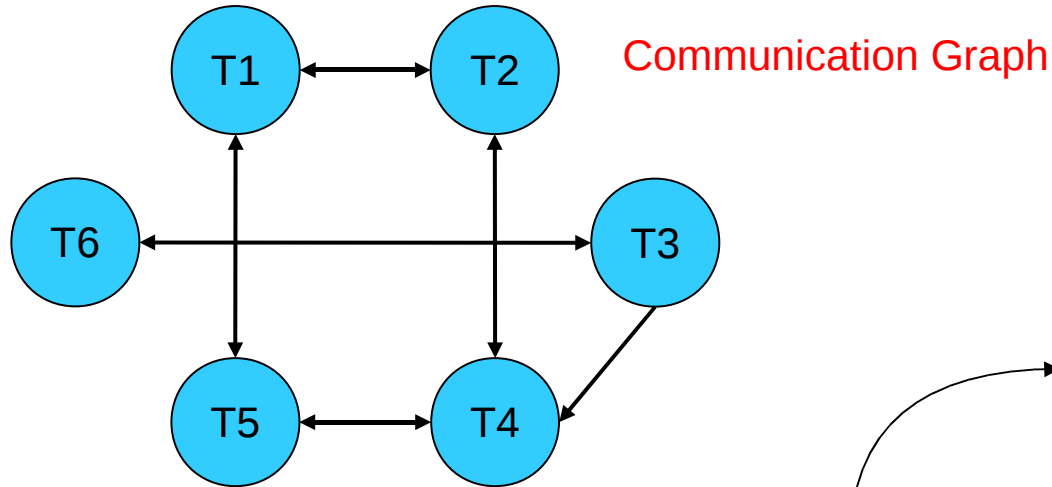


- T1 → T3
- T4 → T3
- T1 → T6



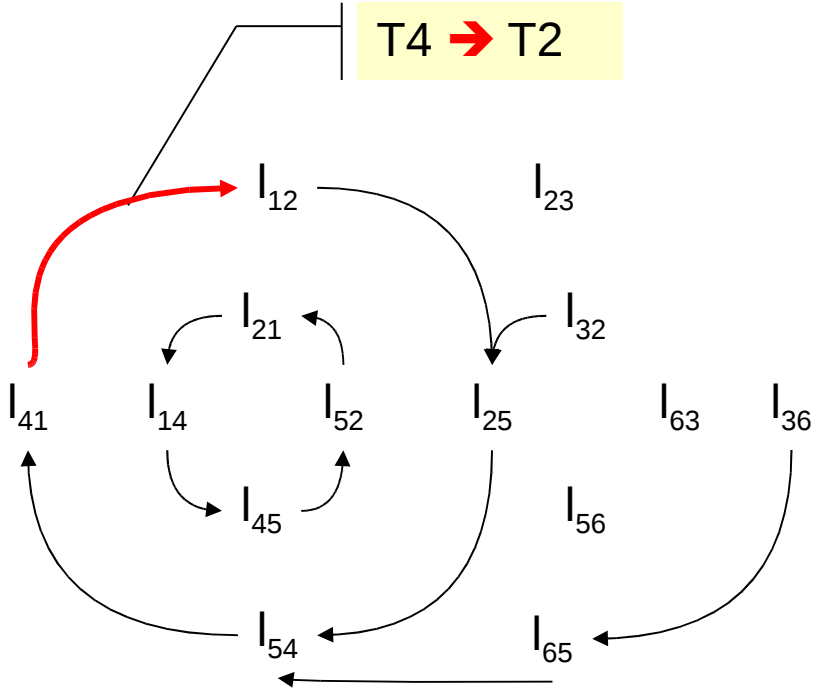
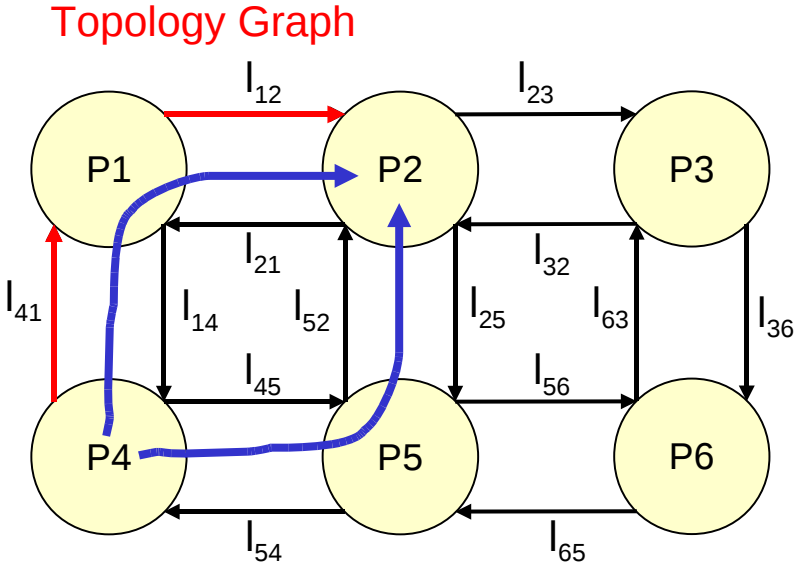
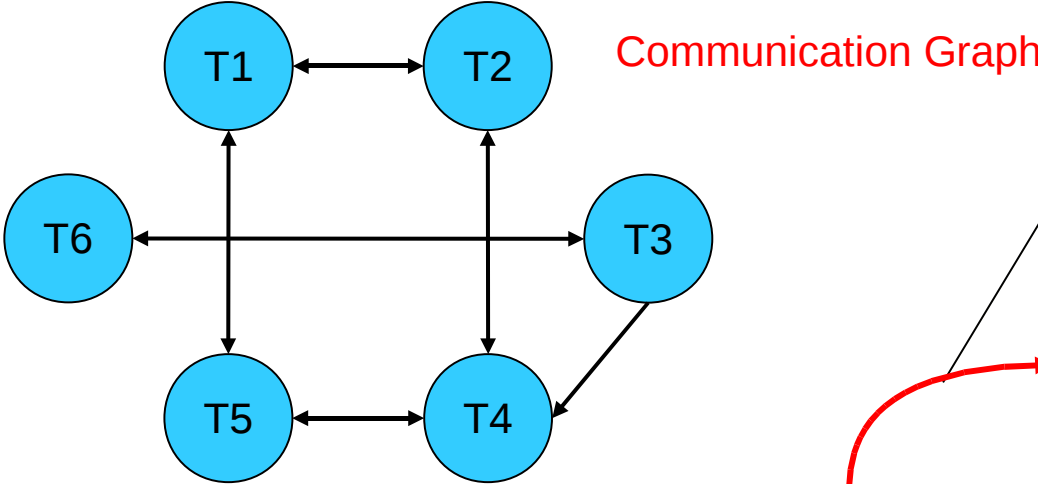
Channel Dependency Graph

# APSRA Example (cnt'd)



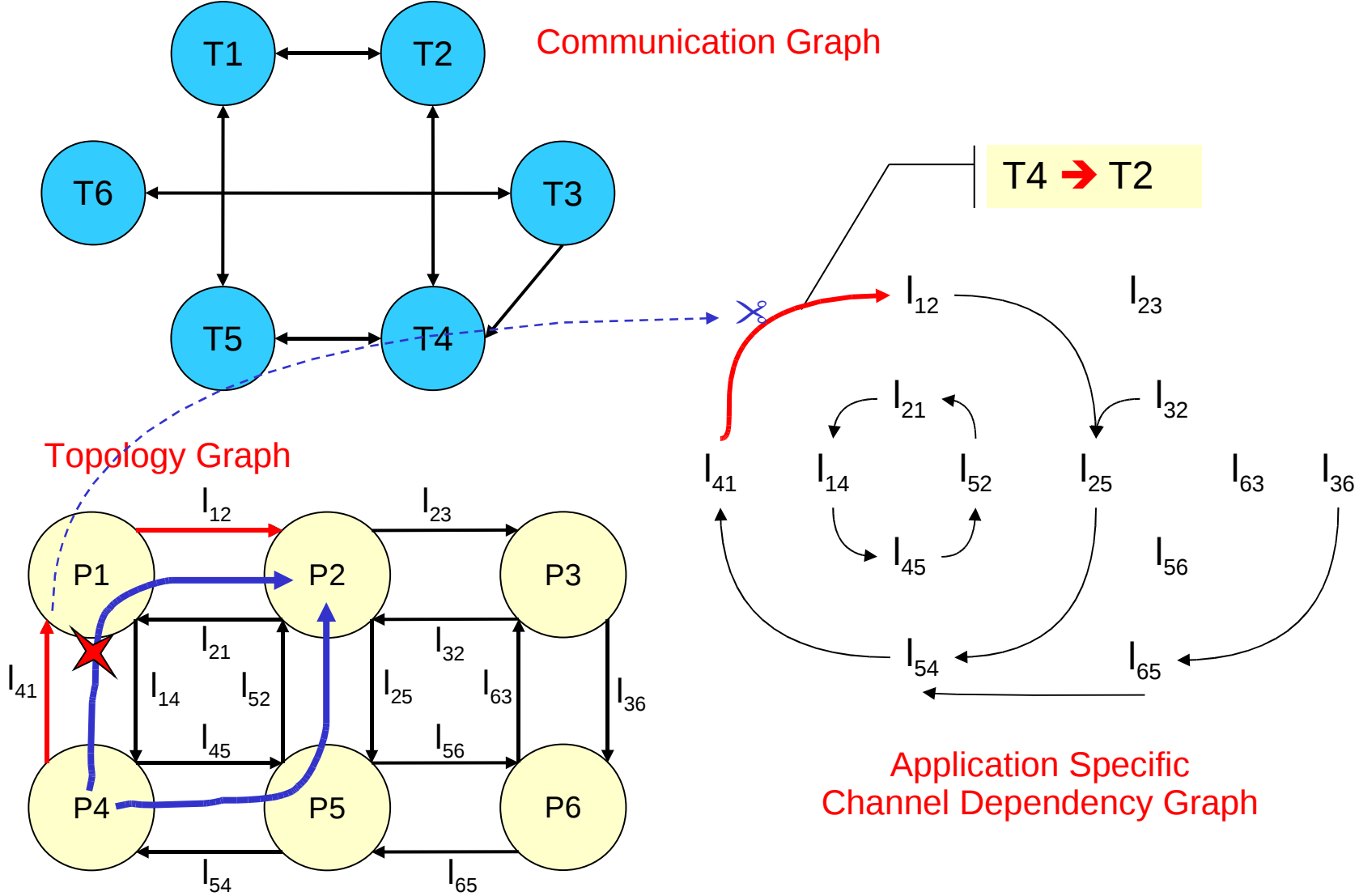
Application Specific Channel Dependency Graph

# APSRA Example (cnt'd)

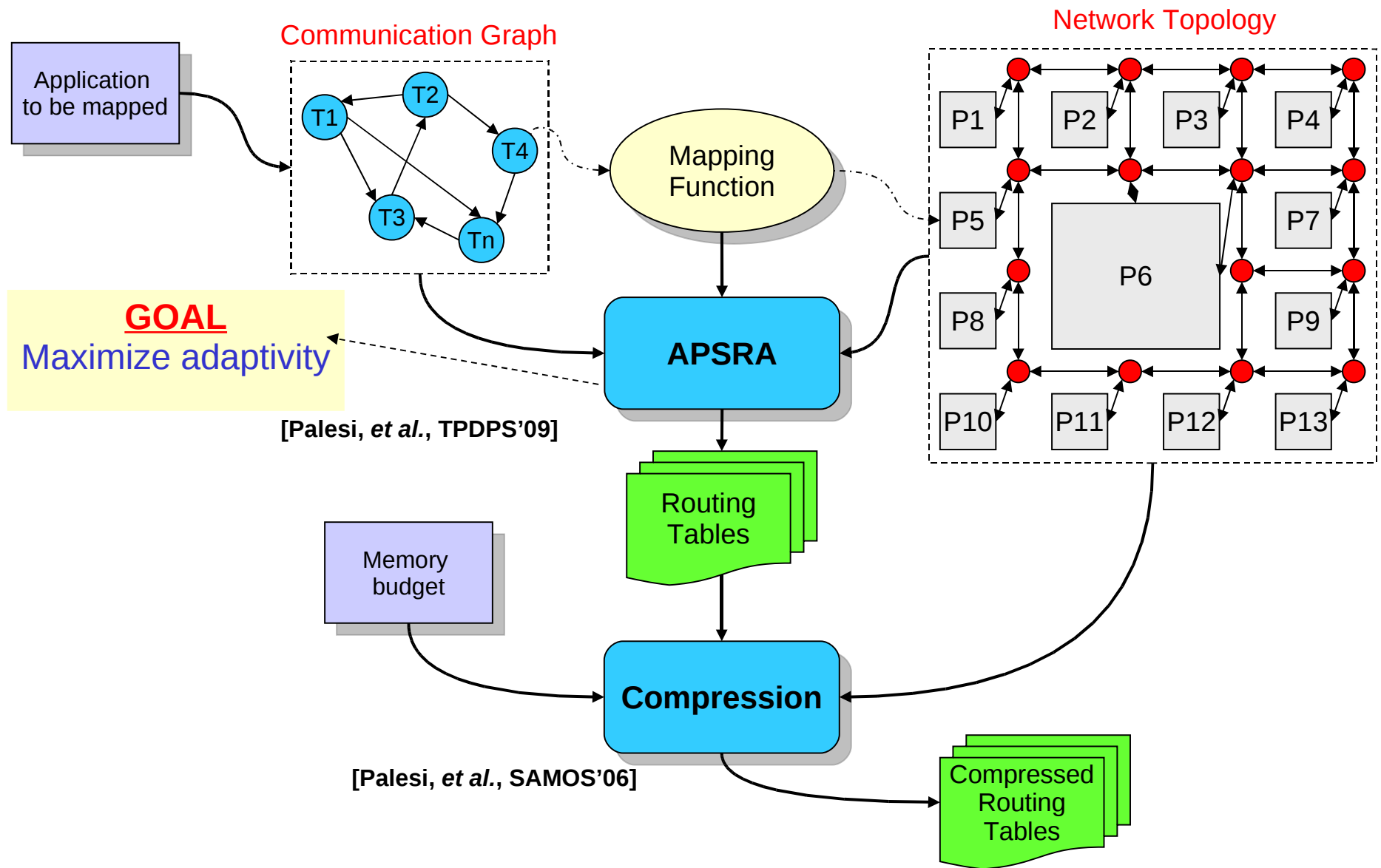


Application Specific Channel Dependency Graph

# APSRA Example (cnt'd)

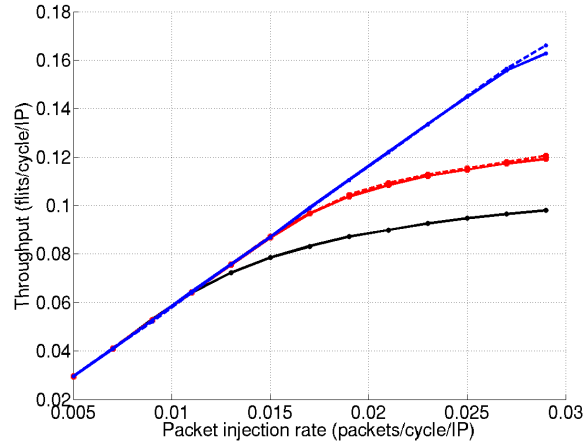
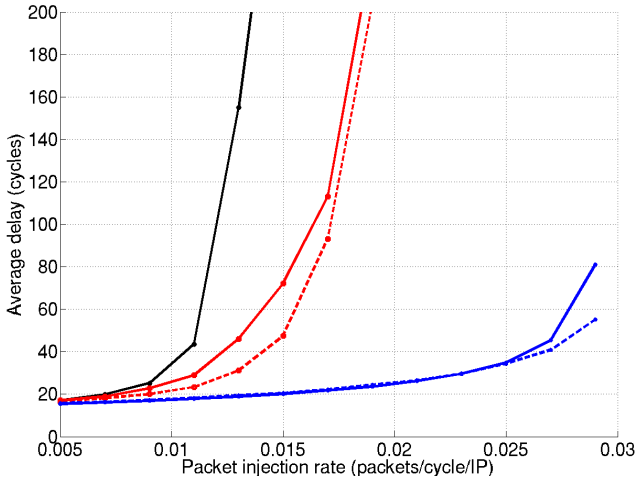


# APPSRA Methodology



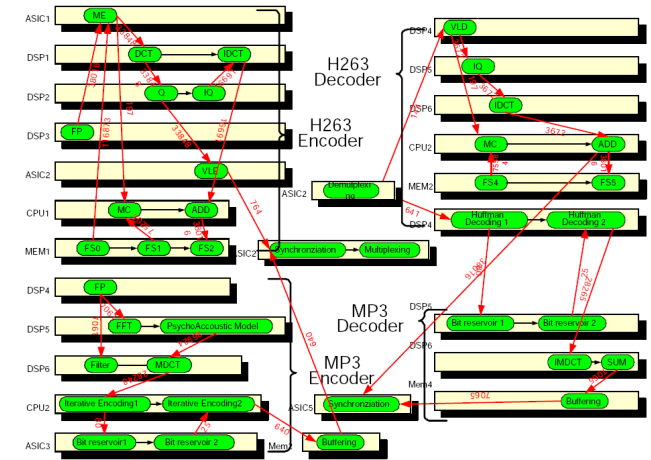
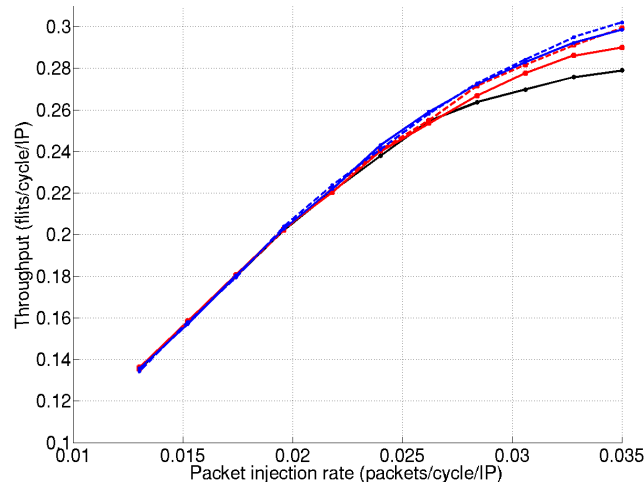
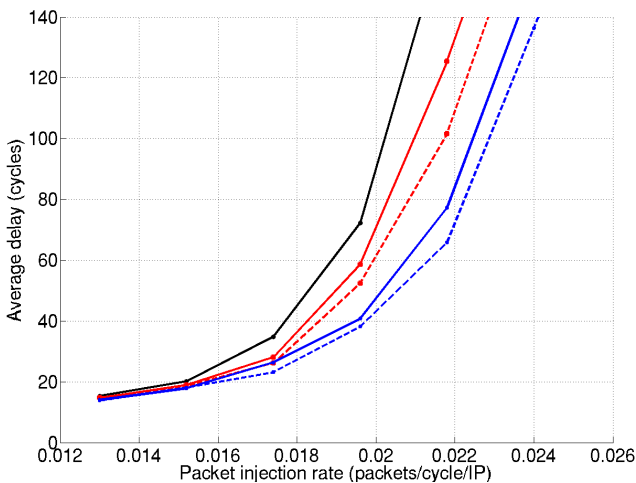
# APSRA Performance (1/2)

## Transpose 1

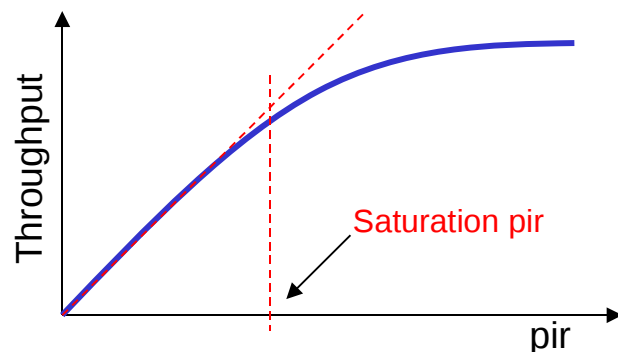


- XY
- Odd-Even (sel=random)
- APSRA (sel=random)
- - - Odd-Even (sel=buffer level)
- - - APSRA (sel=buffer level)

## MMS



# APSRA Performance (2/2)



Traffic scenario	Max. pir (packets/cycle/IP)			APSRA improvement	
	XY	OE	APSRA	vs. XY	vs. OE
Random	0.012	0.011	0.012	0.0%	14.3%
Locality	0.019	0.020	0.021	10.5%	5.0%
Transpose 1	0.011	0.015	0.027	145.5%	80.0%
Transpose 2	0.011	0.016	0.027	145.5%	68.8%
Hotspot-4c	0.003	0.004	0.004	13.6%	7.1%
Hotspot-4tr	0.003	0.003	0.004	29.6%	12.9%
Hotspot-8r	0.004	0.006	0.007	71.8%	13.6%
Mms	0.017	0.017	0.020	12.6%	12.6%
<b>Average improvement</b>				<b>53.6%</b>	<b>26.8%</b>

# Outline

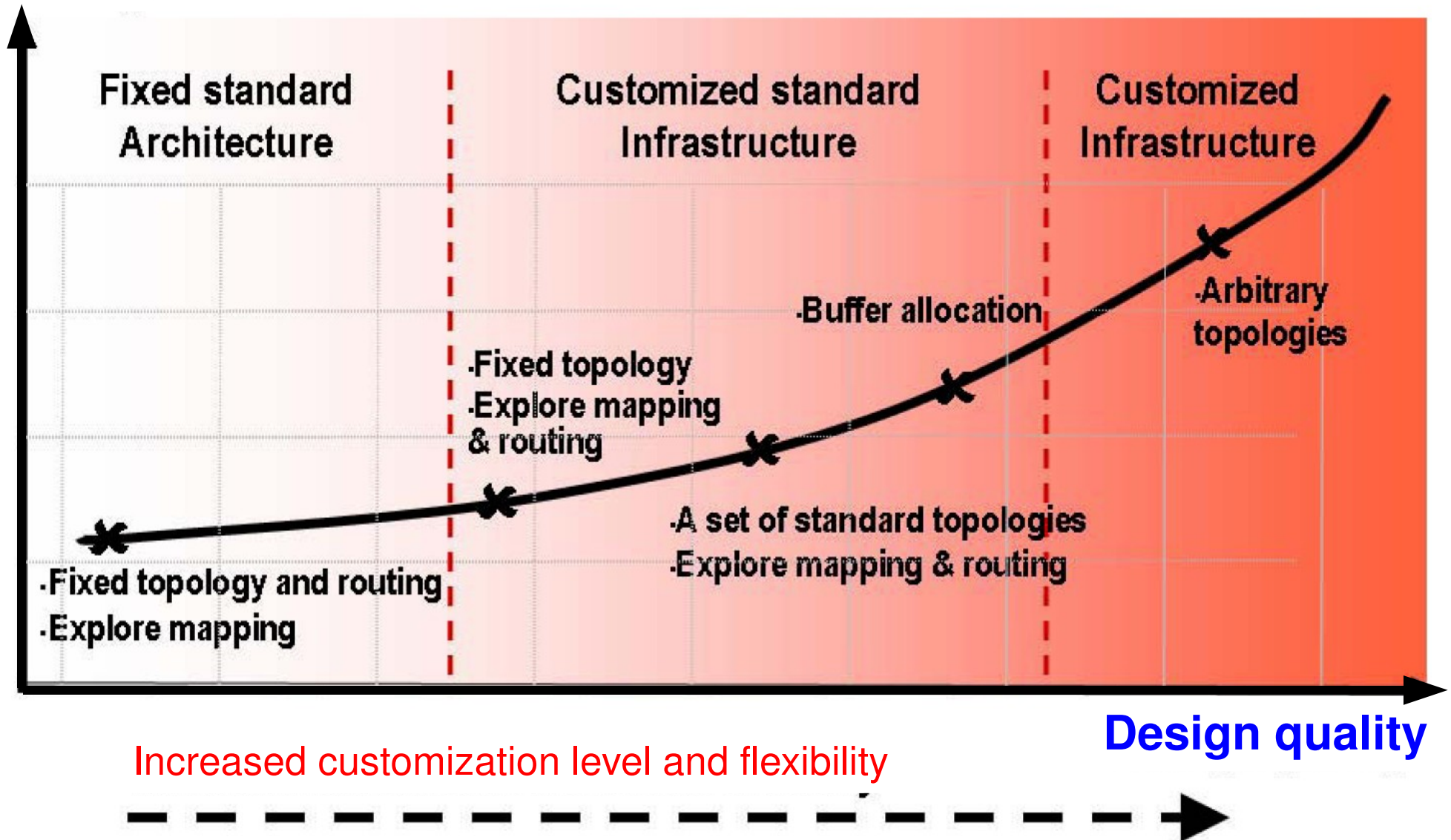
---

- Application Specific Routing Algorithms ✓
- Concurrent Mapping and Routing
- Dealing with Manufacturing Defects
- Encoding Scheme for Low Power

# DSE for NoC Architectures

Design effort

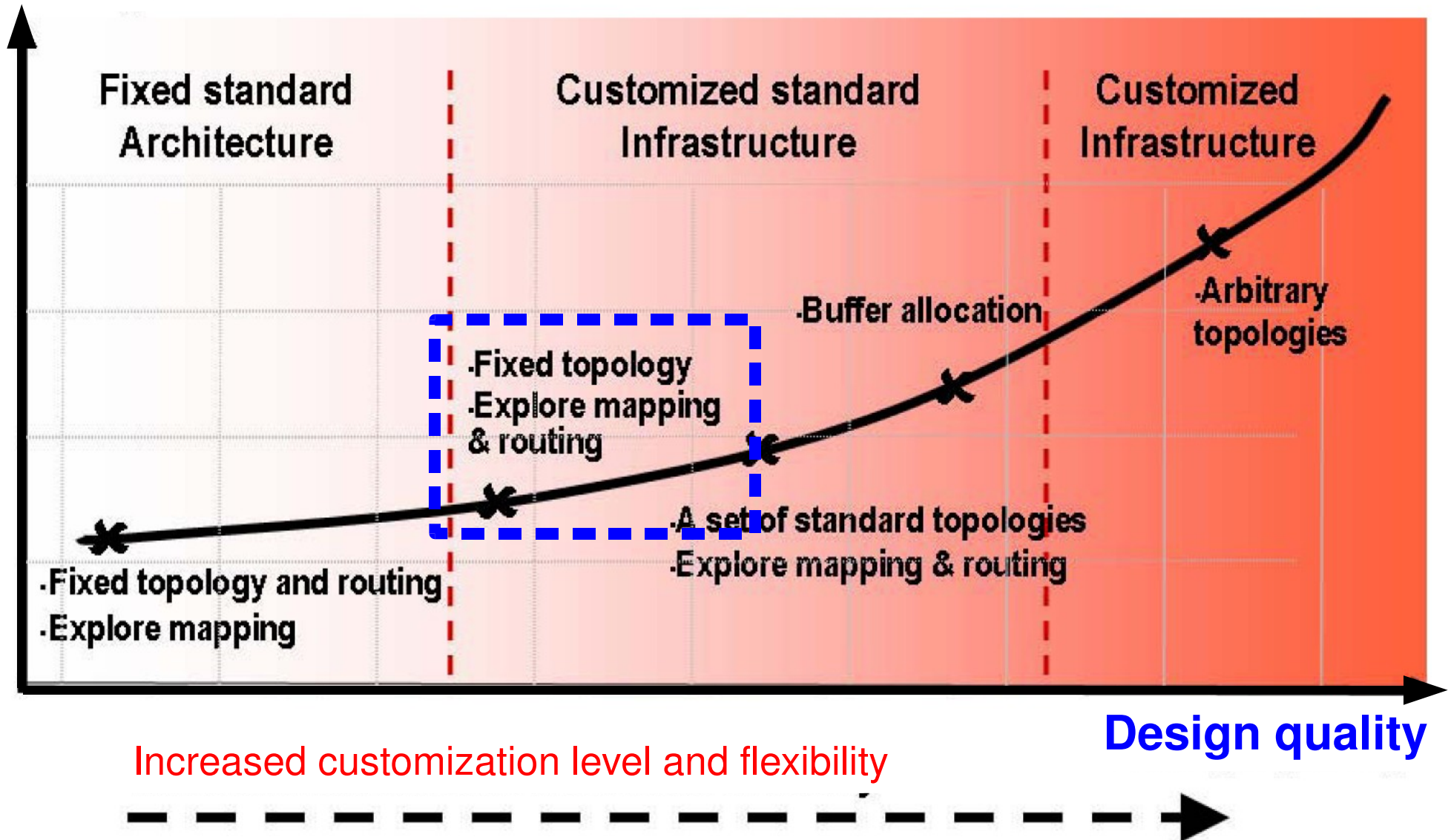
[Ogras et al., ASAP'05]



# DSE for NoC Architectures

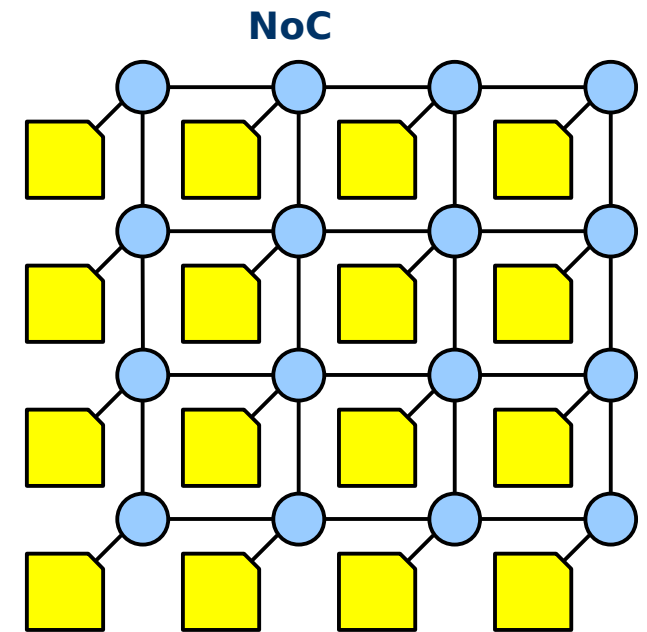
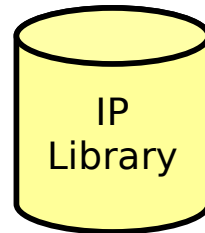
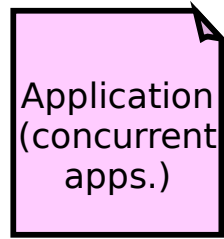
Design effort

[Ogras et al., ASAP'05]

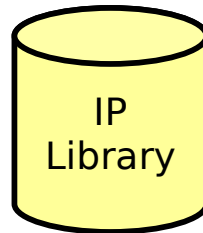
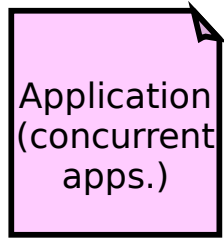


# The Mapping Problem

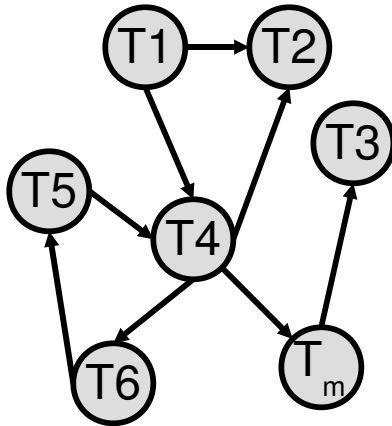
---



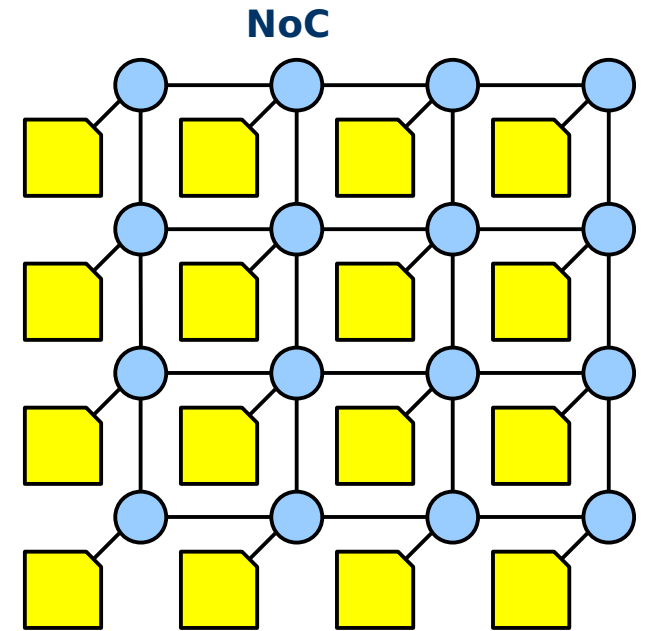
# The Mapping Problem



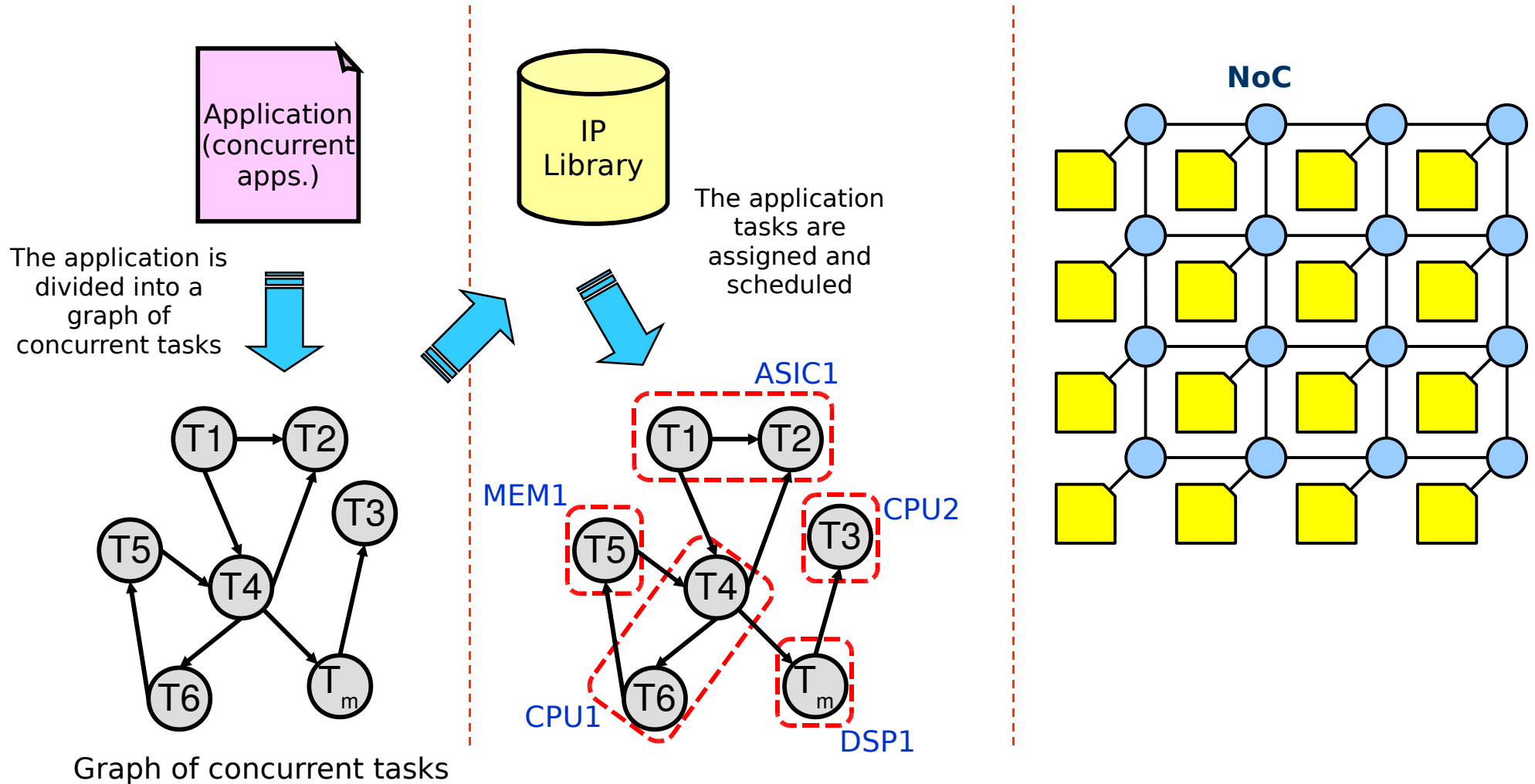
The application is  
divided into a  
graph of  
concurrent tasks



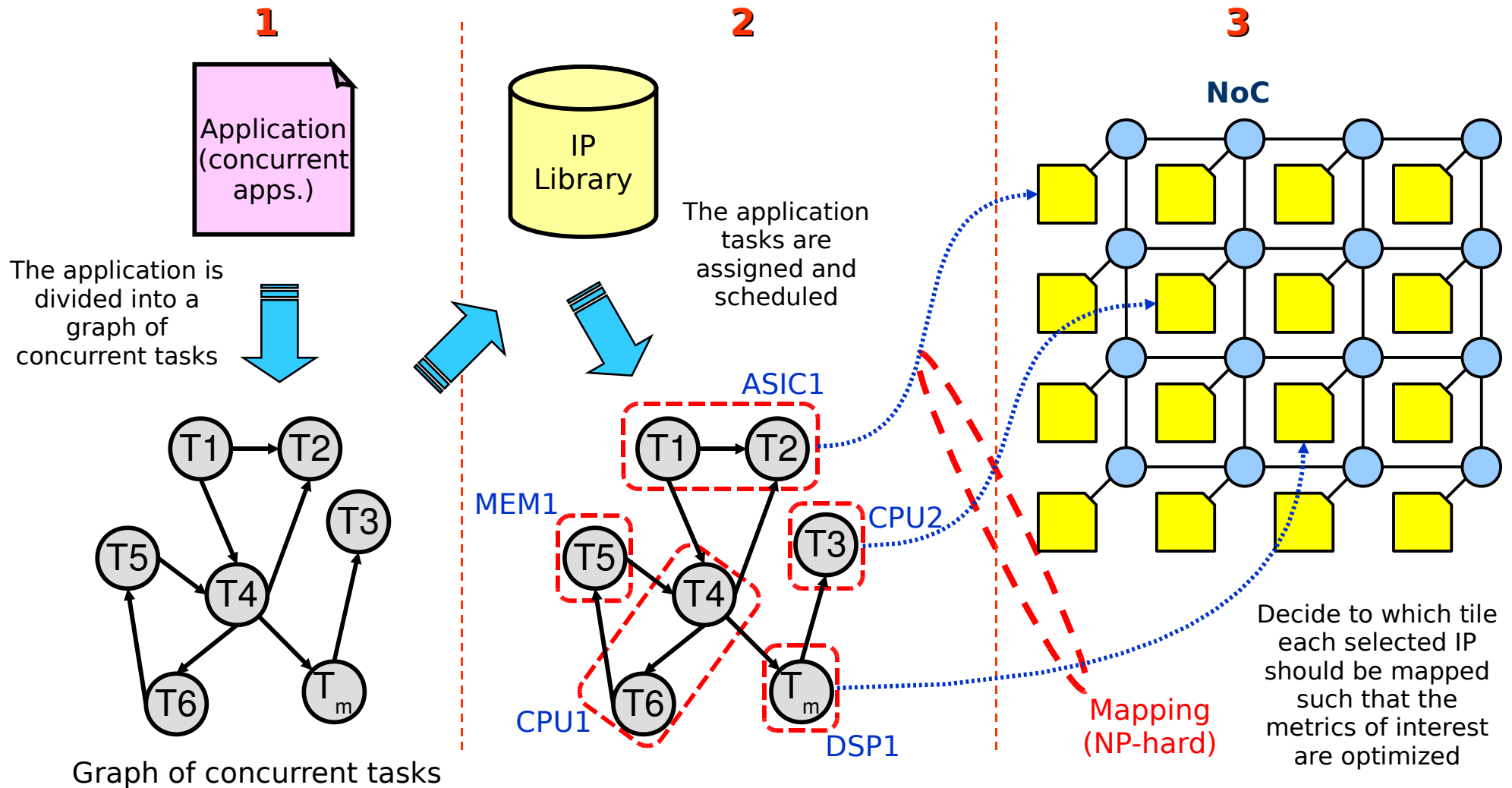
Graph of concurrent tasks



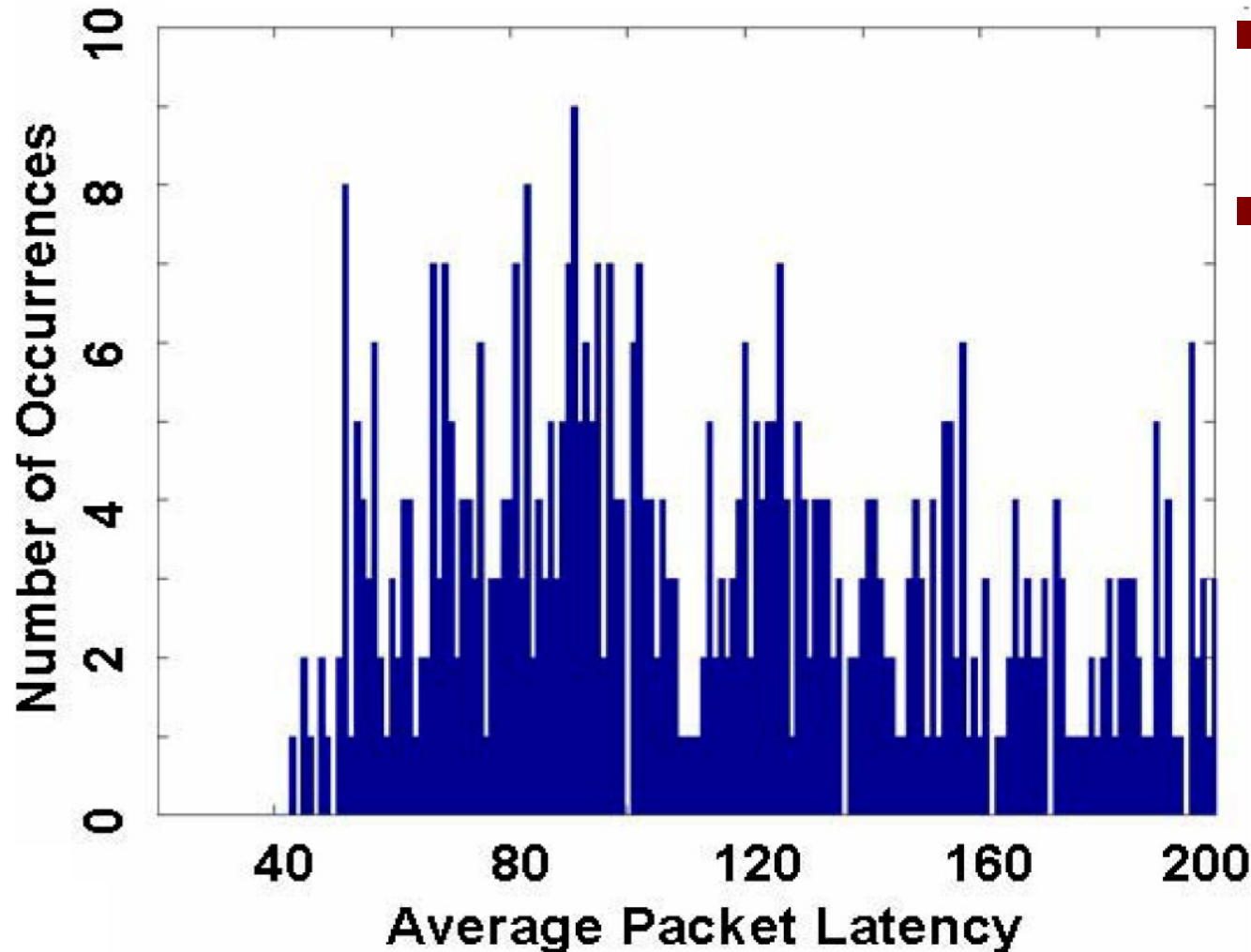
# The Mapping Problem



# The Mapping Problem



# Impact of Mapping on Performance



- A/V multimedia system
  - Mapped on 16 IPs
- Average packet latency of 3000 random mappings
  - Results for the top 478 mappings. The
  - Remaining 2522 mappings have latency much higher than 200 clock cycles

# Problem Formulation

---

- Given

- An application (or a set of concurrent applications) already mapped and scheduled into a set of IPs
- A network topology

- Find **the best mapping** and **the best routing function** which

- Maximize Performance (Minimize the mapping coefficient)
- Maximize fault tolerant characteristics (Maximize the *robustness index*)

- Such that

- The aggregated communications assigned to any channel do not exceed its capacity

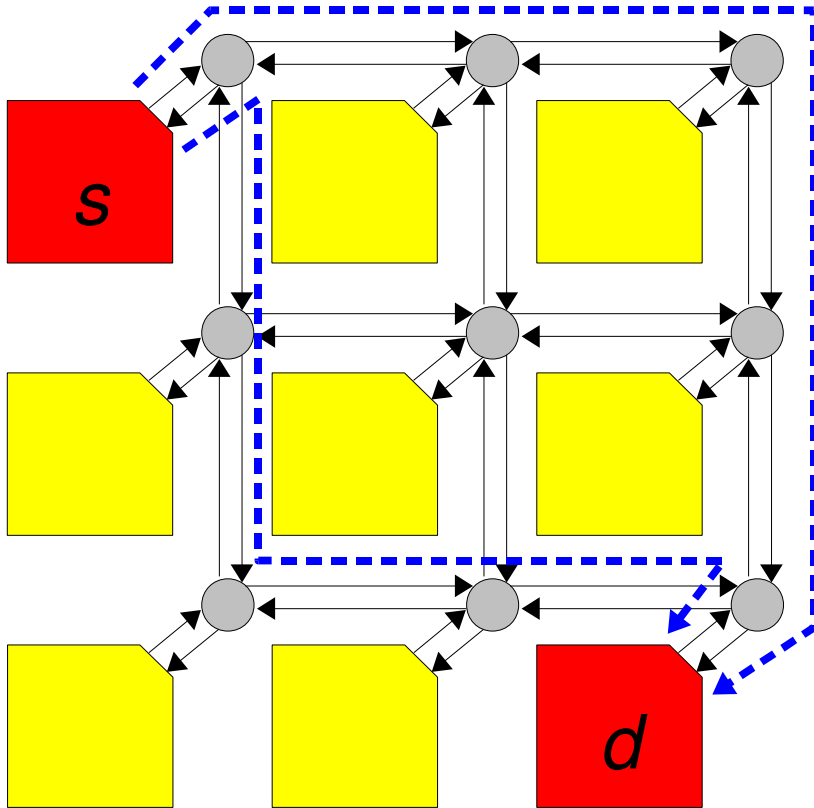
# Robustness Index

---

- Is an extension of the concept of *path diversity*

# Robustness Index

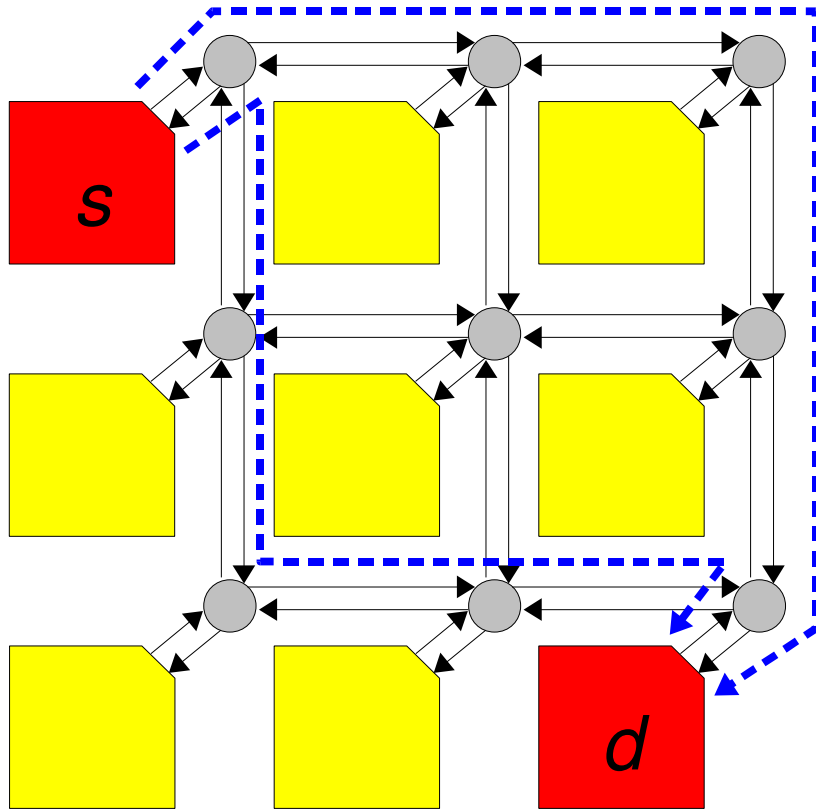
- Is an extension of the concept of *path diversity*



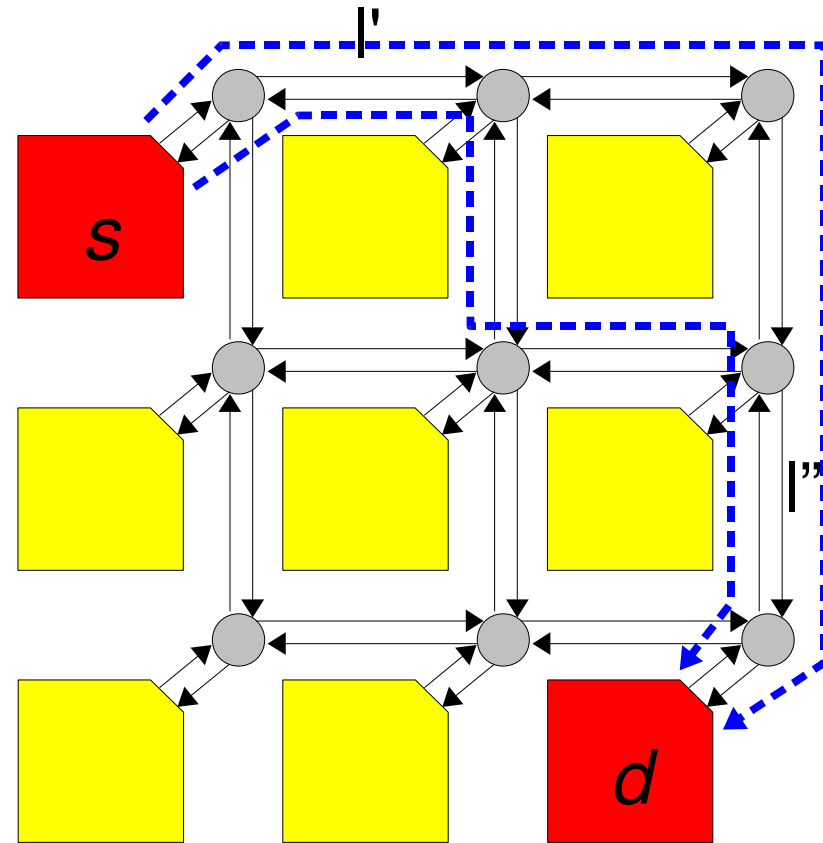
A single link fault does not compromise the communication from *s* to *d*

# Robustness Index

- Is an extension of the concept of *path diversity*



A single link fault does not compromise the communication from *s* to *d*

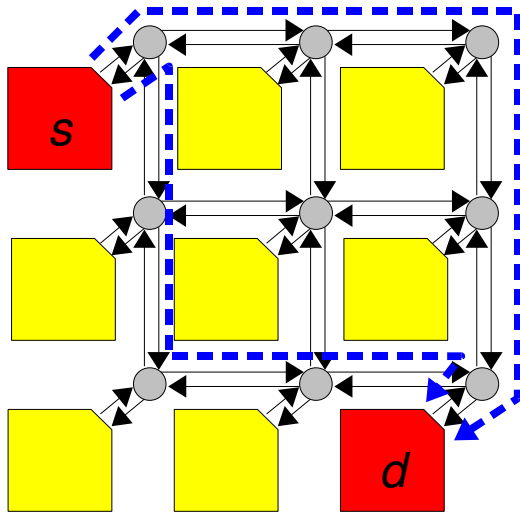


A single link fault in either *l'* or *l'''* makes it impossible the communication from *s* to *d*

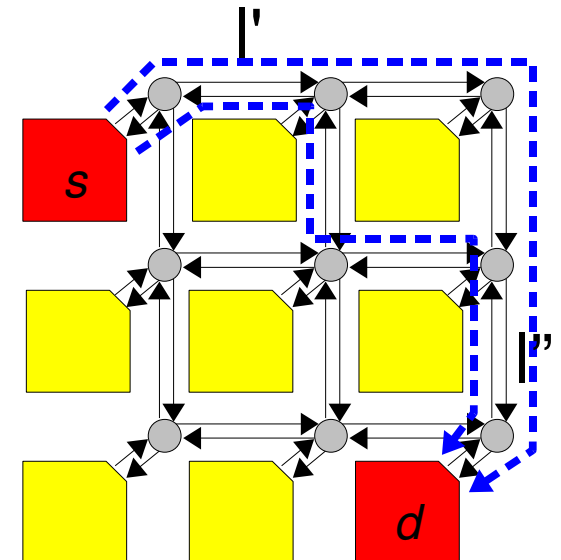
# Robustness Index

- Is an extension of the concept of *path diversity*

$$RI(c) = \frac{1}{|L(c)|} \sum_{l \in L(c)} |P(c) \setminus P(c, l)|$$



$$R(s \rightarrow d) = \frac{1+1+1+1+1+1+1+1}{8} = 1$$



$$R(s \rightarrow d) = \frac{0+1+1+1+1+0}{6} = 0.67$$

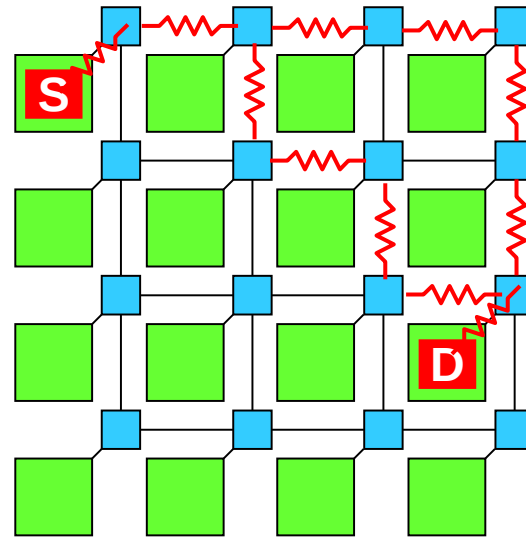
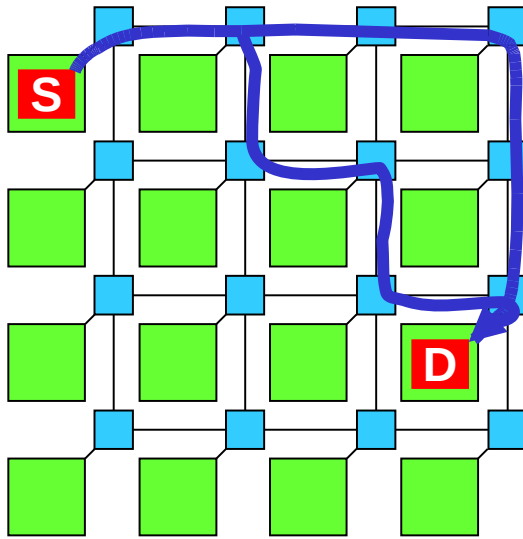
# Armament to Deal with the Mapping Problem

---

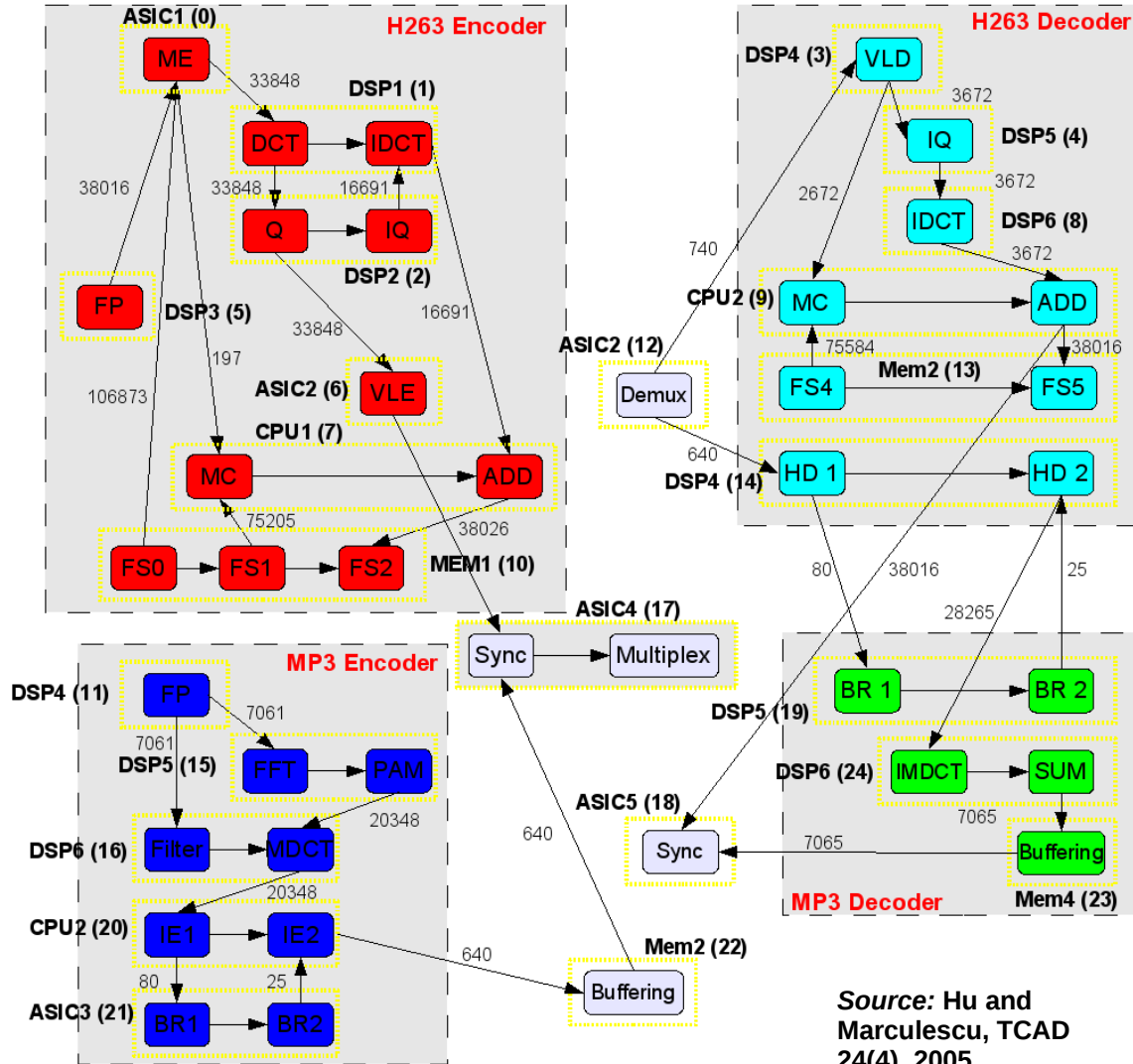
- Characterization of NoC resources
  - As the first step to develop a mapping technique for NoCs
- Find a model of the communication cost
  - Correlated with the performance metrics
  - Which does not require expensive simulations

# Model of Communication Cost

- Define a metric that does not depend on the traffic pattern
  - Based exclusively on the internode distance
    - ✓ Topology
    - ✓ Routing algorithm
- Equivalent distance
  - Analogy to the electrical equivalent resistance



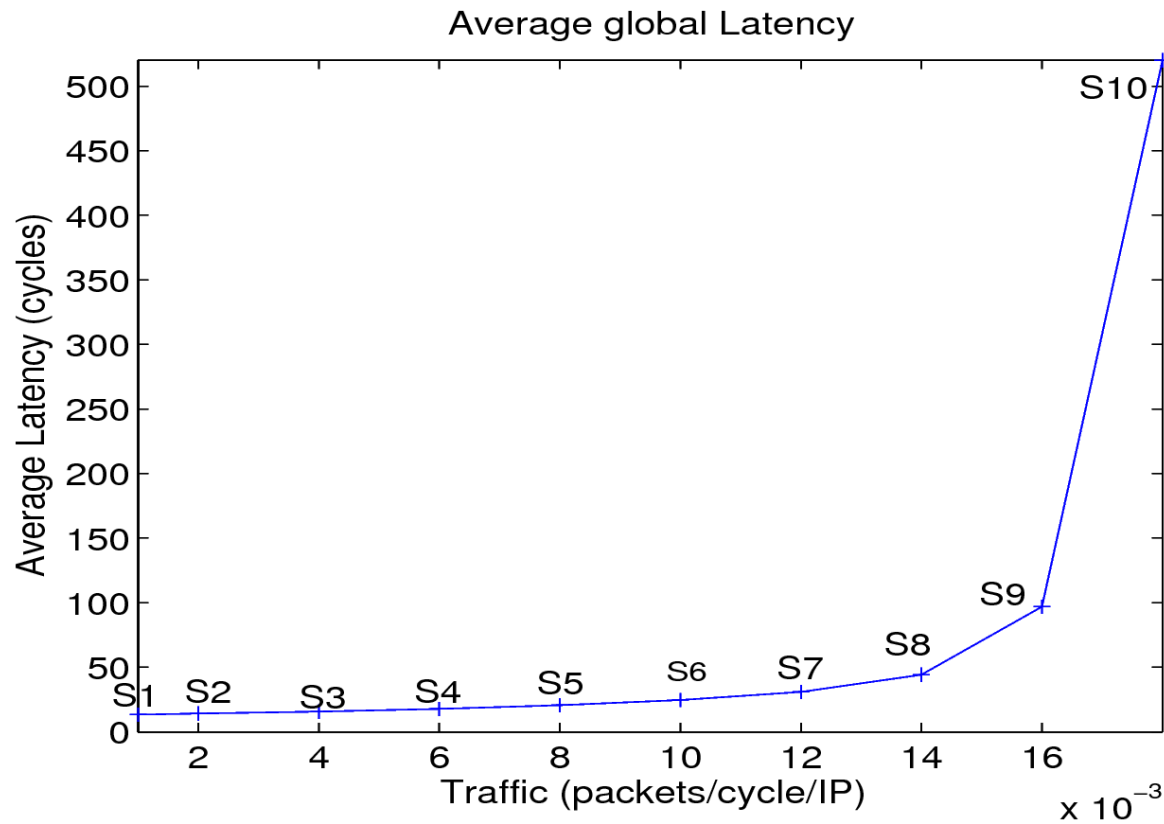
# Multi Media System



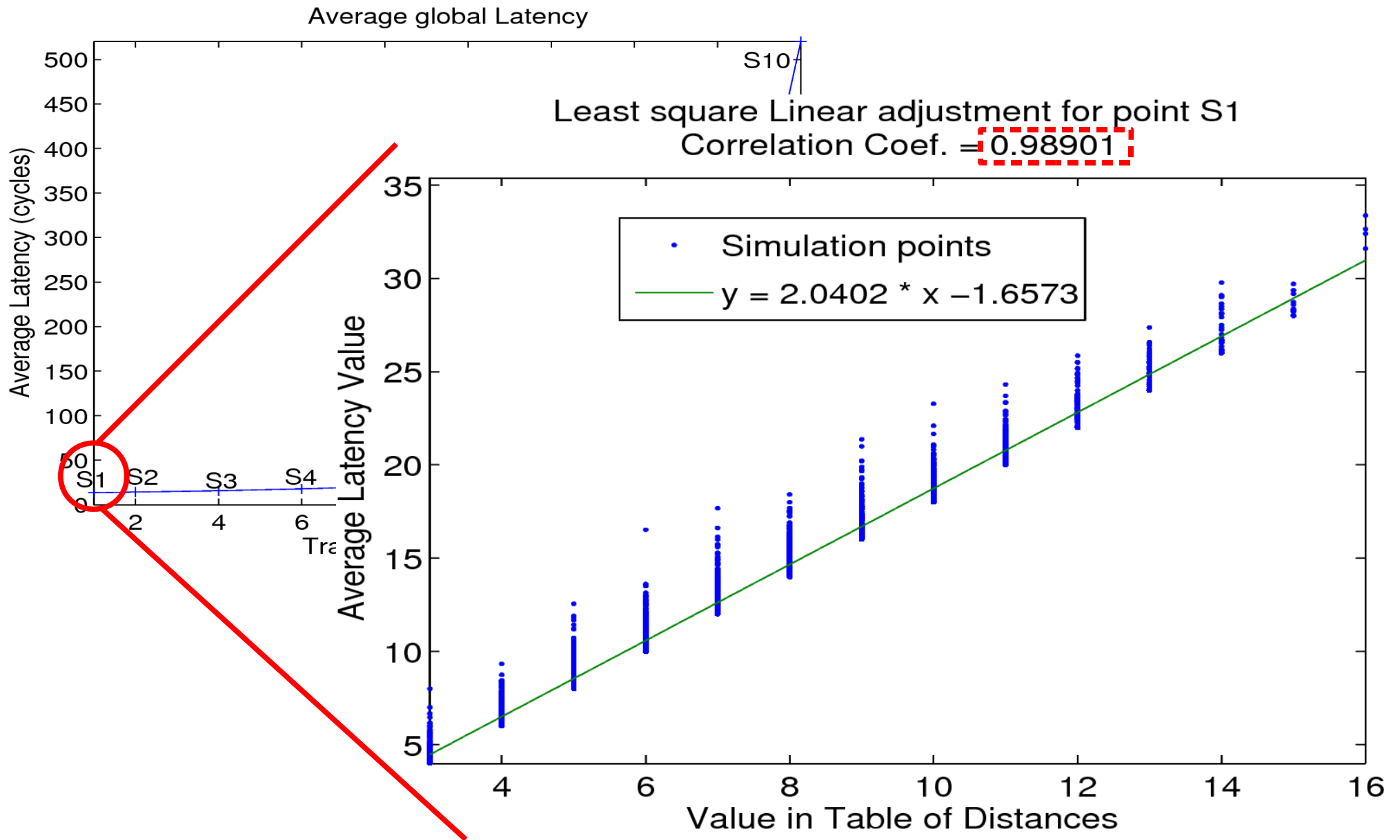
- 8x8 mesh-based NoC
- 8-flits packets
- 3-flits buffers
- SS packets injection distribution
- 50 simulations per point

Source: Hu and Marculescu, TCAD 24(4), 2005

# Simulation Results



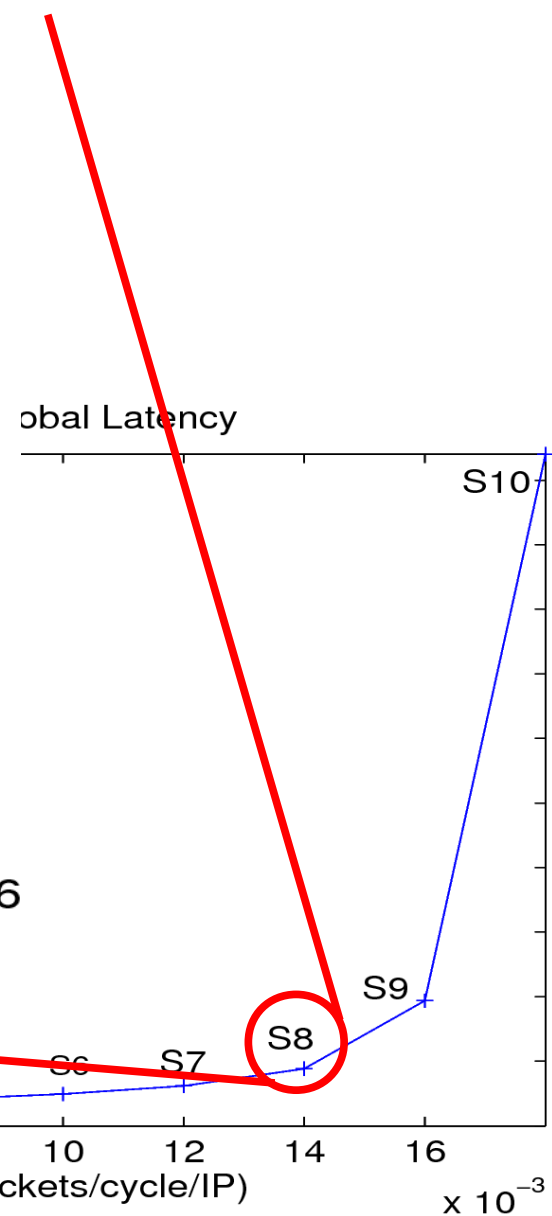
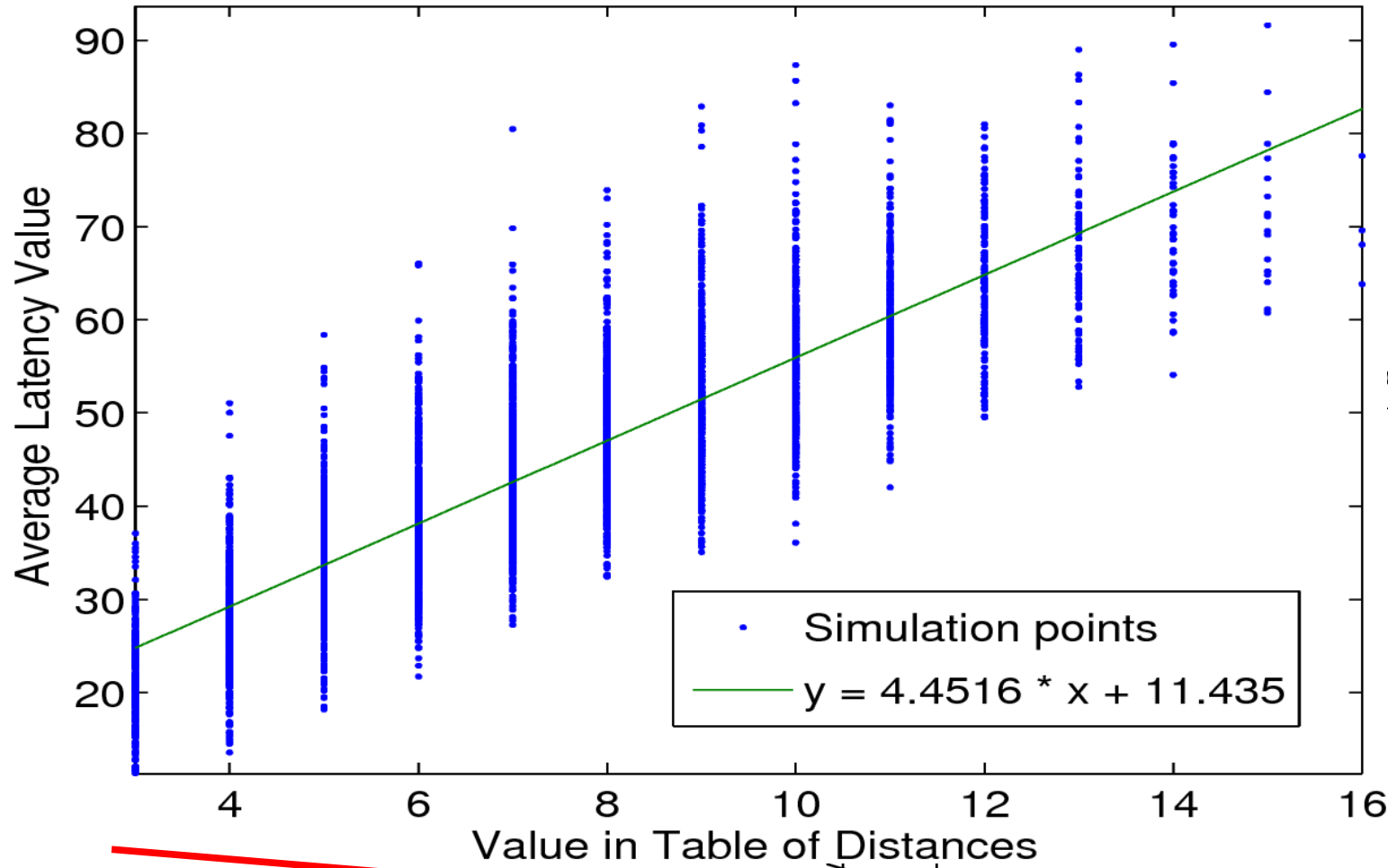
# Correlation Index for S1



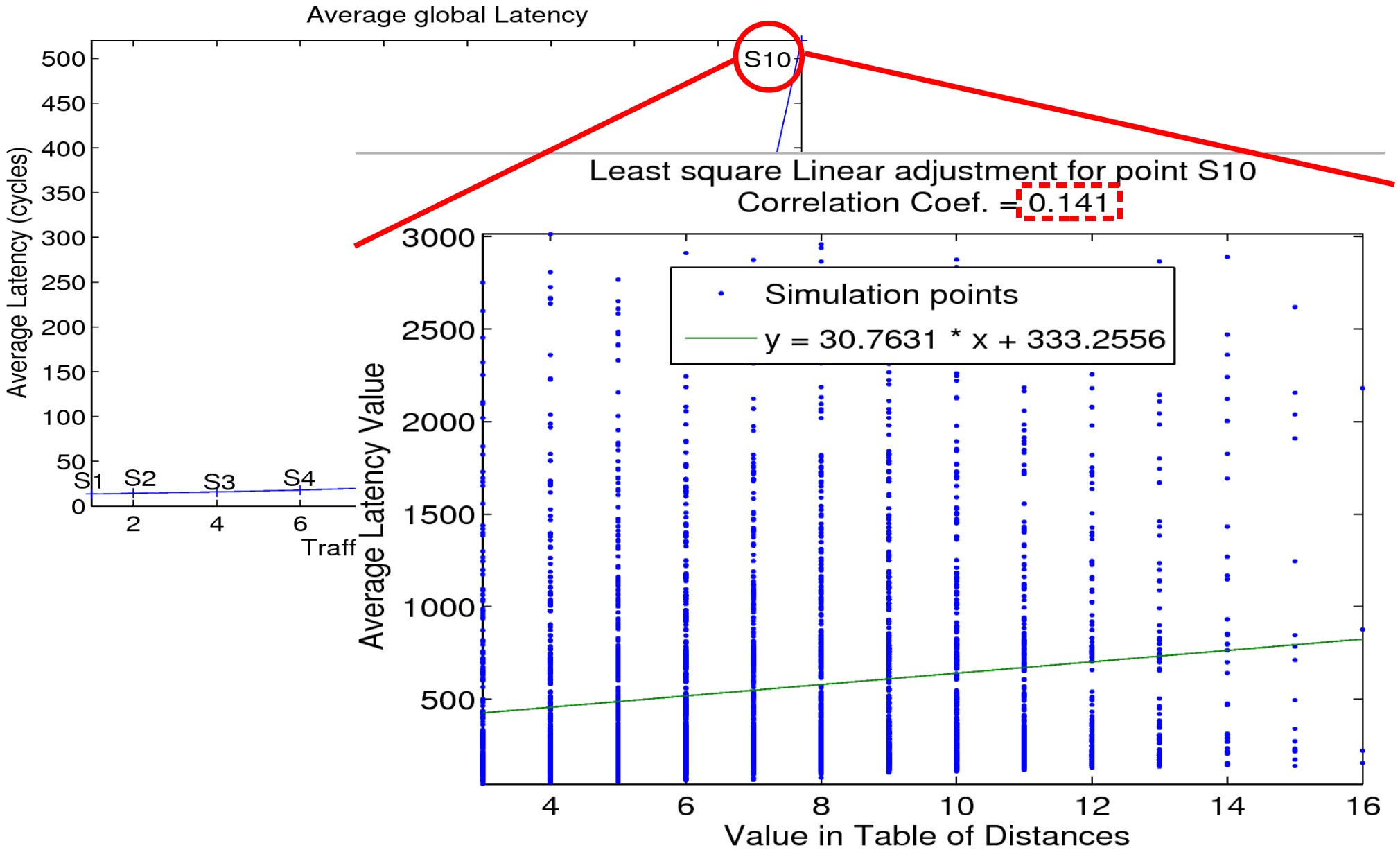
# Correlation Index for S8

Least square Linear adjustment for point S8

Correlation Coef. = 0.85507

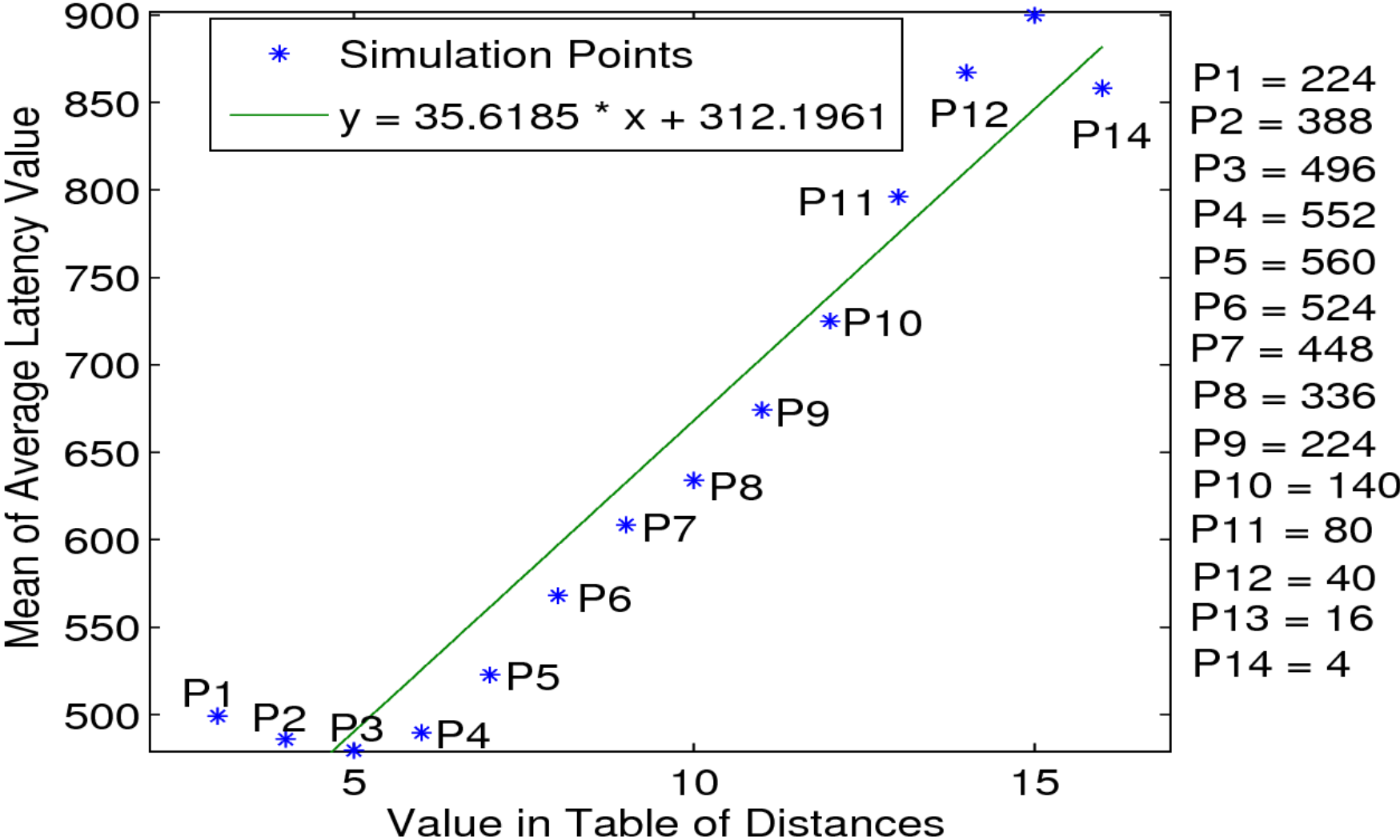


# Correlation Index for S10

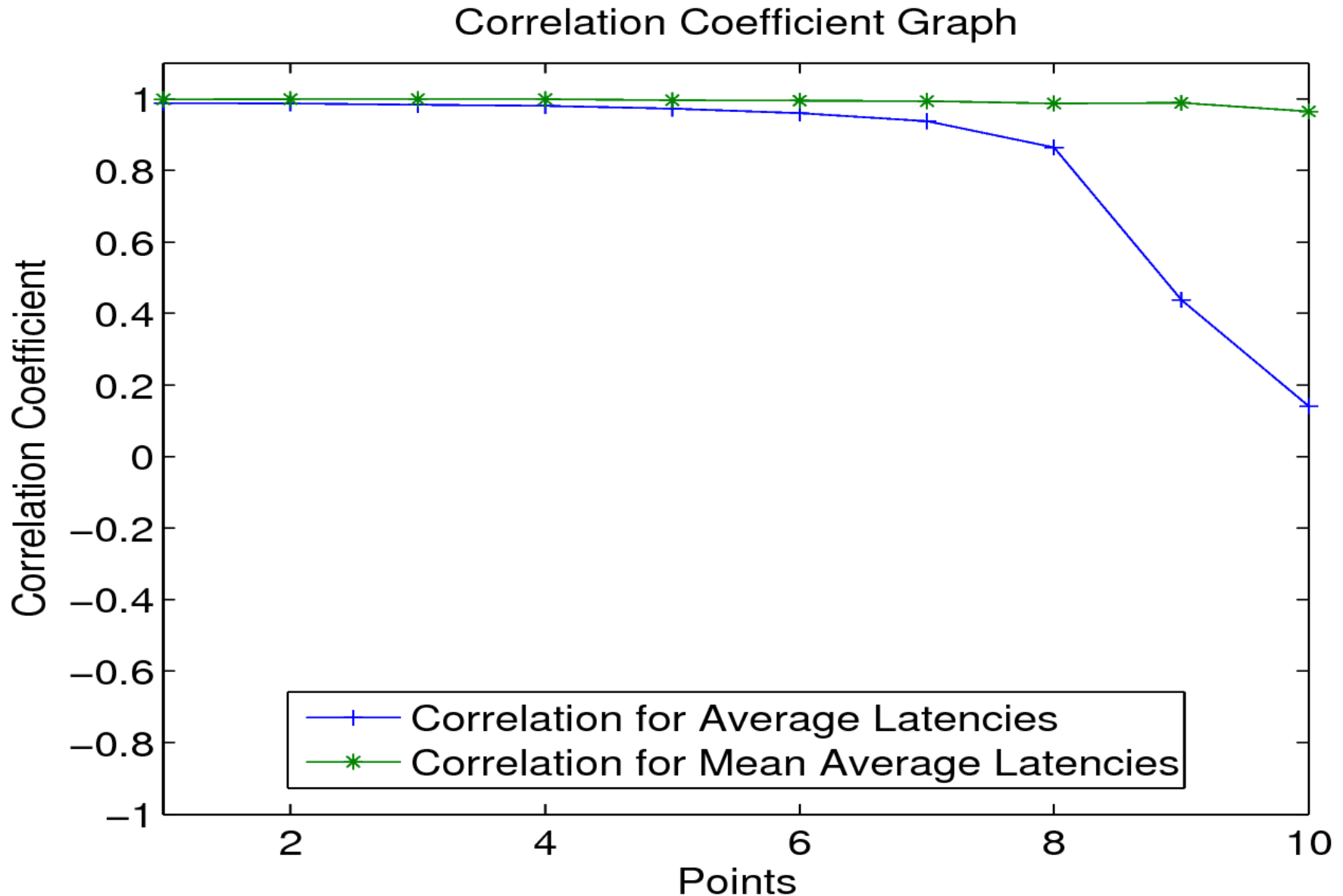


# Correlation with the Mean of the Avg Latencies

Least square Linear adjustment for point S10 (Mean)  
 Correlation Coef. = 0.96539

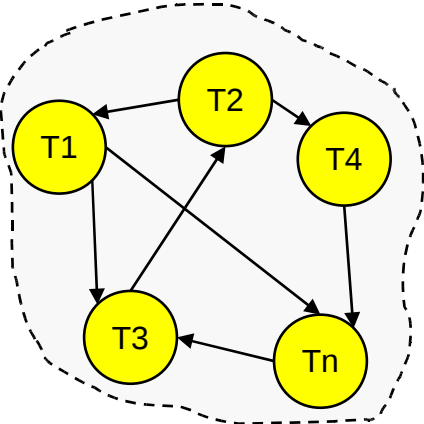


# Correlation Coefficients

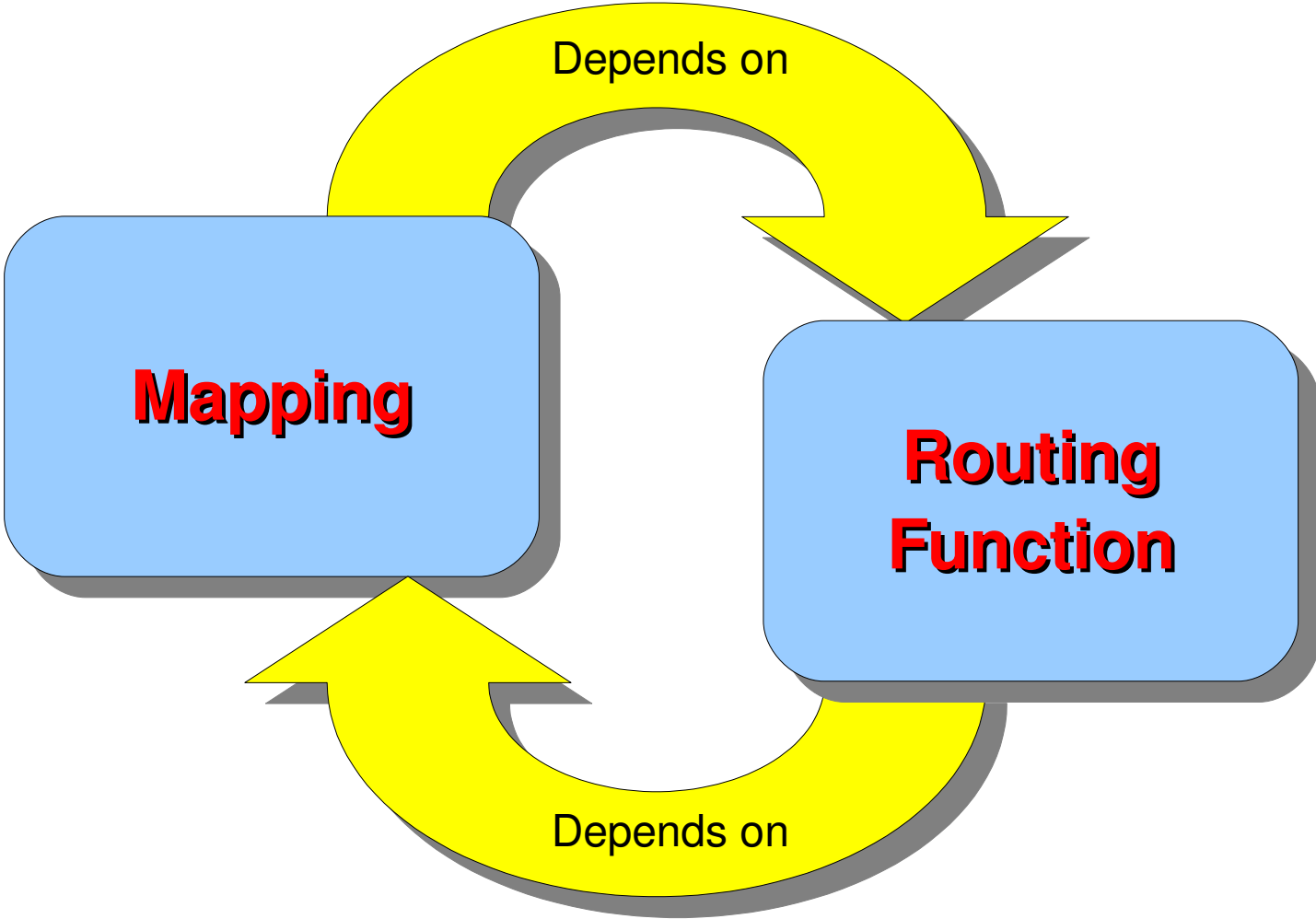
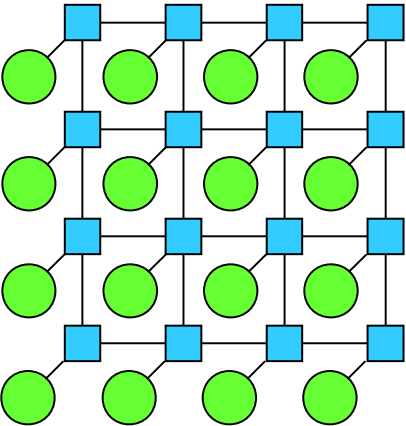


# Cyclic Dependency

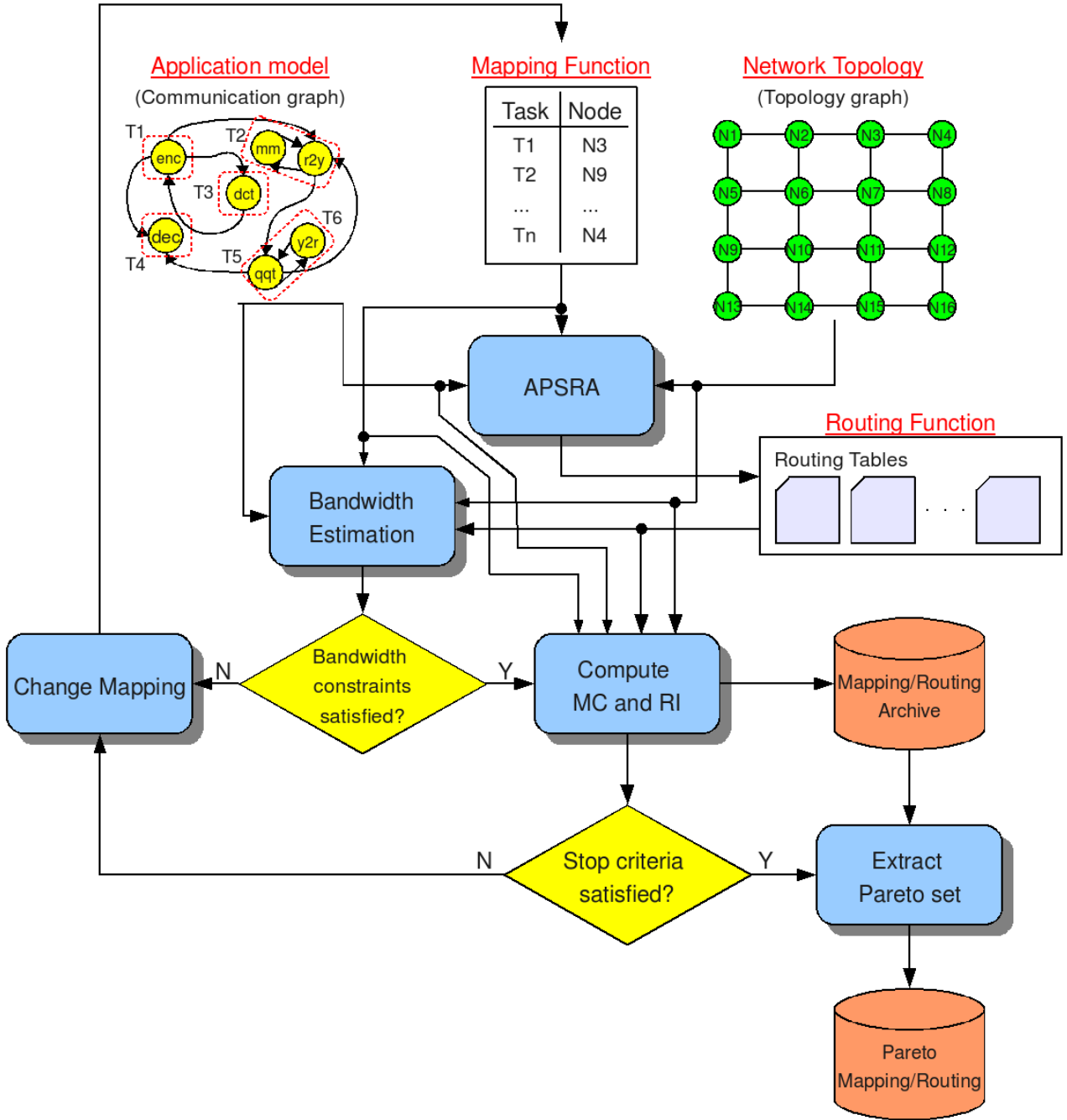
Application



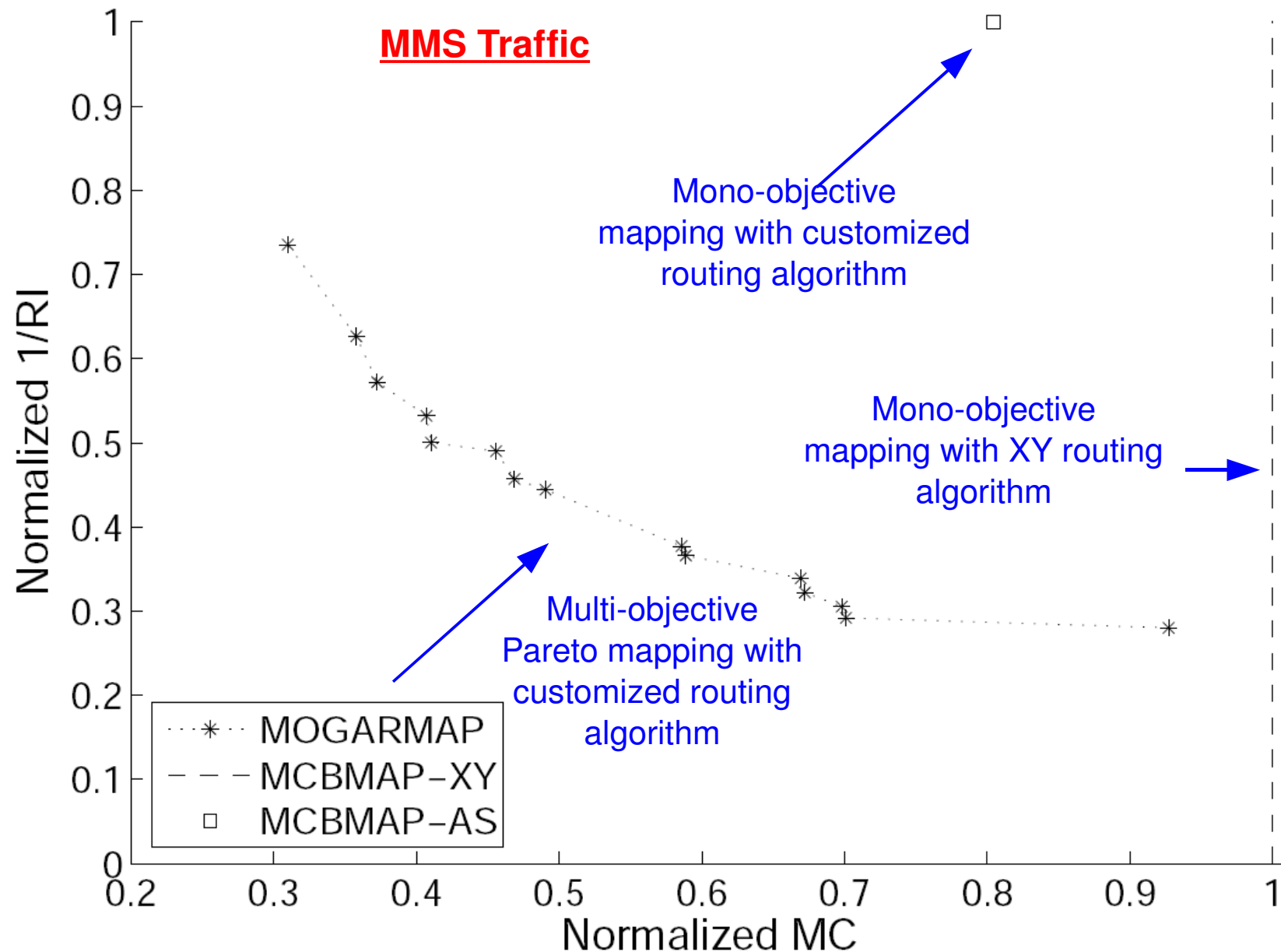
Network Topology



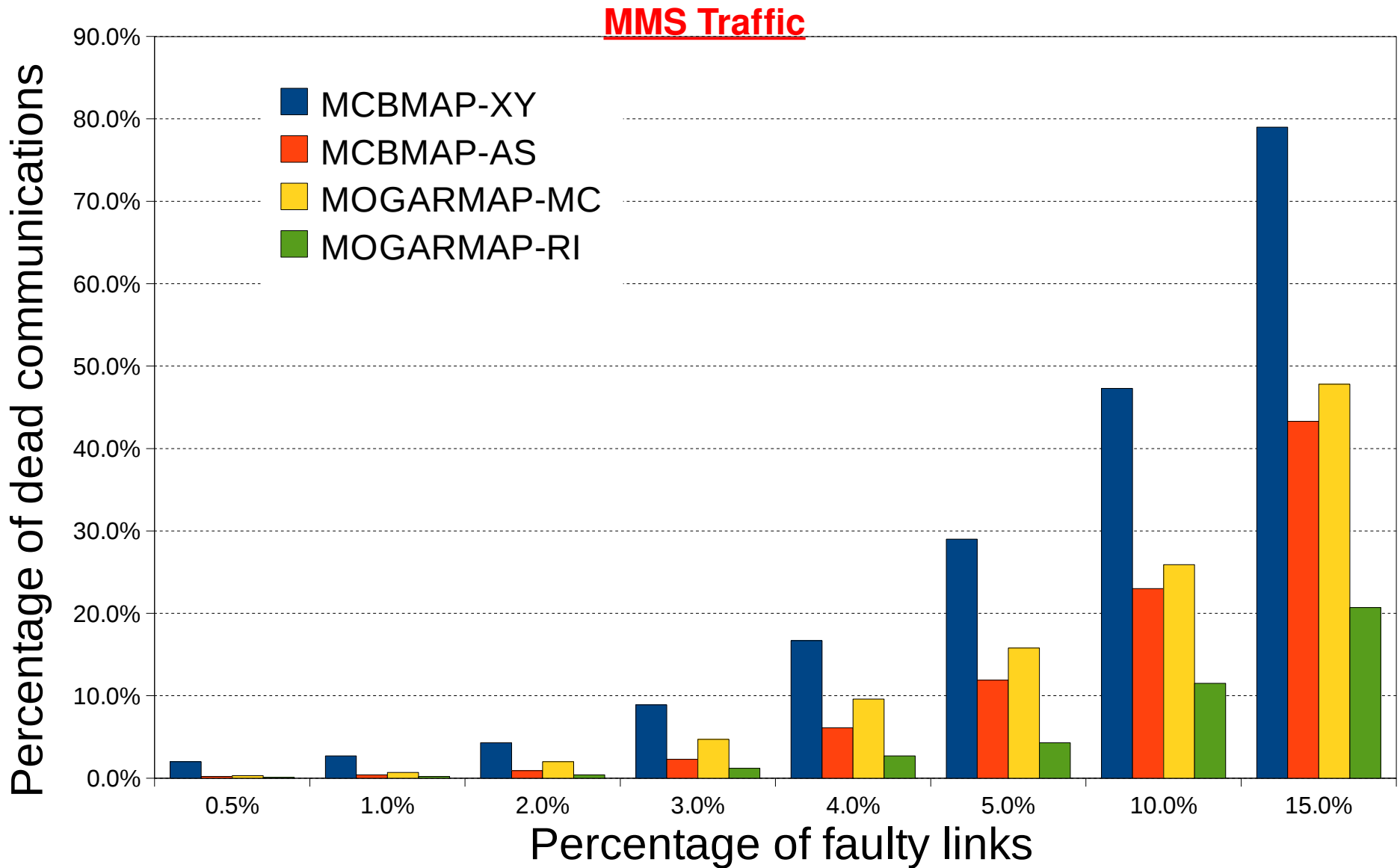
# Design Space Exploration Flow



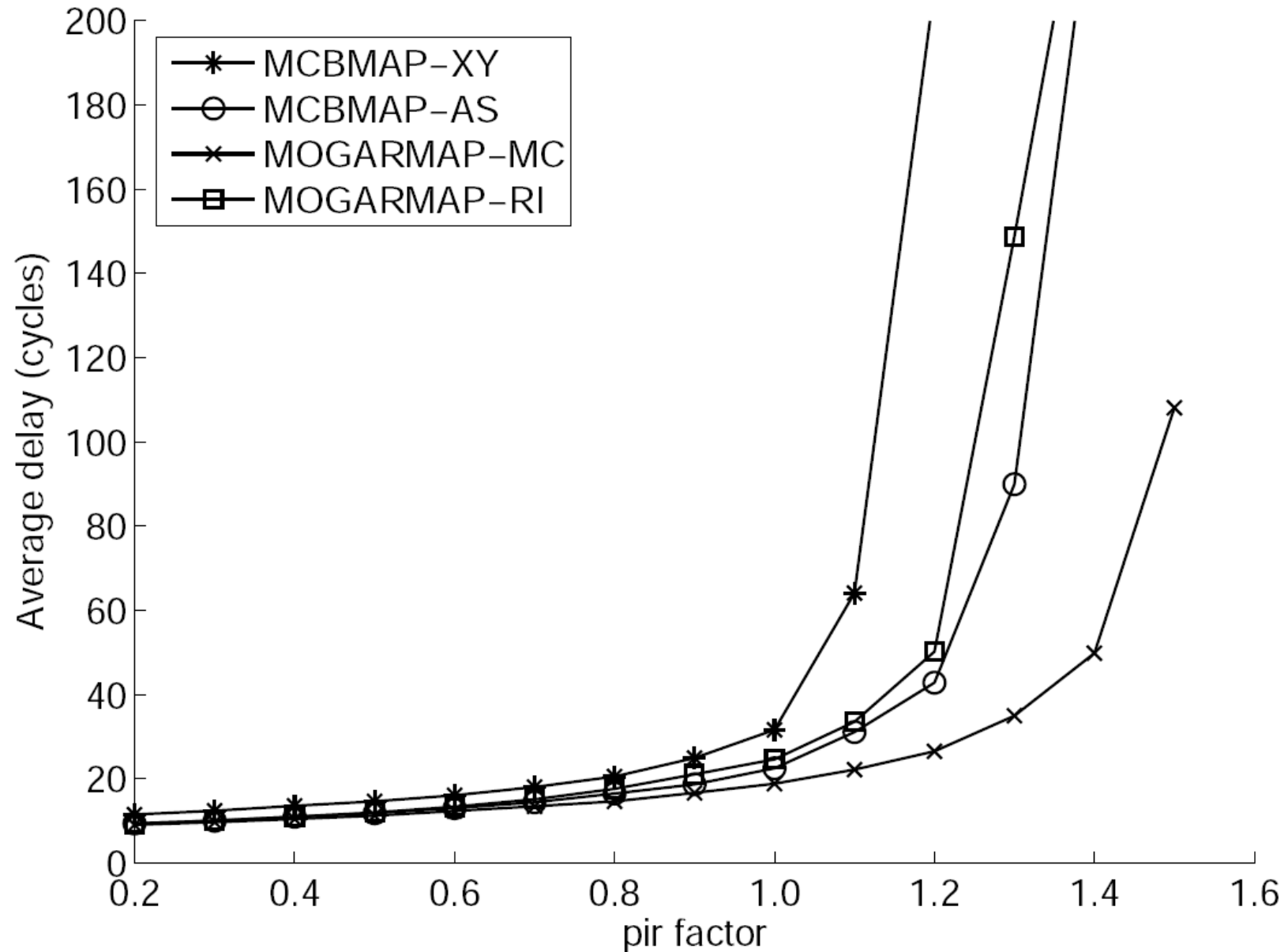
# Experiments: Pareto front



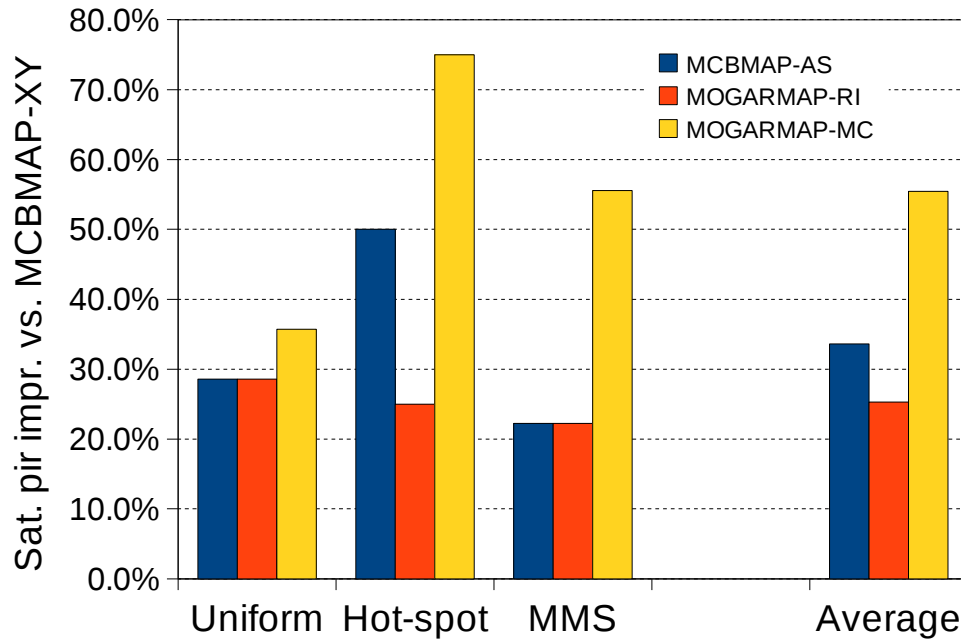
# Experiments: Dead Comms



# Experiments: Delay

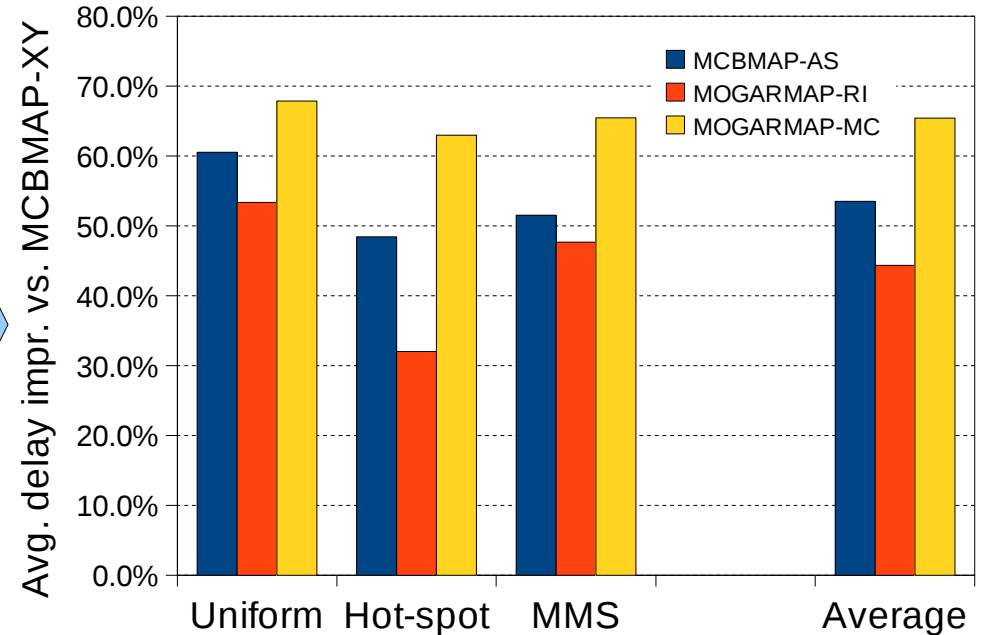


# Experiments: Summary Improvement



Saturation *pir* improvement (MCBMAP-XY as baseline)

Average delay reduction (MCBMAP-XY as baseline)



# Outline

---

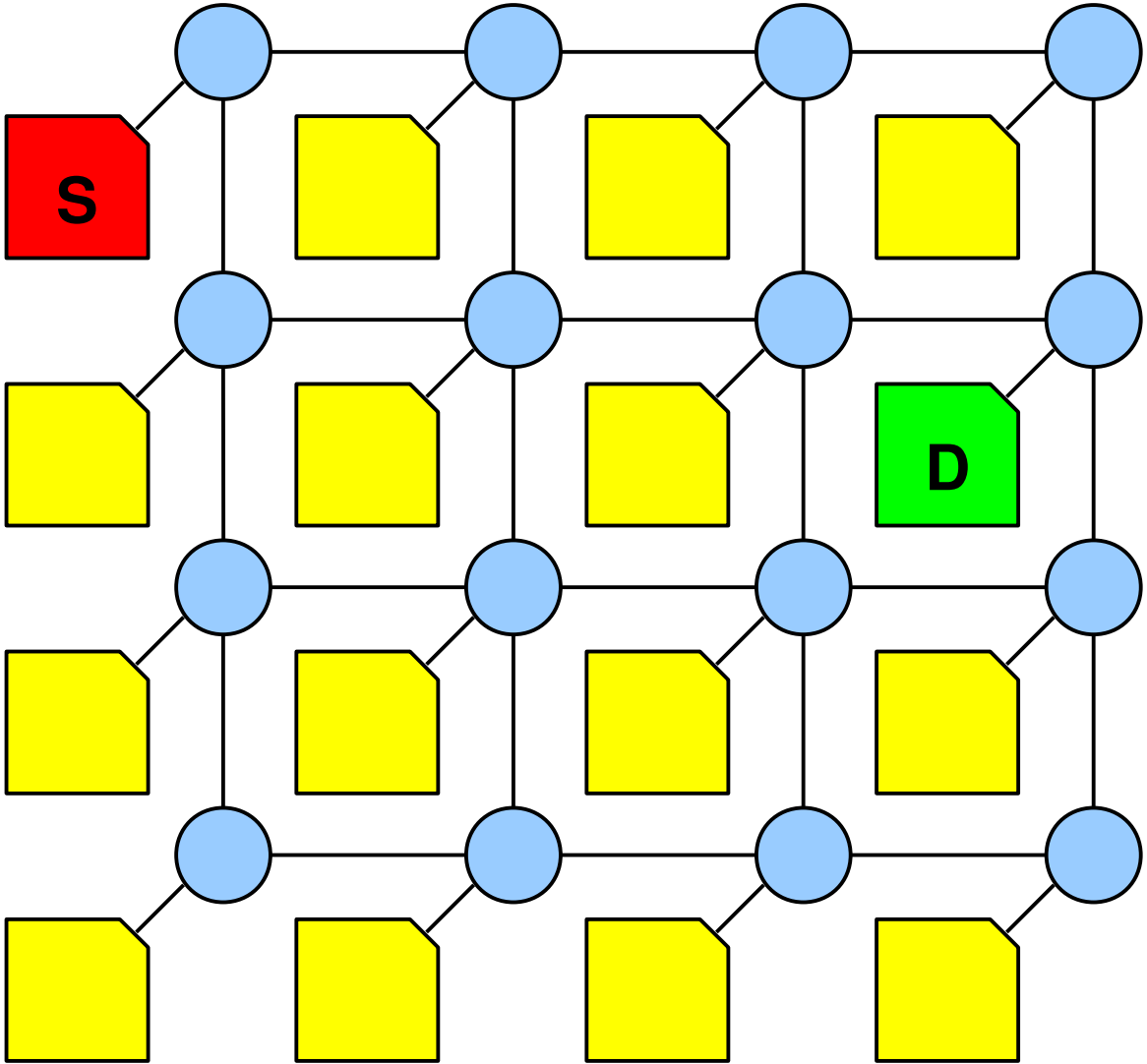
- Application Specific Routing Algorithms ✓
- Concurrent Mapping and Routing ✓
- Dealing with Manufacturing Defects
- Encoding Scheme for Low Power

# Introduction and Motivations

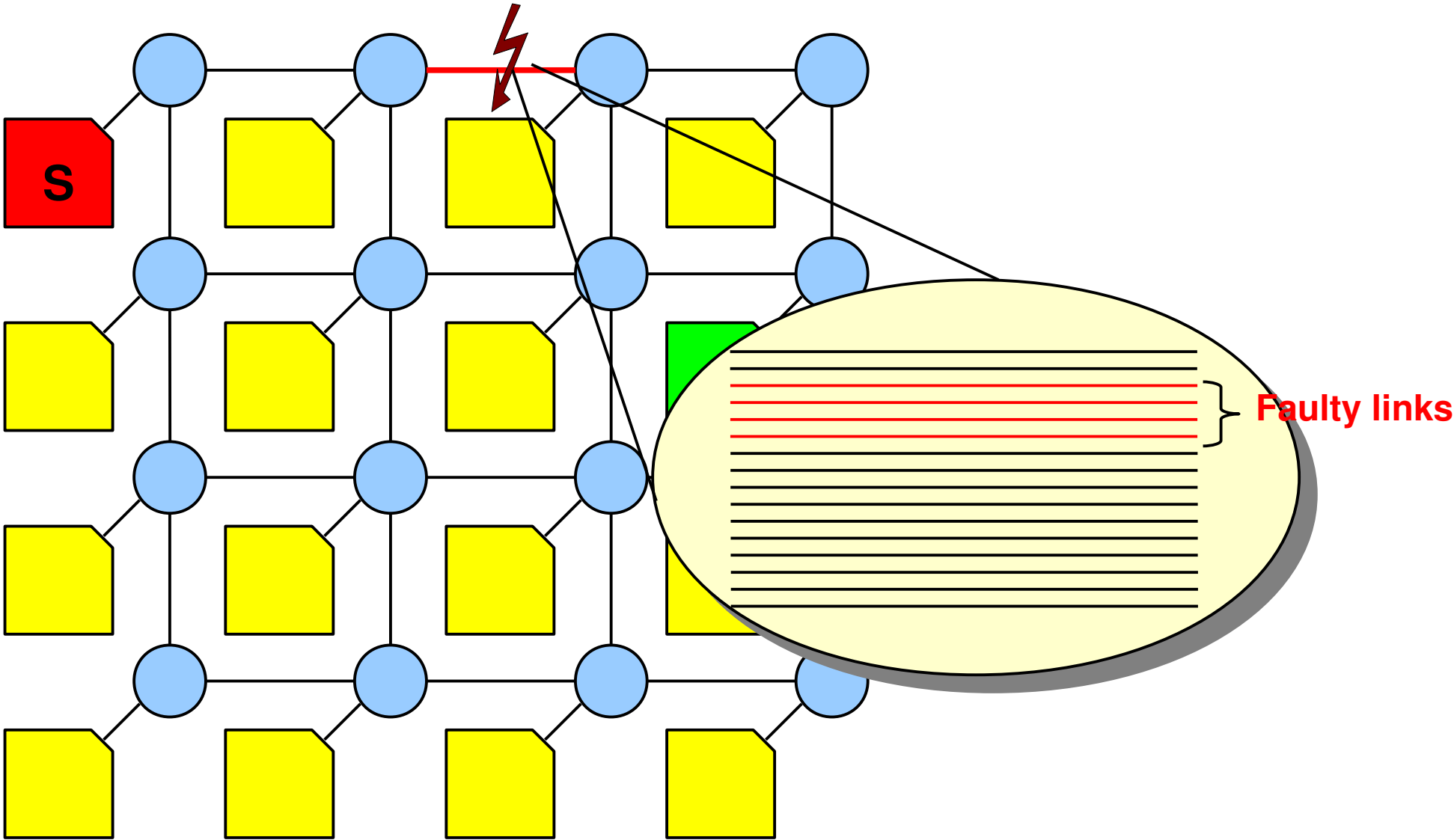
---

- Initial yield of complex SoC is very small
  - Yield goes down with size and complexity of the chip
  - Dealing with the reduction of yield due to manufacturing defects
    - ✓ Isolate faulty blocks of the chip
    - ✓ Using the chip as a “depowered” chip (E.g., Sun UltraSparc T1)
- On-chip communication system
  - Represents the heart of the system
  - Gets a quite high percent of the system silicon area
    - ✓ E.g., 20% of the silicon area in Intel's TeraScale 80-cores chip
    - ✓ High probability of being affected by manufacturing defects

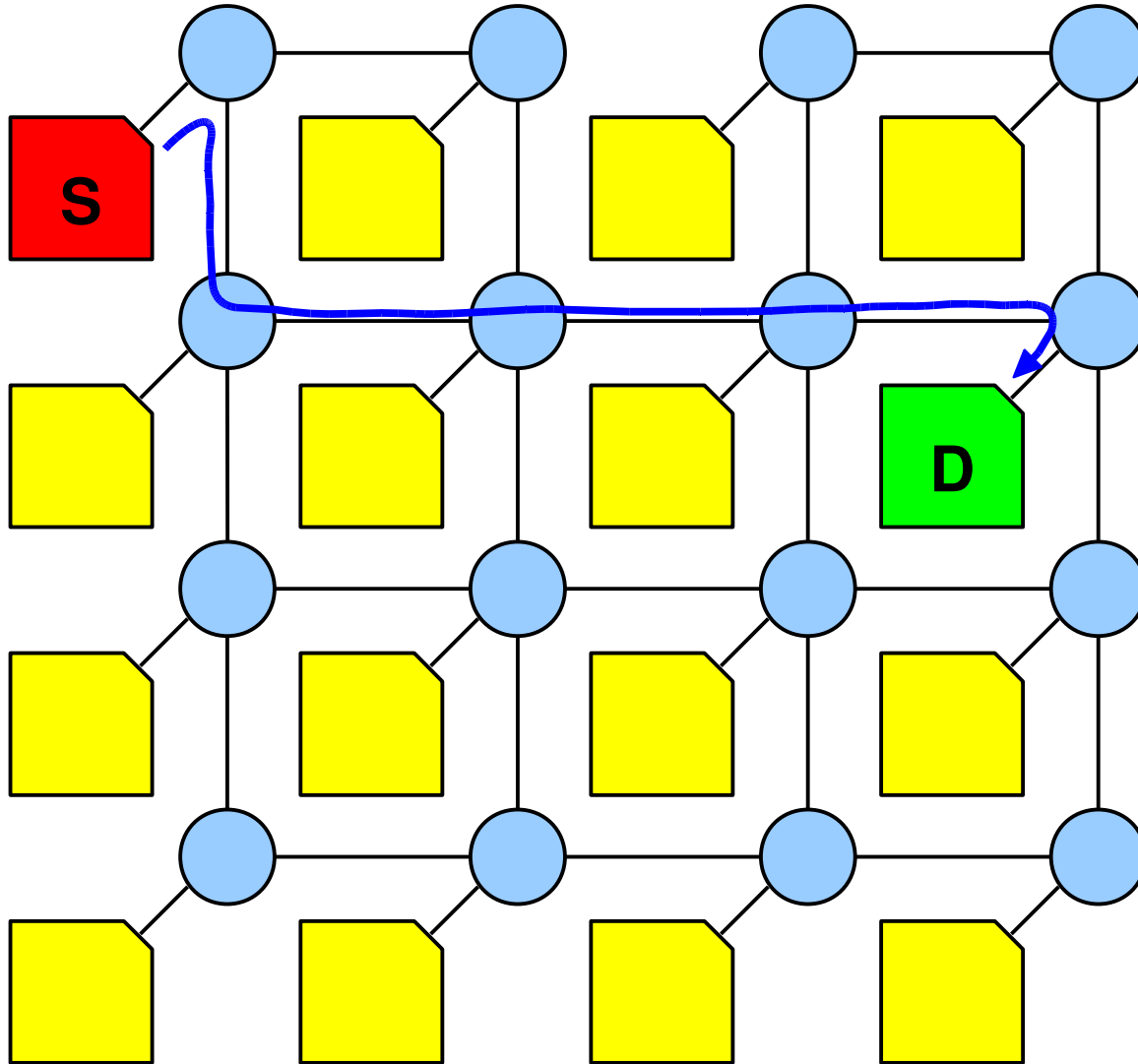
# Managing Faulty Links



# Managing Faulty Links



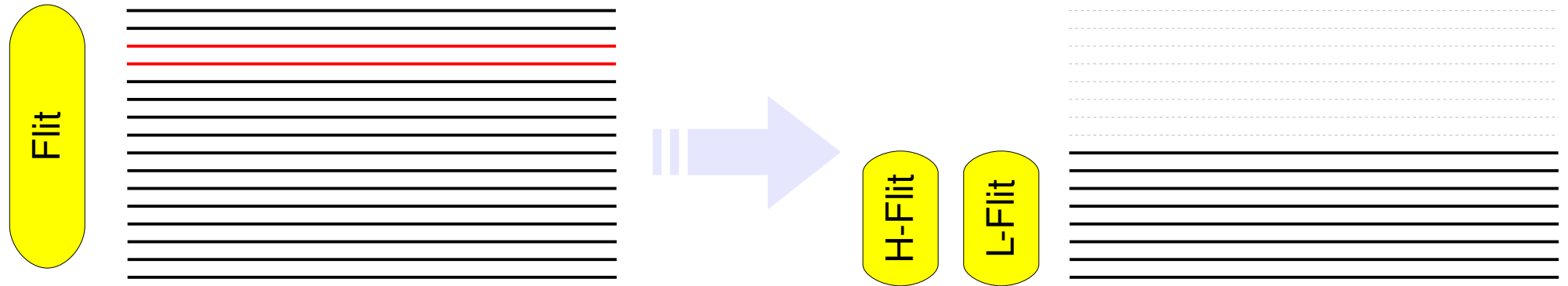
# Managing Faulty Links



## ■ Faulty links elimination

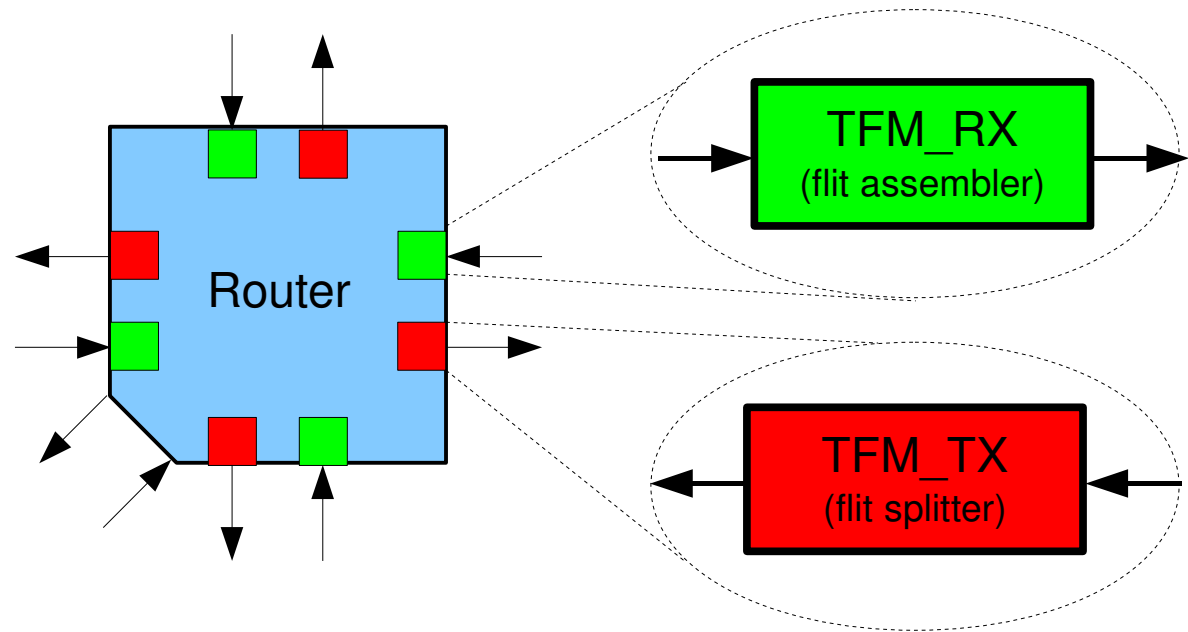
- ➔ The *routing function* is computed on the basis of the network **filtered** by all the *fully faulty* or *partially faulty* links

# Managing Faulty Links



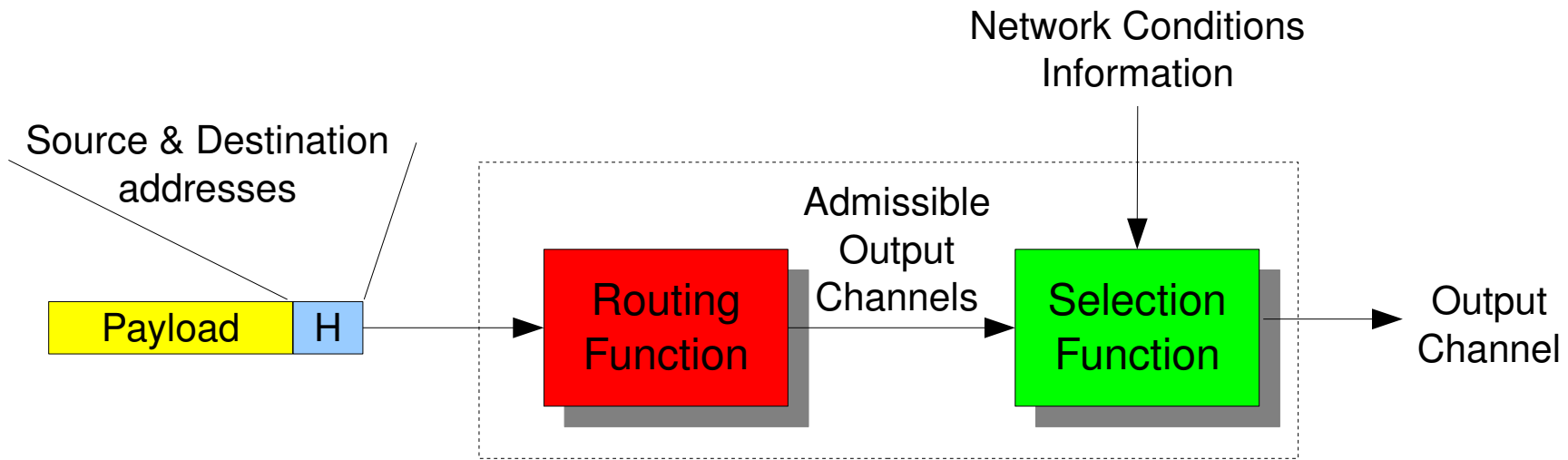
## ■ Partially Faulty Links usage

- The *routing function* is computed on the basis of the network filtered by all partially faulty links with a *fault degree* greater than a certain threshold

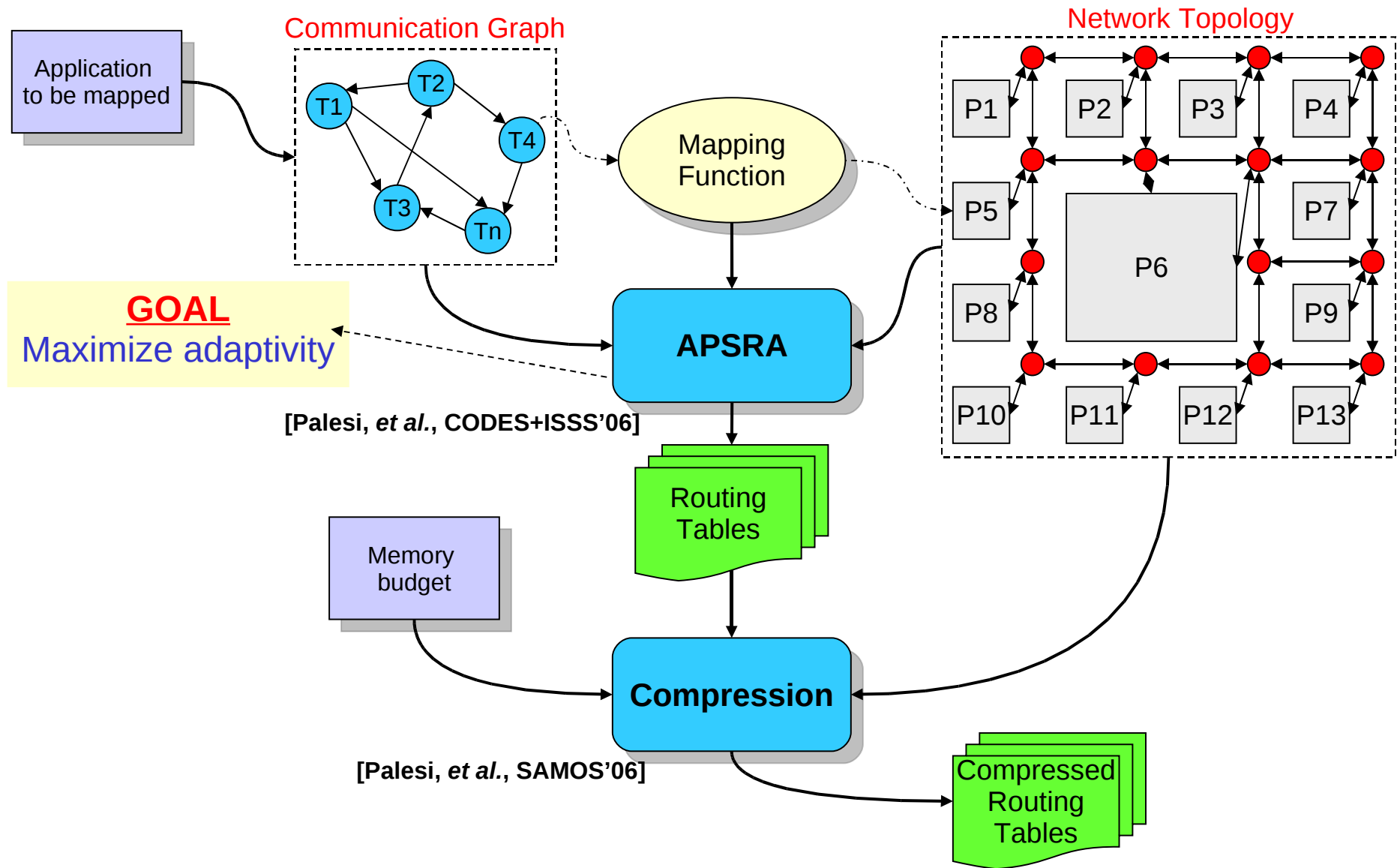


# Two Questions

- How routing paths are determined?
  - Routing function
- When does a partially faulty link should be used?
  - Selection function



# Routing Function



# Selection Function

---

- Faulty Links elimination (**FE**)
- Partially Faulty Links usage strategy (**FU**)
- Partially Faulty Links usage with Look-ahead strategy (**FUL**)
- Load Balancing based strategies (**LB**)
  - **FE+LB**
  - **FU+LB**
  - **FUL+LB**

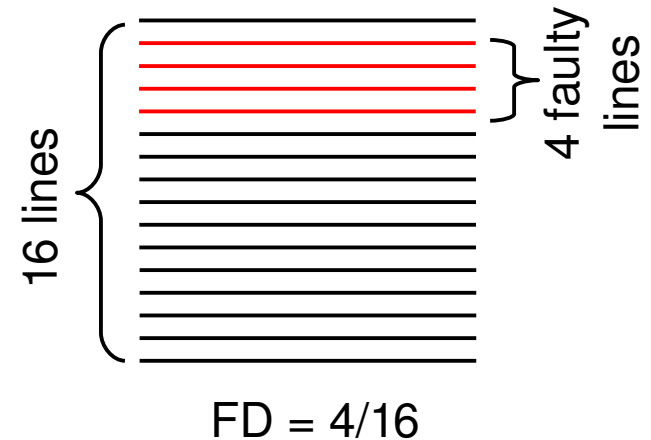
# Partially Faulty Links usage Strategy

## ■ In *low traffic* conditions

→ Only fault free links are chosen, if available. Otherwise, links with lowest FD are chosen

## ■ In *high traffic* conditions

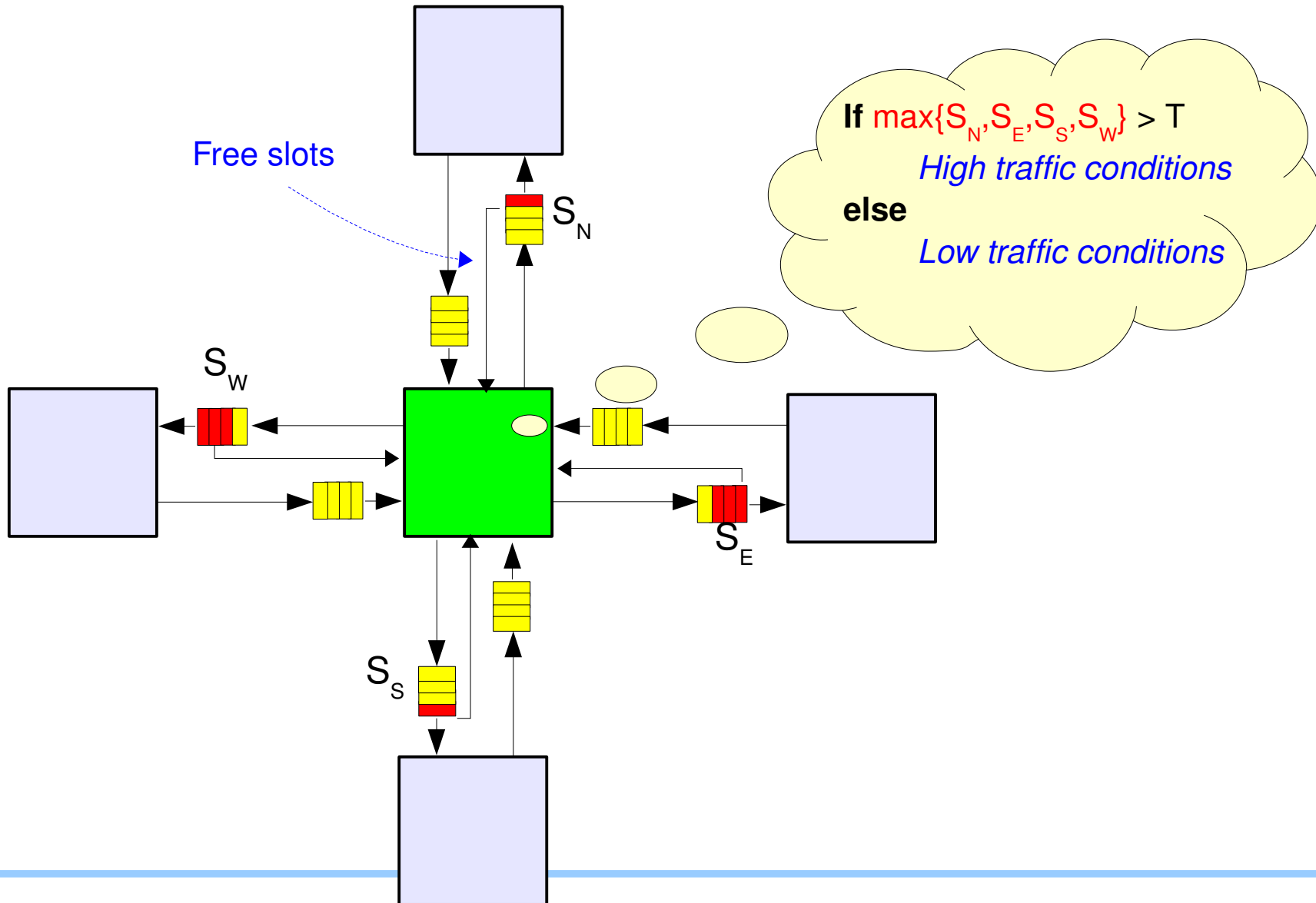
→ A link is selected with a probability inversely proportional to its FD



$$Pr^{(F)}(l_i) = \frac{1 - FD(l_i)}{1 - \sum_{l_j \in L_{ao}} FD(l_j)}$$

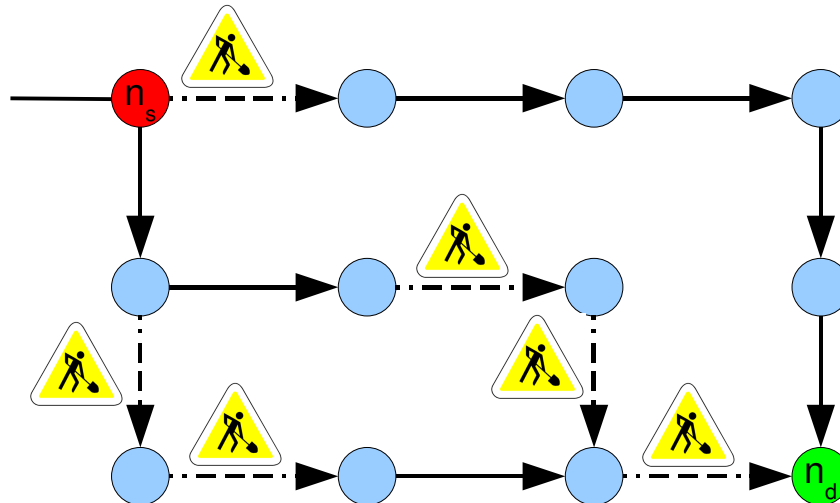
# Detecting Traffic Conditions

- Hu and Marculescu, *DyAD: Smart Routing for Networks-on-Chip*, DAC 2004



# Limitation of FU strategy

- The selection function used in FU, does not take into consideration the entire path
  - It takes the decision based solely on the quality of the next link
  - This can cause inefficiencies
    - ✓ *E.g.*, although the next link is of high-quality (*e.g.*, fault free), the rest of the path(s), in which the message will be obliged to travel on is/are formed by many low quality links

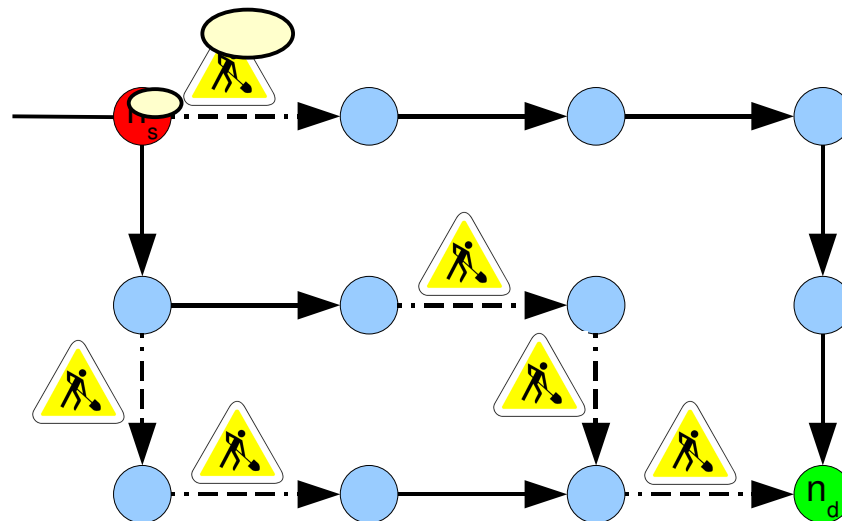
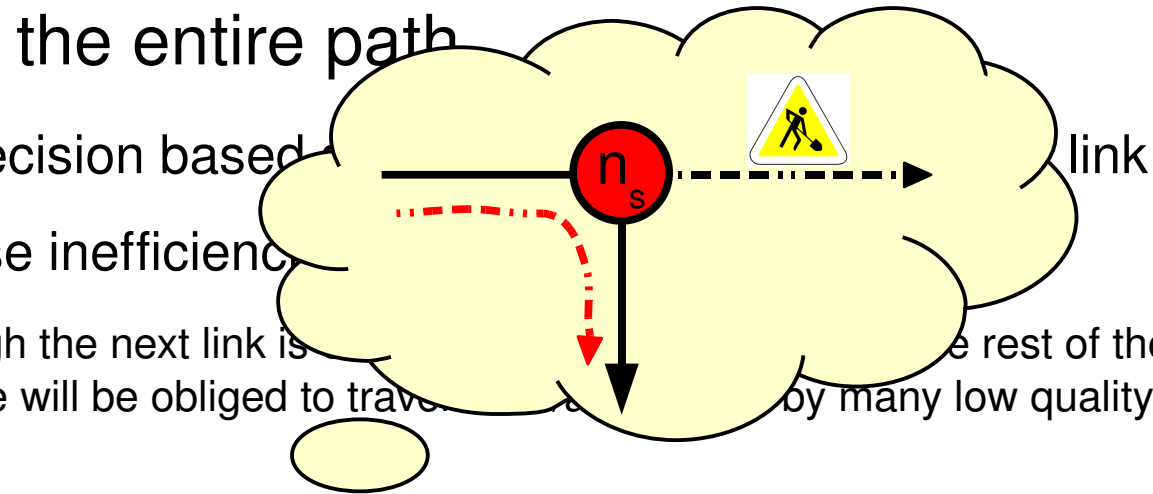


# FU with Look-ahead Strategy (FUL)

- The selection function used in FU, does not take into consideration the entire path

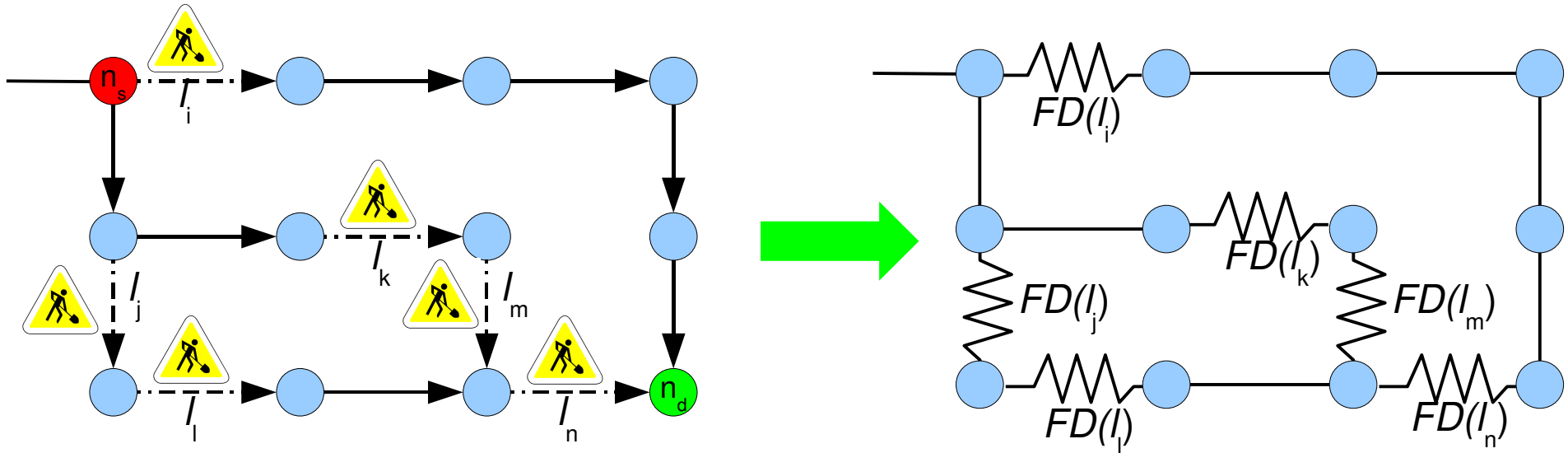
- It takes the decision based on the current link
- This can cause inefficiency

- ✓ E.g., although the next link is good, the rest of the path(s), in which the message will be obliged to travel, is made up of many low quality links



# FU with Look-ahead Strategy (FUL)

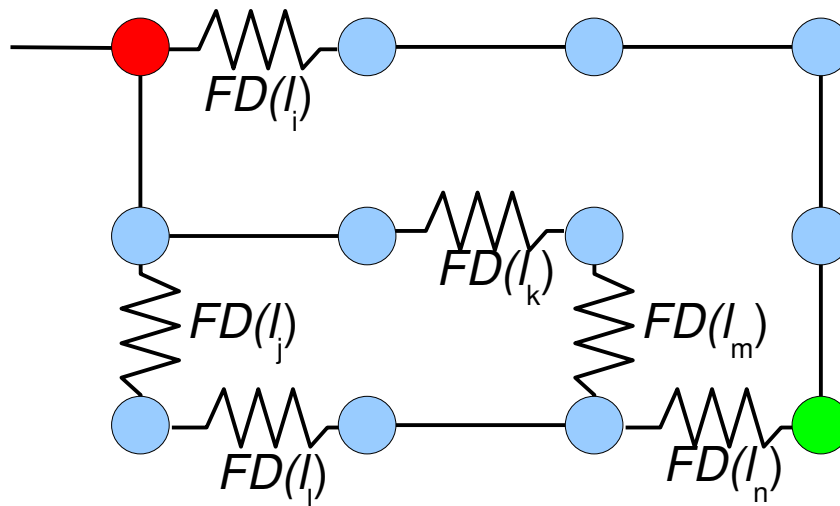
- The selection function used in FU, does not take into consideration the entire path
  - ➔ It takes the decision based solely on the quality of the next link
  - ➔ This can cause inefficiencies
    - ✓ *E.g.*, although the next link is of high-quality (*e.g.*, fault free), the rest of the path(s), in which the message will be obliged to travel on is/are formed by many low quality links



# FU with Look-ahead Strategy (FUL)

*Routing Table*

	AOs				ERM			
dst	N	E	S	W	N	E	S	W
●	0	1	1	0	0	1	0	0
...	...				...			



Set if the equivalent resistance of paths having as first link **E** and ending at node ● is minimum

# FU with Look-ahead Strategy (FUL)

---

## ■ In *low traffic* conditions

→ One of the *AO* links  $l_i$  such that the  $i$ -th bit of ERM is set, is *randomly chosen*

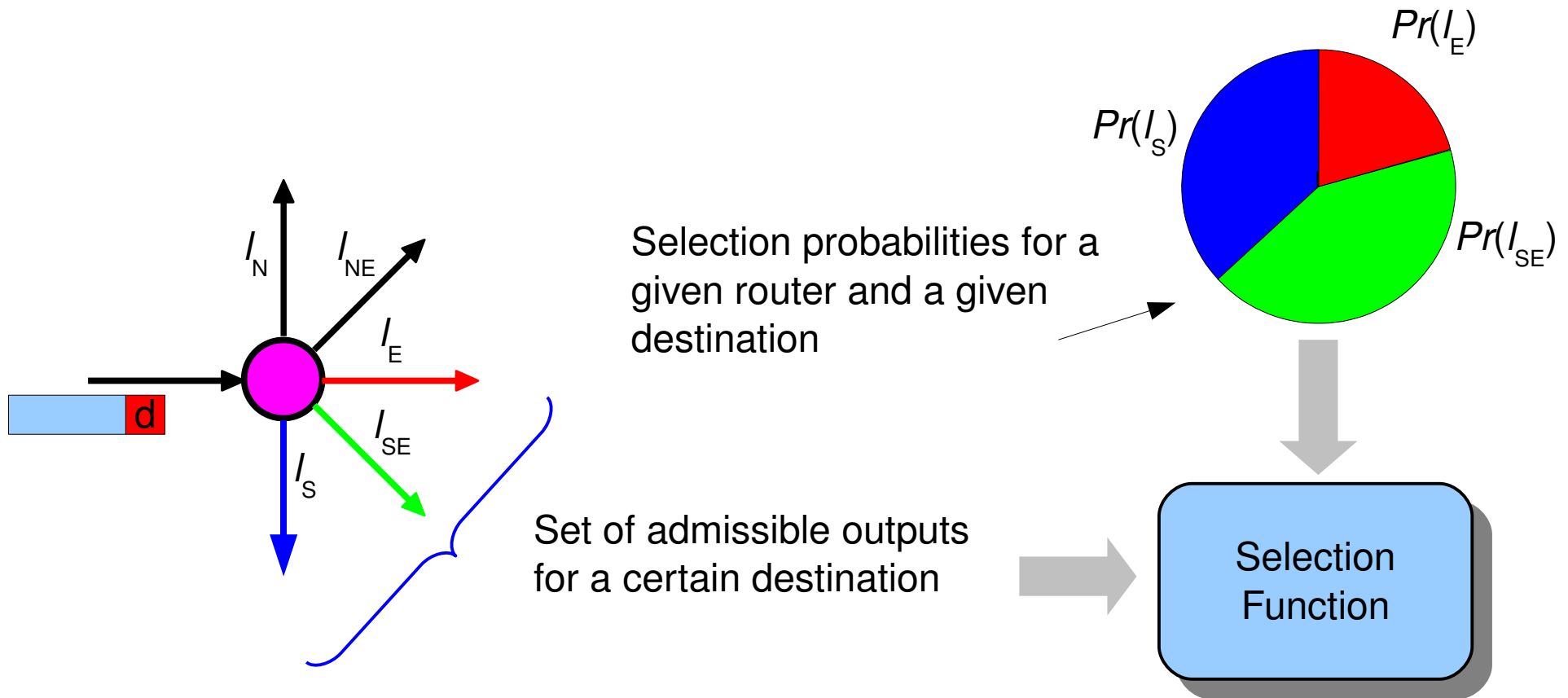
✓ If multiple faulty free paths exist, they are always used

## ■ In *high traffic* conditions

→ The same as FU

# Load Balancing Technique

- Determining the best set of *selection probabilities* which allow to *optimally distributing* the traffic over the network



# Load Balancing Technique

---

## ■ Traffic Function $TF(I,X)$

→ Returns an **estimation of the traffic load** on network link  $l$  when the set of selection probabilities  $X$  is used

## ■ Goal

$$\min_{X} \text{var}_{l \in L} TF(I,X)$$

→ Such that  $X$  is a **feasible** set of selection probabilities  
( $x_i \in [0,1]$ ,  $\sum_{l_i, d, lo} x_i = 1$ )

→ Sequential Quadratic Programming Optimization

# Load Balancing based Strategies

## ■ Fault Elimination + LB (**FE+LB**)

→ Selection function based on the selection probabilities computed resolving the LB problem ( $Pr^{(LB)}$ )

## ■ Partially Faulty Links usage strategy + LB (**FU+LB**)

→ In *low traffic* conditions

✓ Selection function uses  $Pr^{(LB)}$

In *high traffic* conditions

$$Pr^{(FLB)} = Pr^{(LB)} \sqrt{Pr^{(F)}}$$

## ■ Partially Faulty Links usage with Look-ahead strategy + LB (**FUL+LB**)

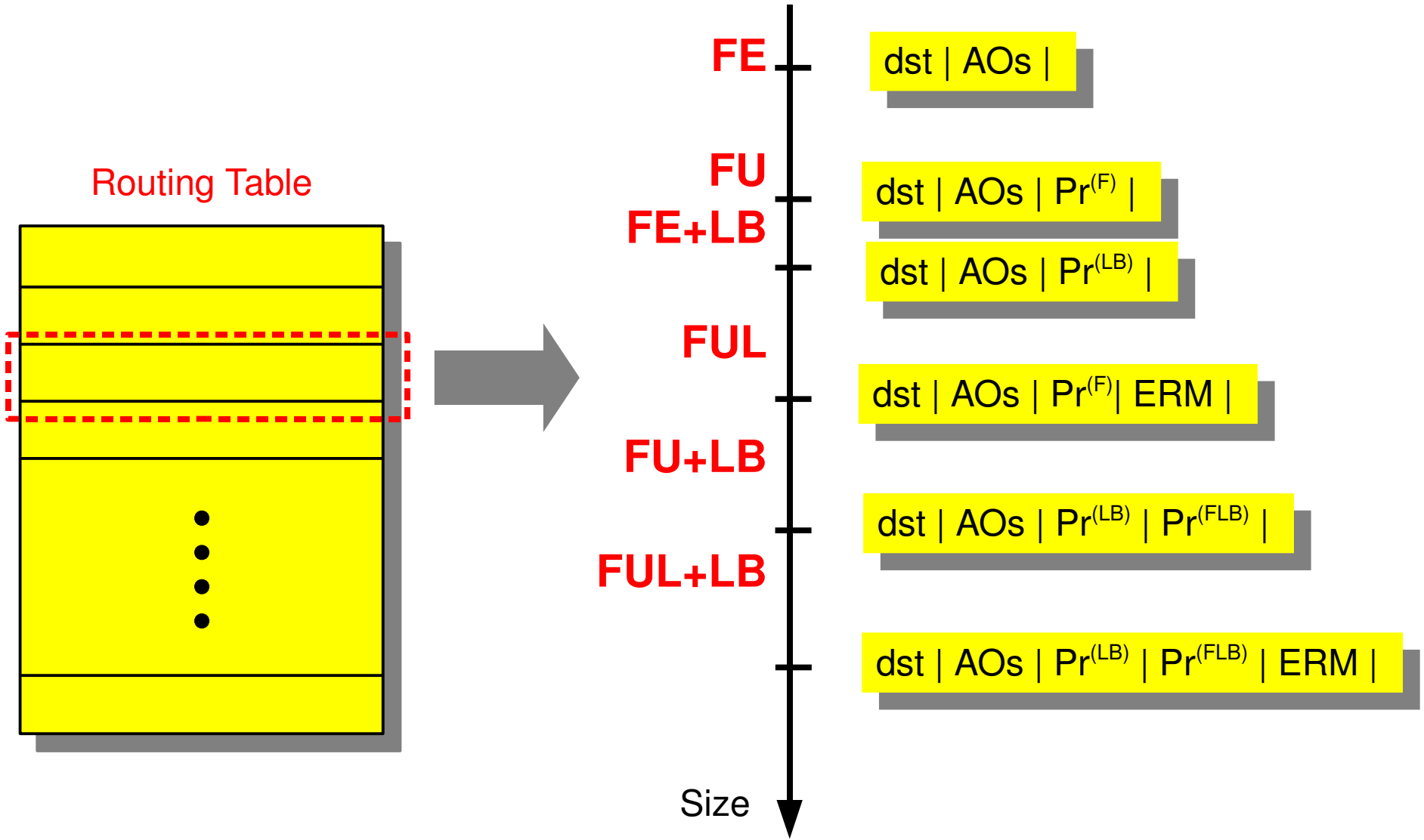
→ In *low traffic* conditions

✓ Selection function uses  $Pr^{(LB)}$   
between AOs where ERM is set

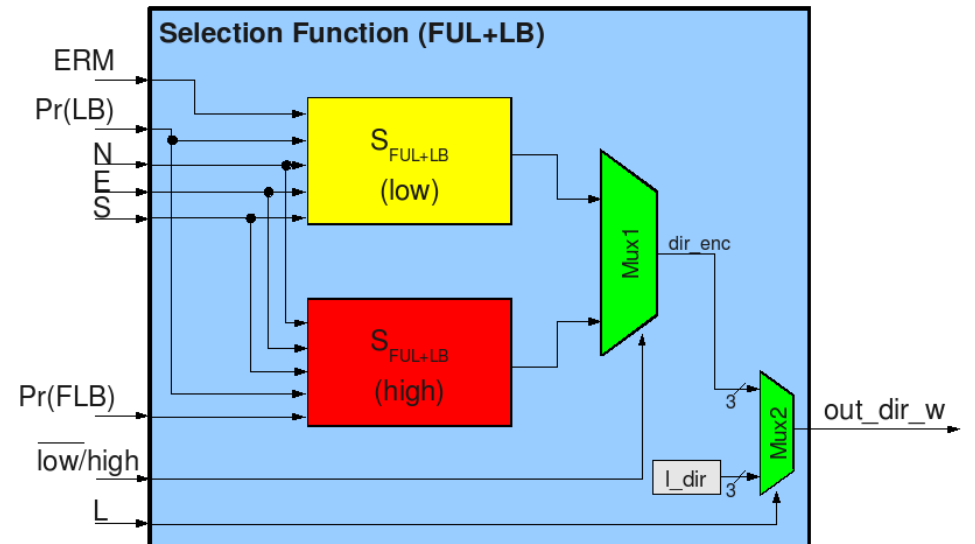
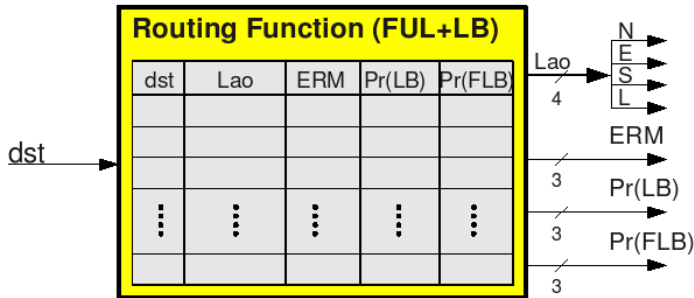
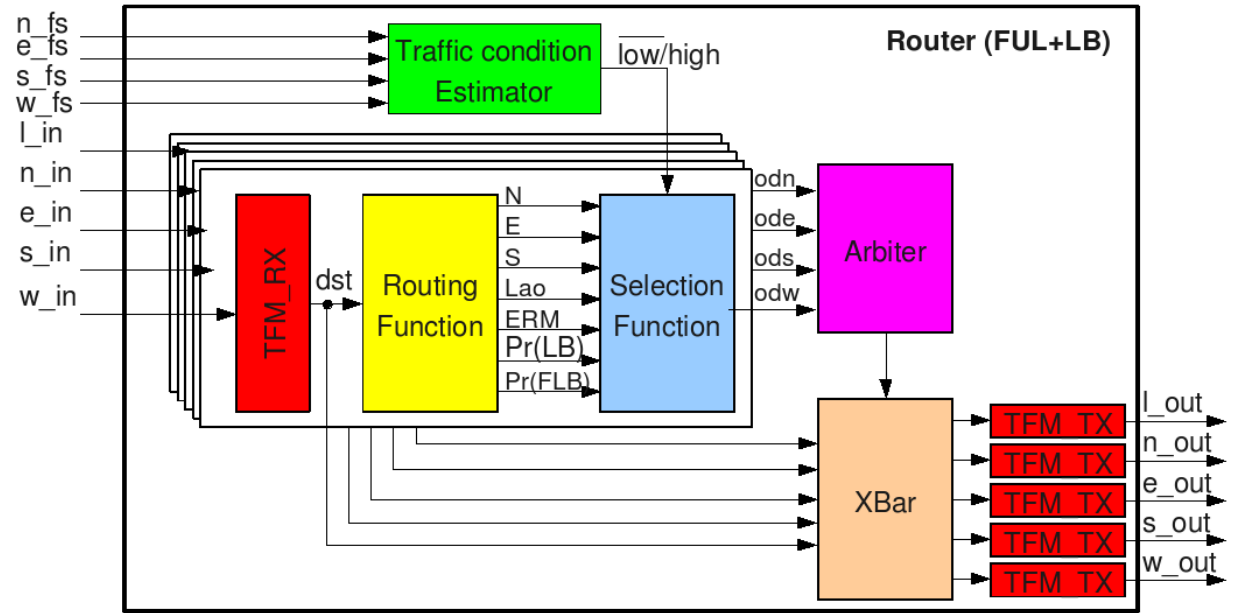
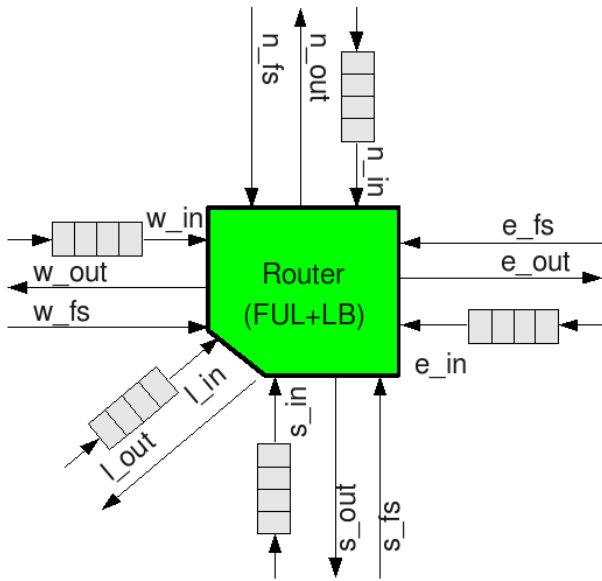
In *high traffic* conditions

Like FUL+LB

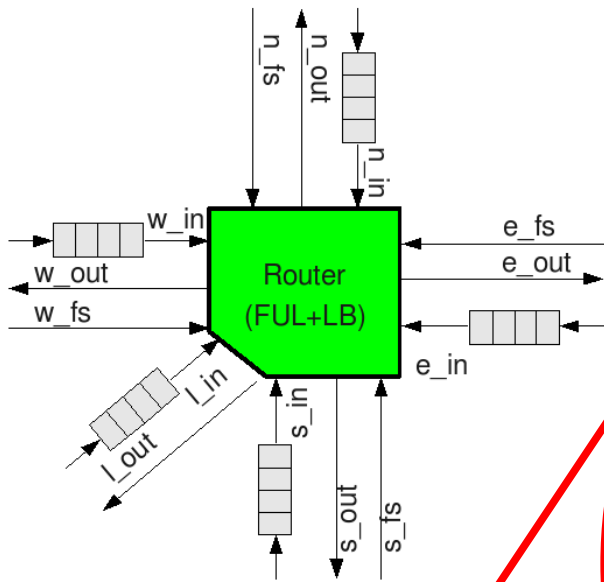
# Entry in Routing Table



# Router Architecture (FUL+LB)



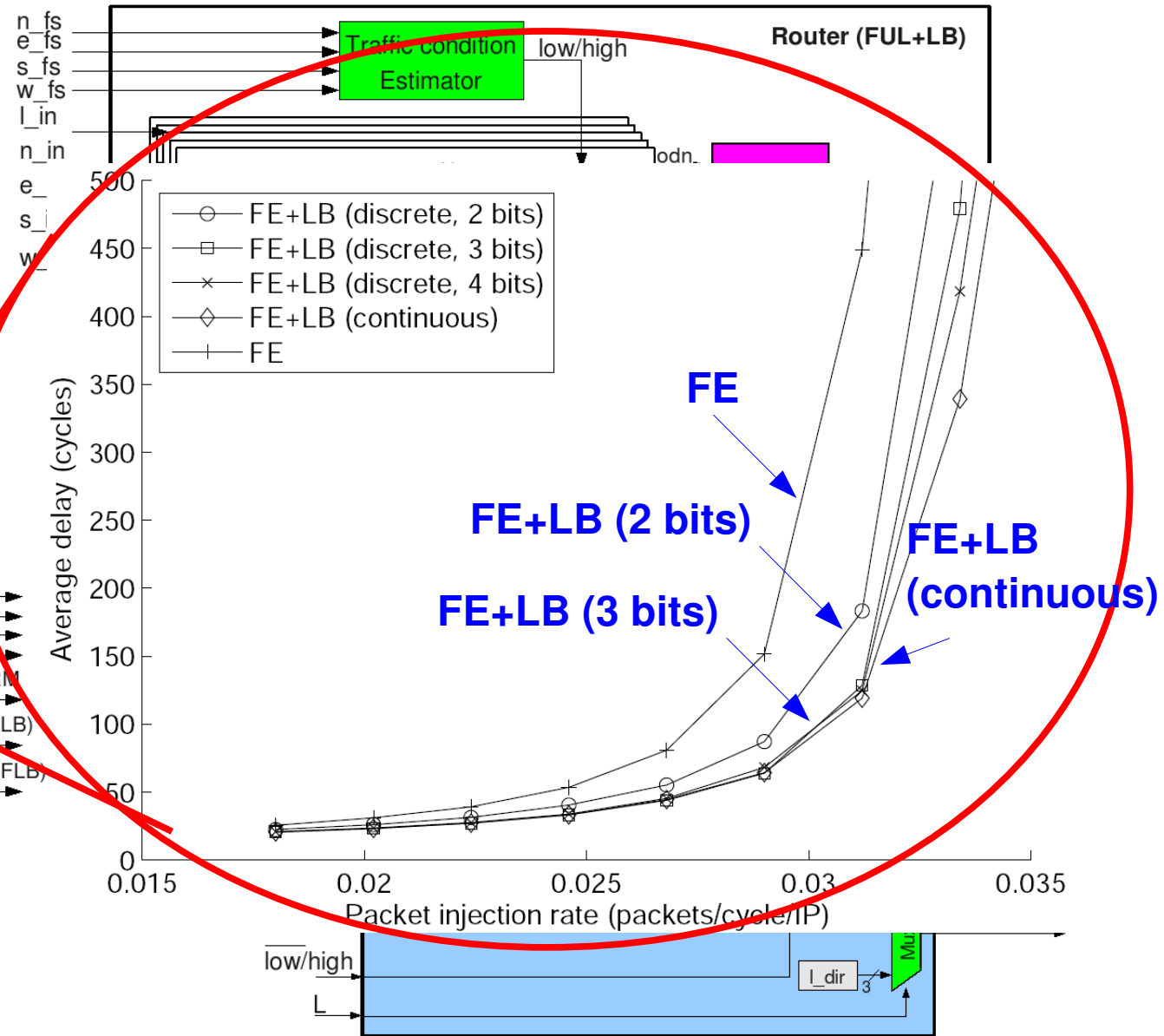
# Router Architecture (FUL+LB)



**Routing Function (FUL+LB)**

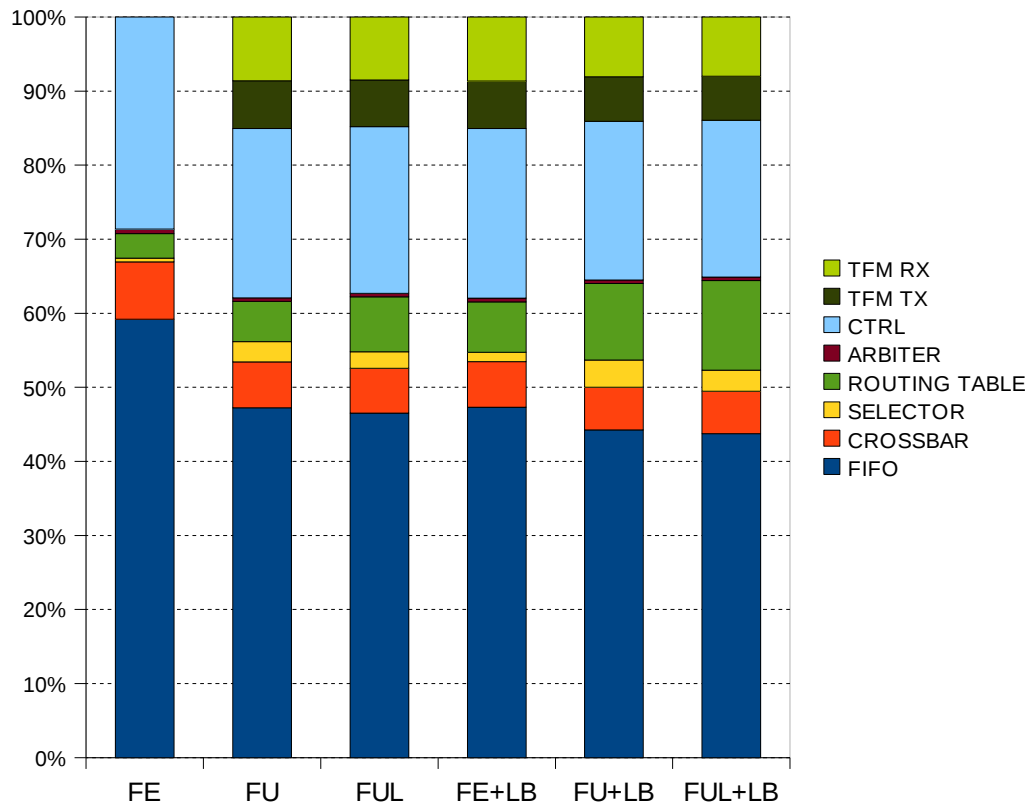
dst	Lao	ERM	Pr(LB)	Pr(FLB)

Inputs: dst, Lao, ERM, Pr(LB), Pr(FLB).  
Outputs: Lao, ERM, Pr(LB), Pr(FLB).

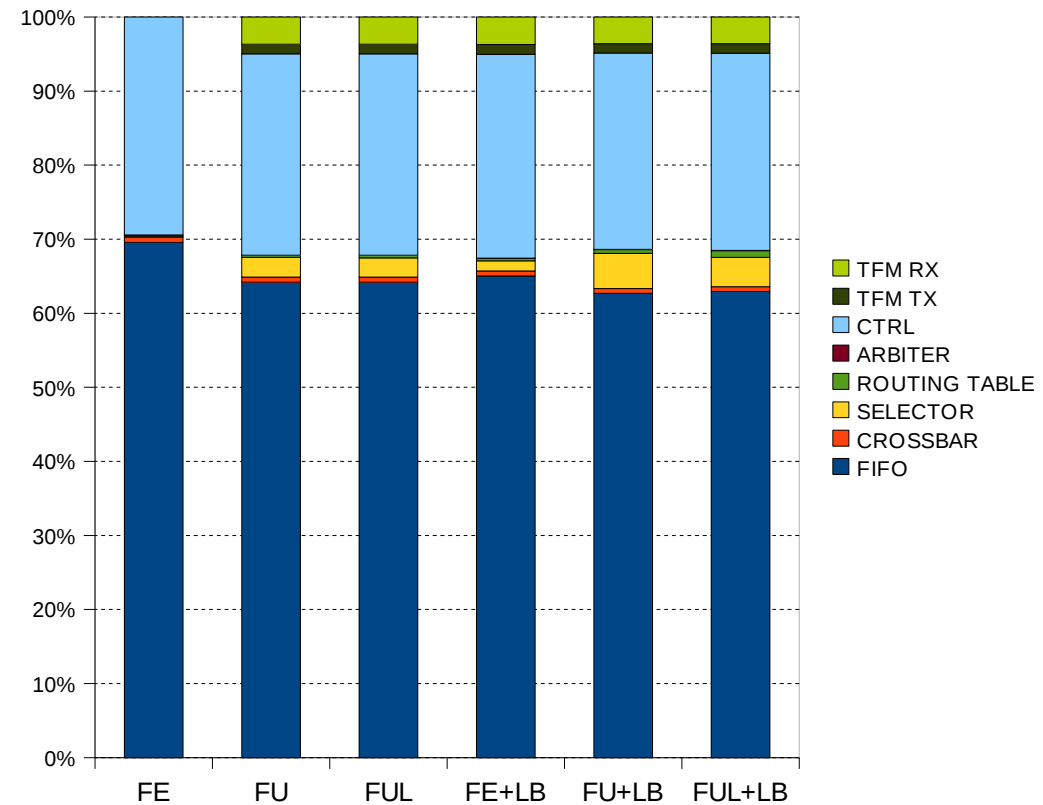


# Implication on Router Design

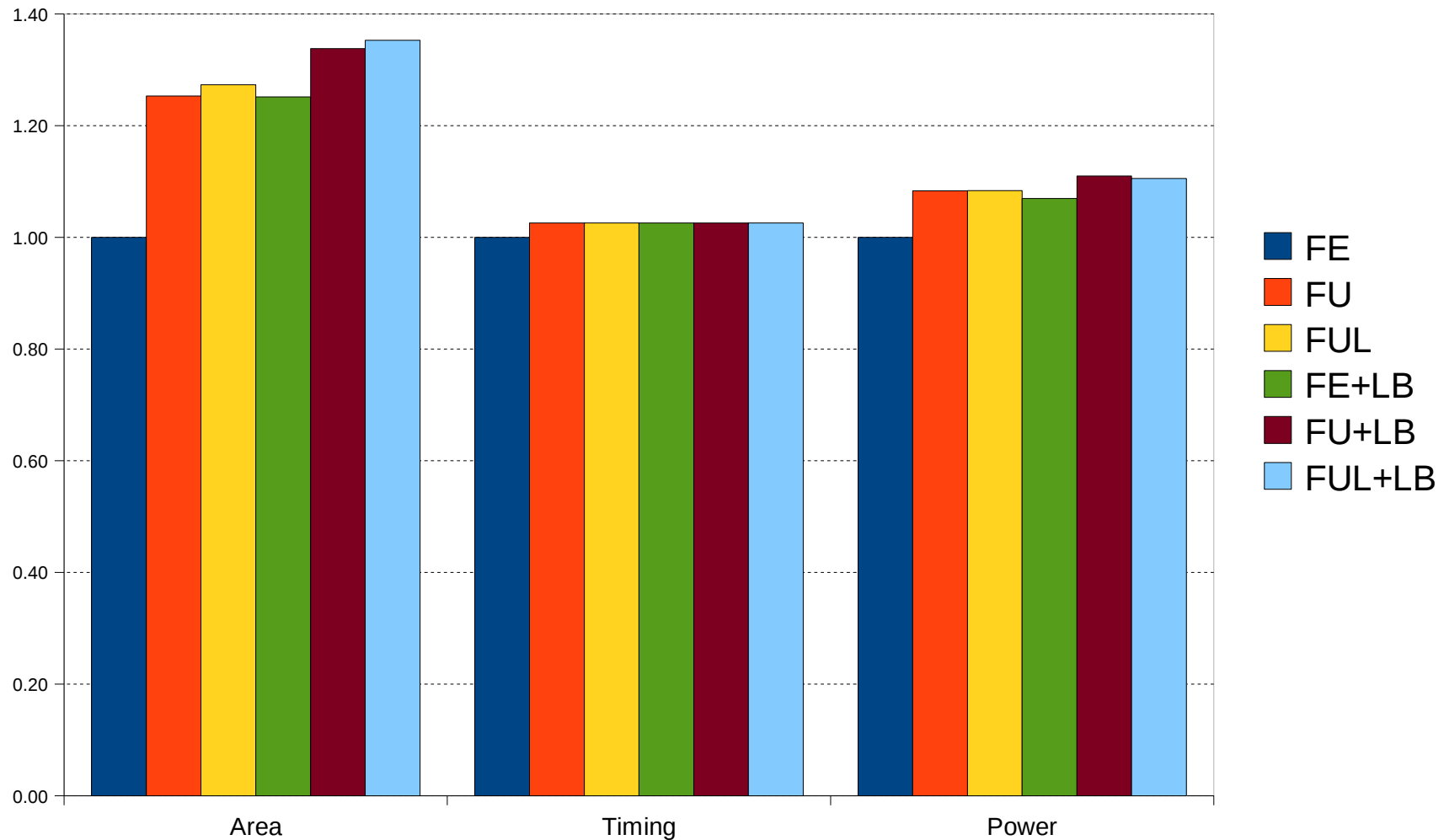
Area Breakdown



Power Breakdown

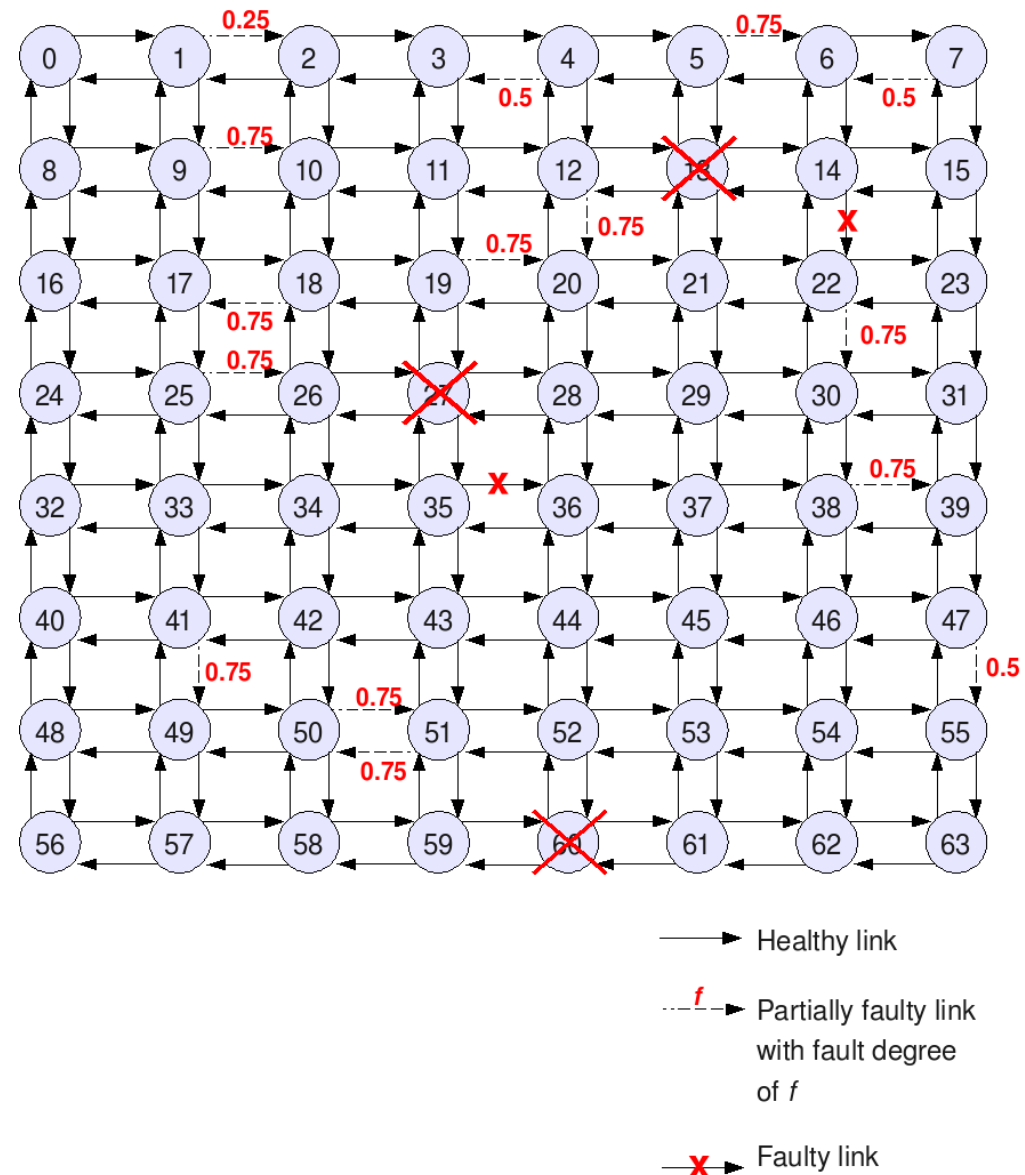


# Implication on Router Design



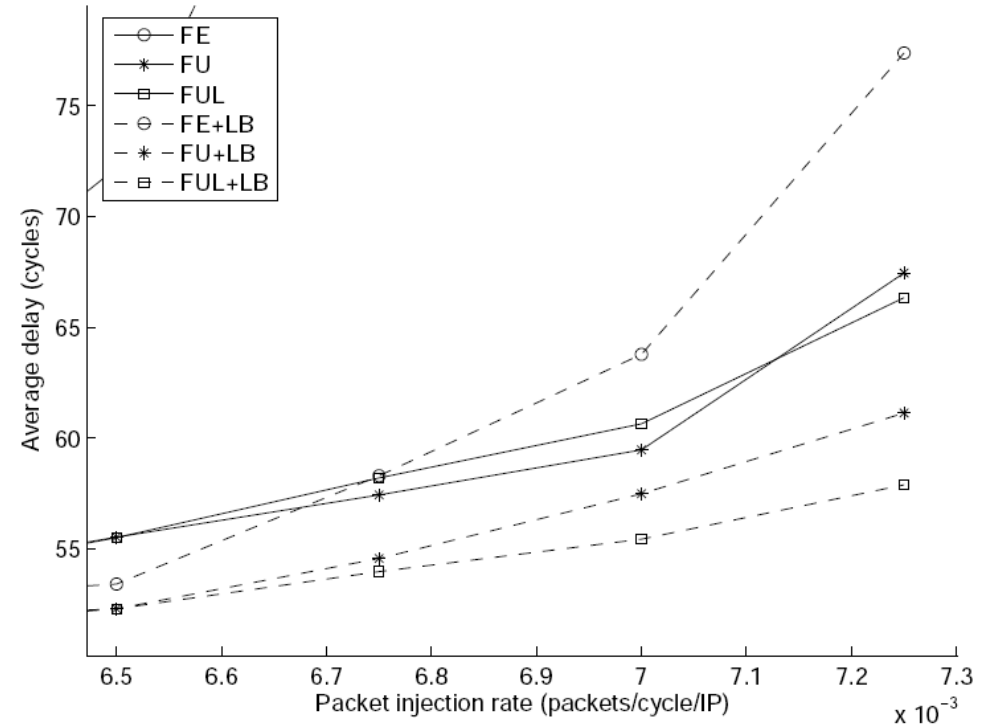
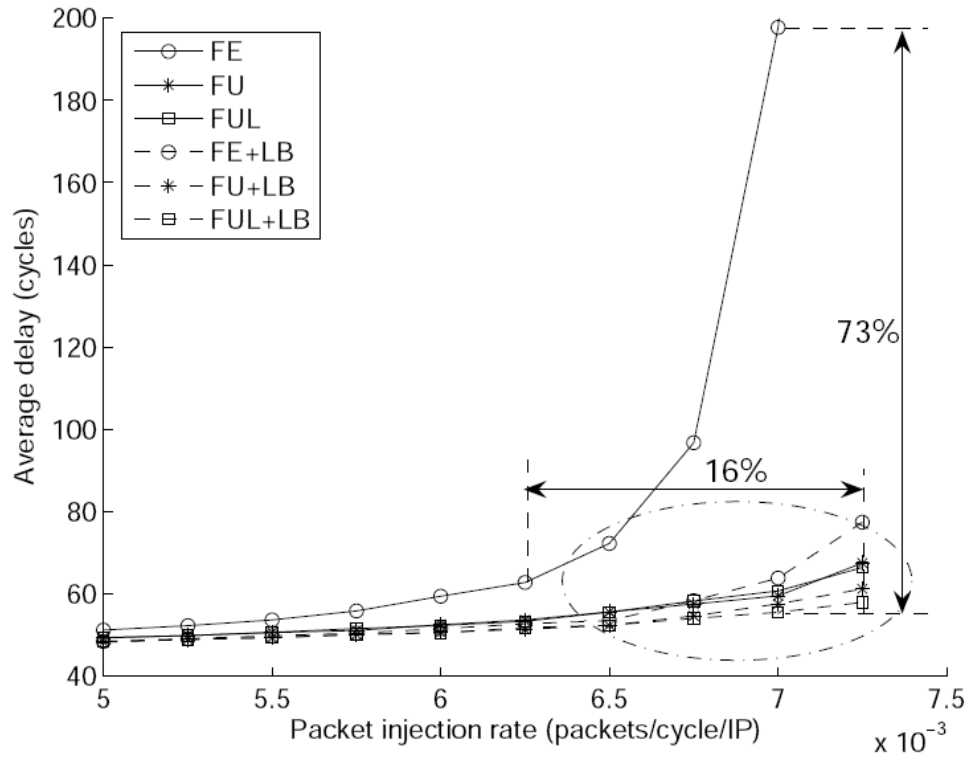
# Performance Evaluation

- Simulation platform: *Noxim*
- Traffic scenarios: Uniform, transpose, bit-reversal, butterfly, shuffle
- 8x8 mesh-based NoC architecture
  - ➔ Buffers: 4 flits
  - ➔ Packet size: random 2-16 flits
  - ➔ Poisson packet injection distribution
  - ➔ Simulation time: 100,000 clock cycles (warm-up session: 10,000 clock cycles)



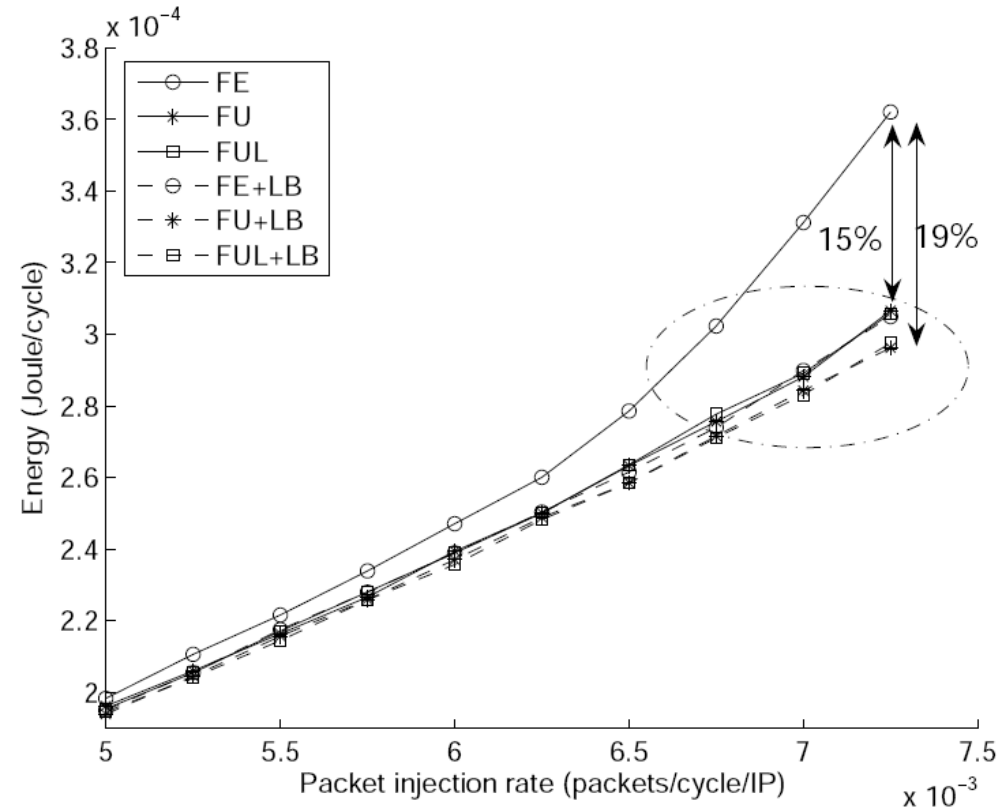
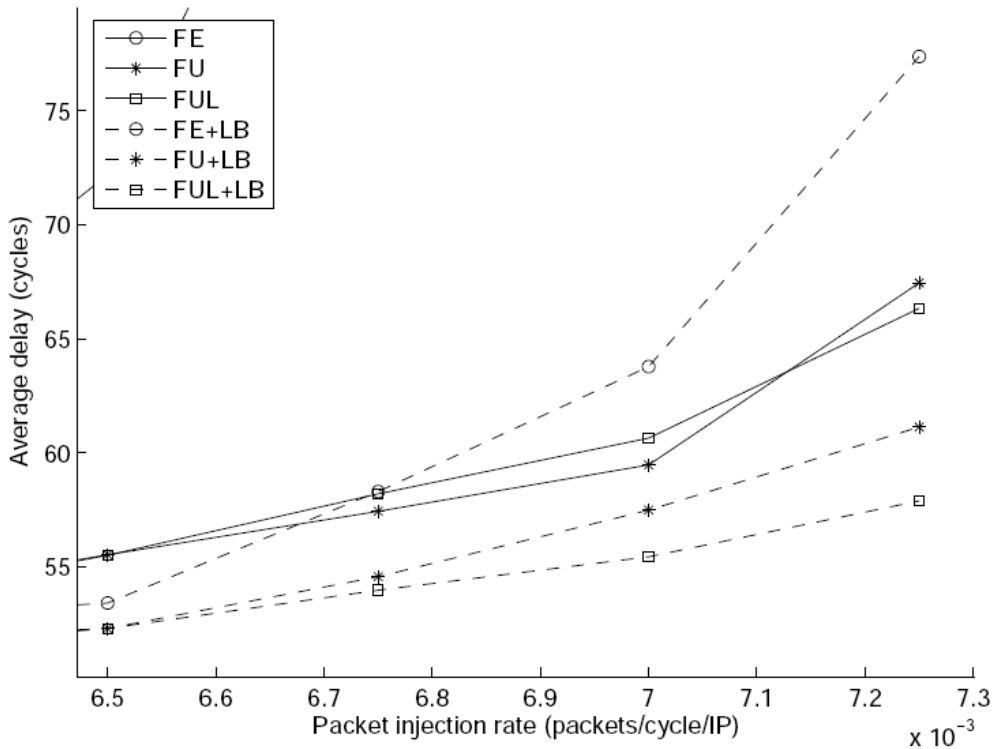
# Delay Variation

Delay variation under *uniform* traffic



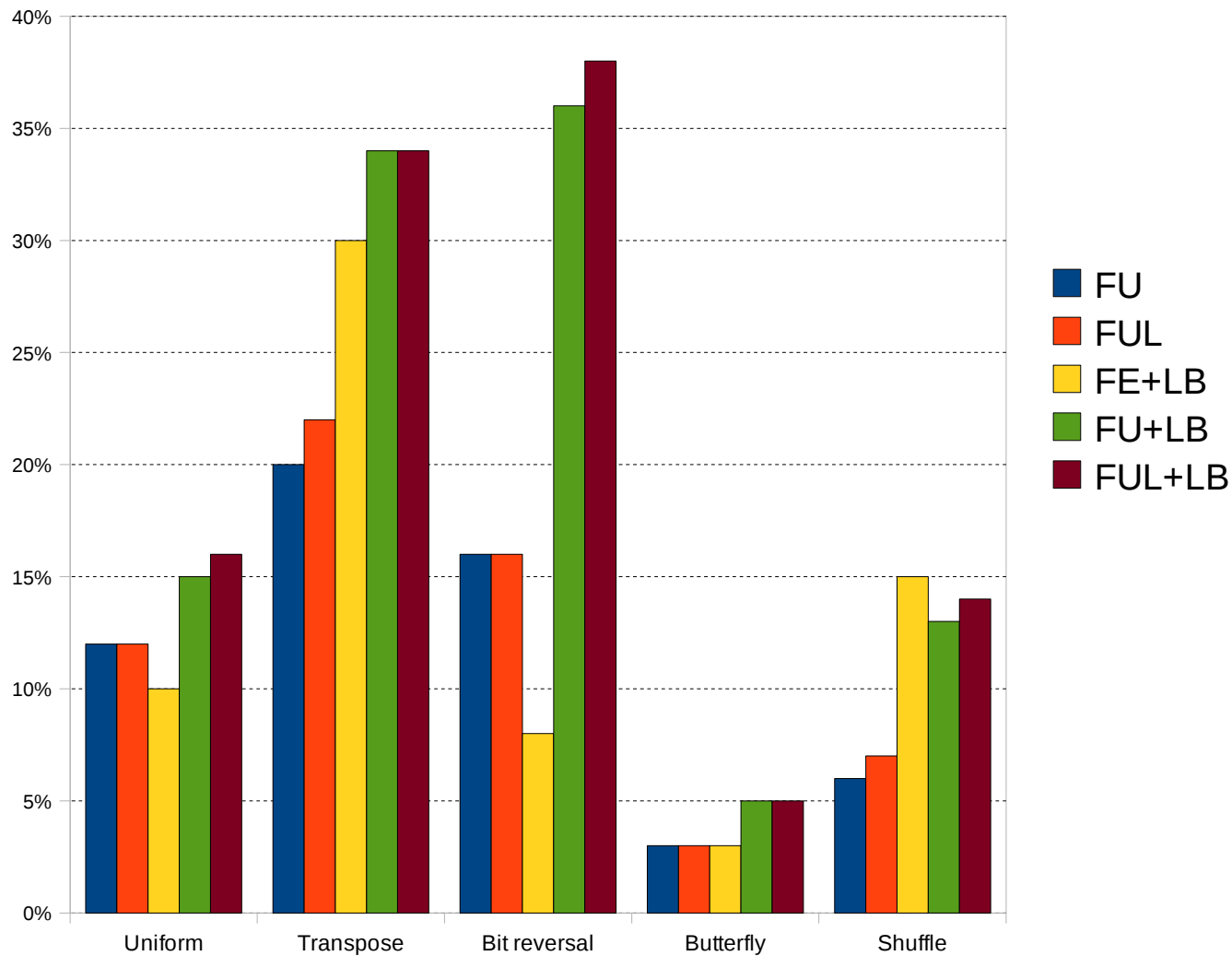
# Energy Variation

Energy variation under *uniform* traffic



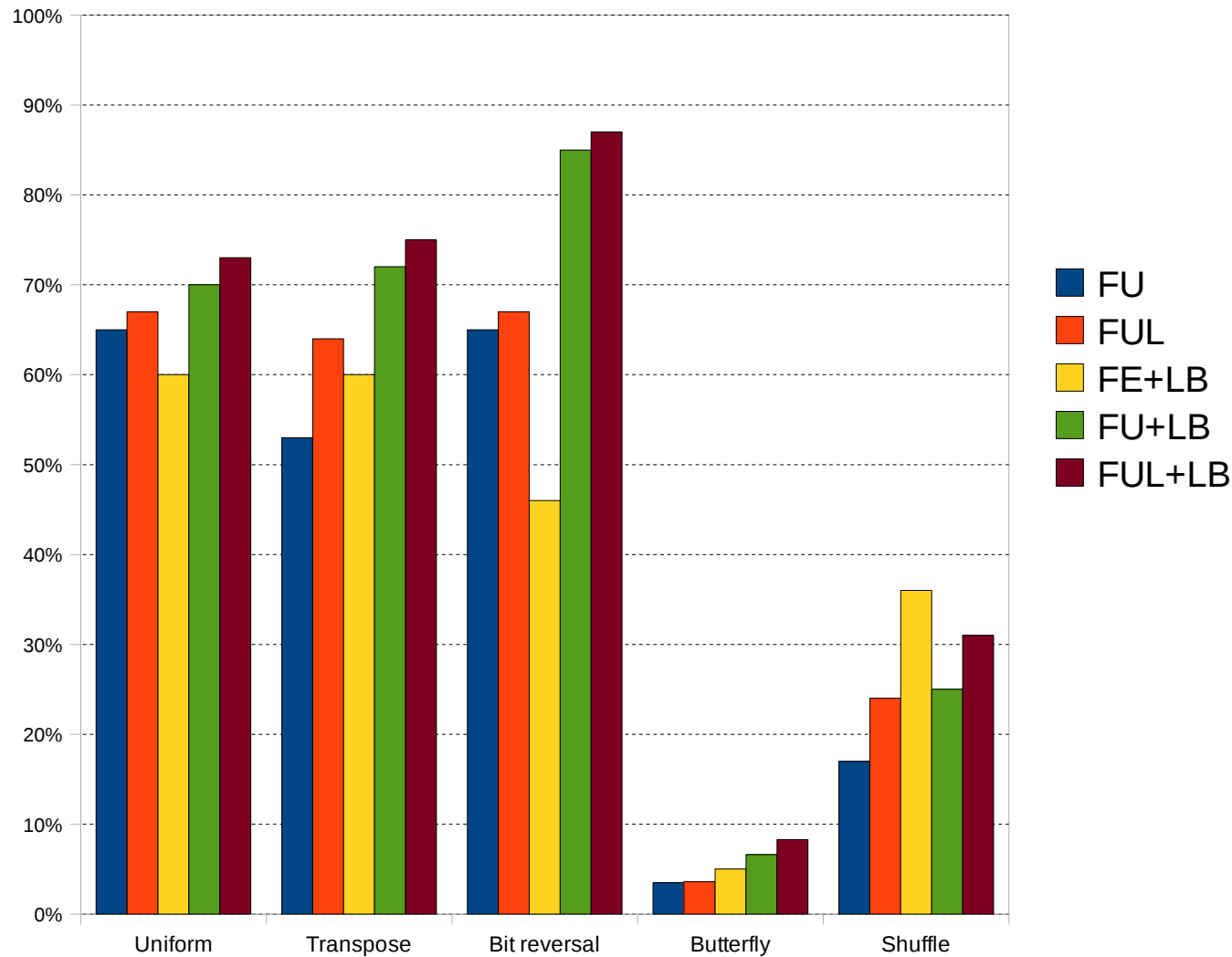
# Improvement in Saturation *pir*

Increase in saturation *pir*



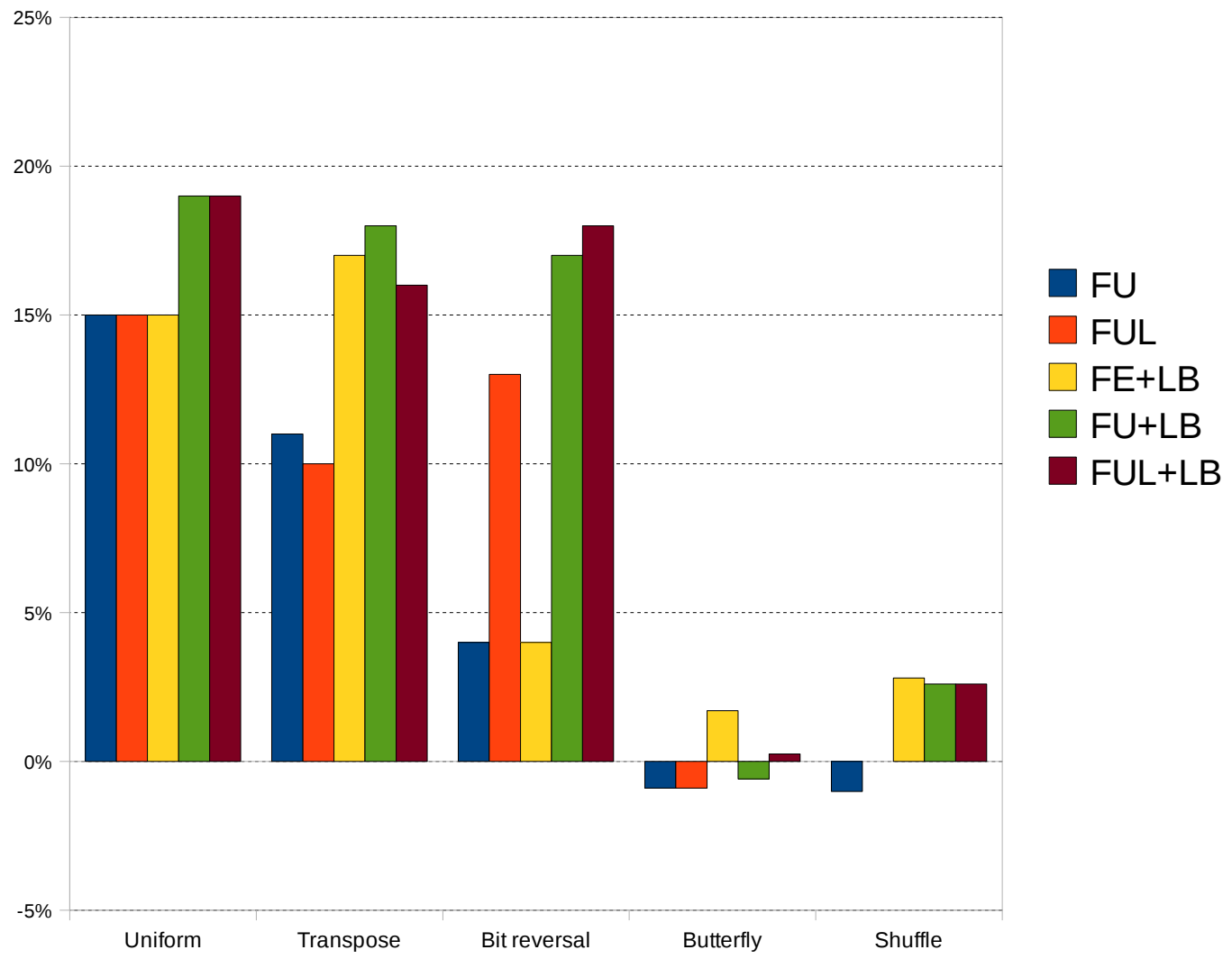
# Improvement in Average Delay

Reduction in average delay



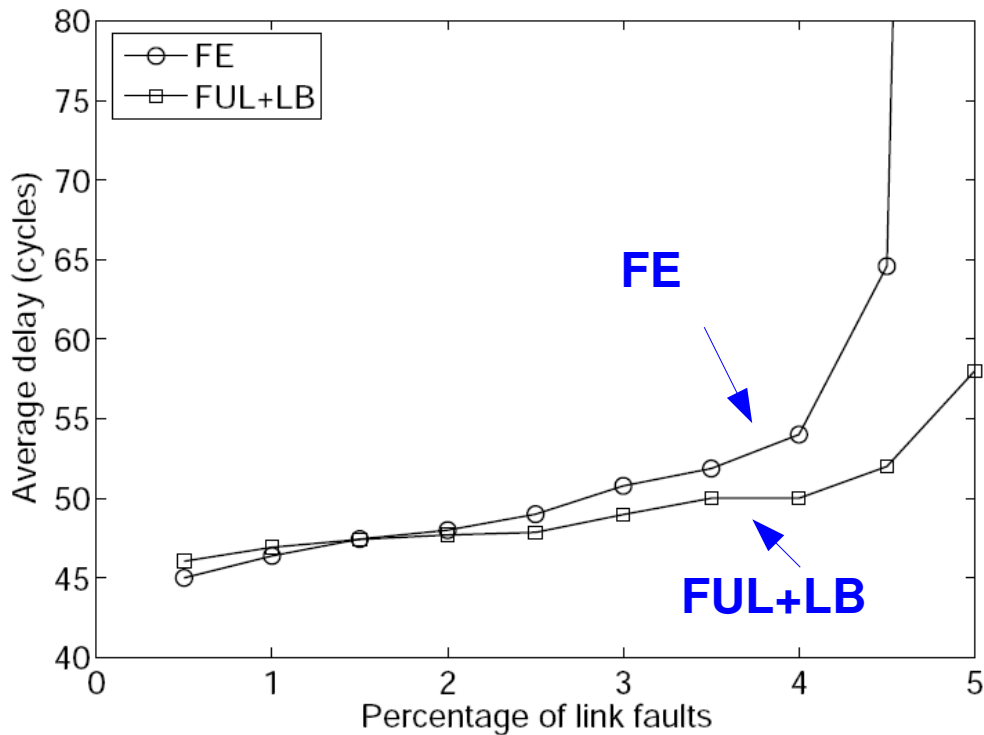
# Reduction in Energy

Reduction in energy

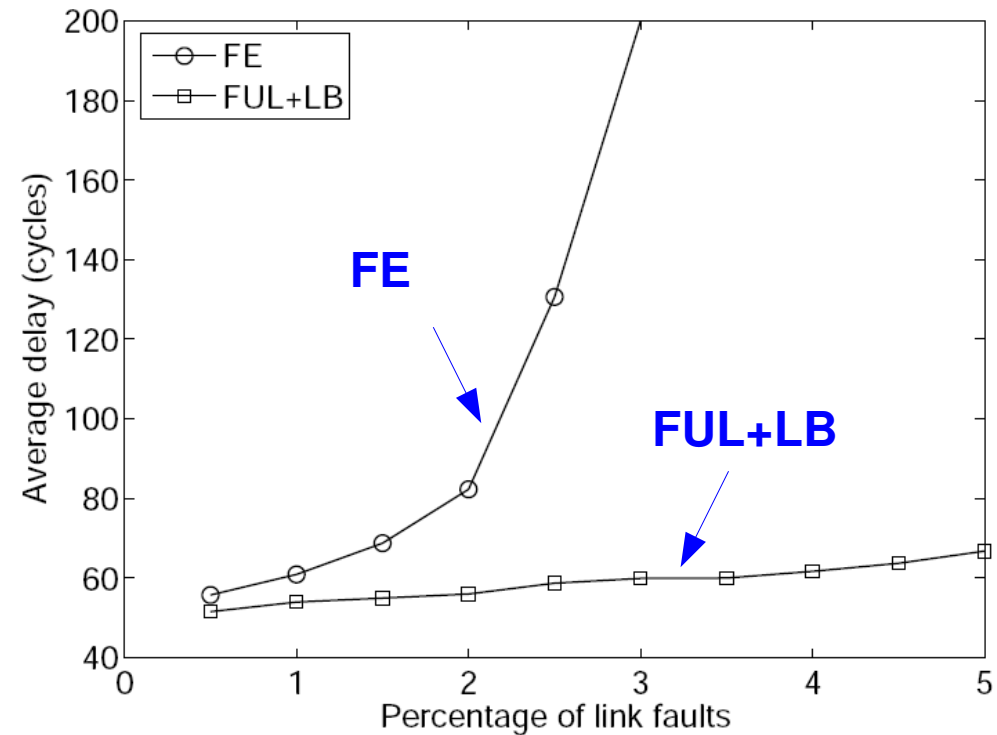


# Delay vs. Percentage of Link Faults

Faults random distributed



Faults random clustered



# Outline

---

- Application Specific Routing Algorithms ✓
- Concurrent Mapping and Routing ✓
- Dealing with Manufacturing Defects ✓
- Encoding Scheme for Low Power

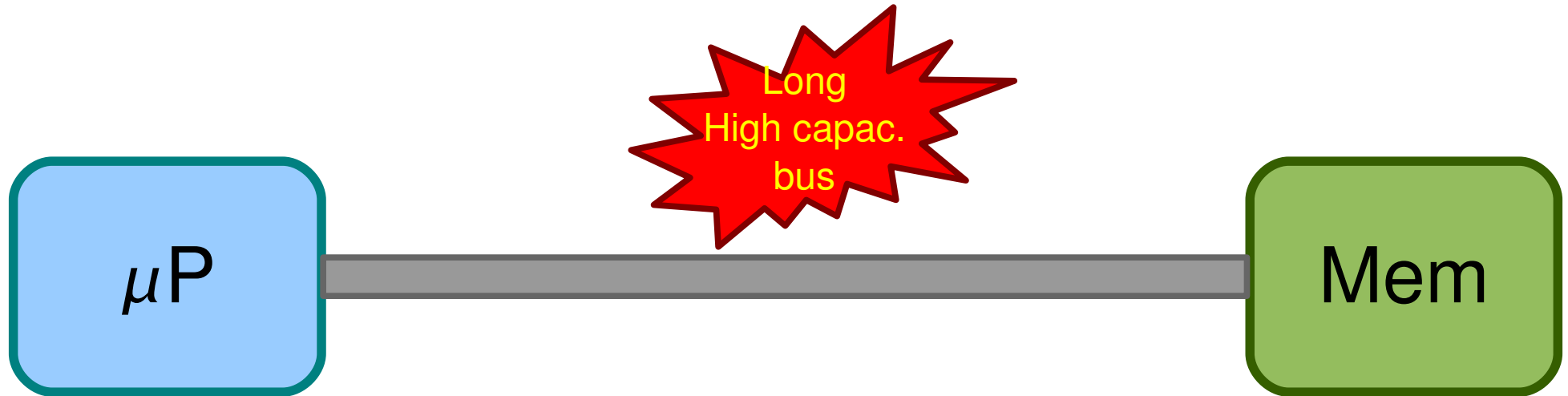
# Motivations

---

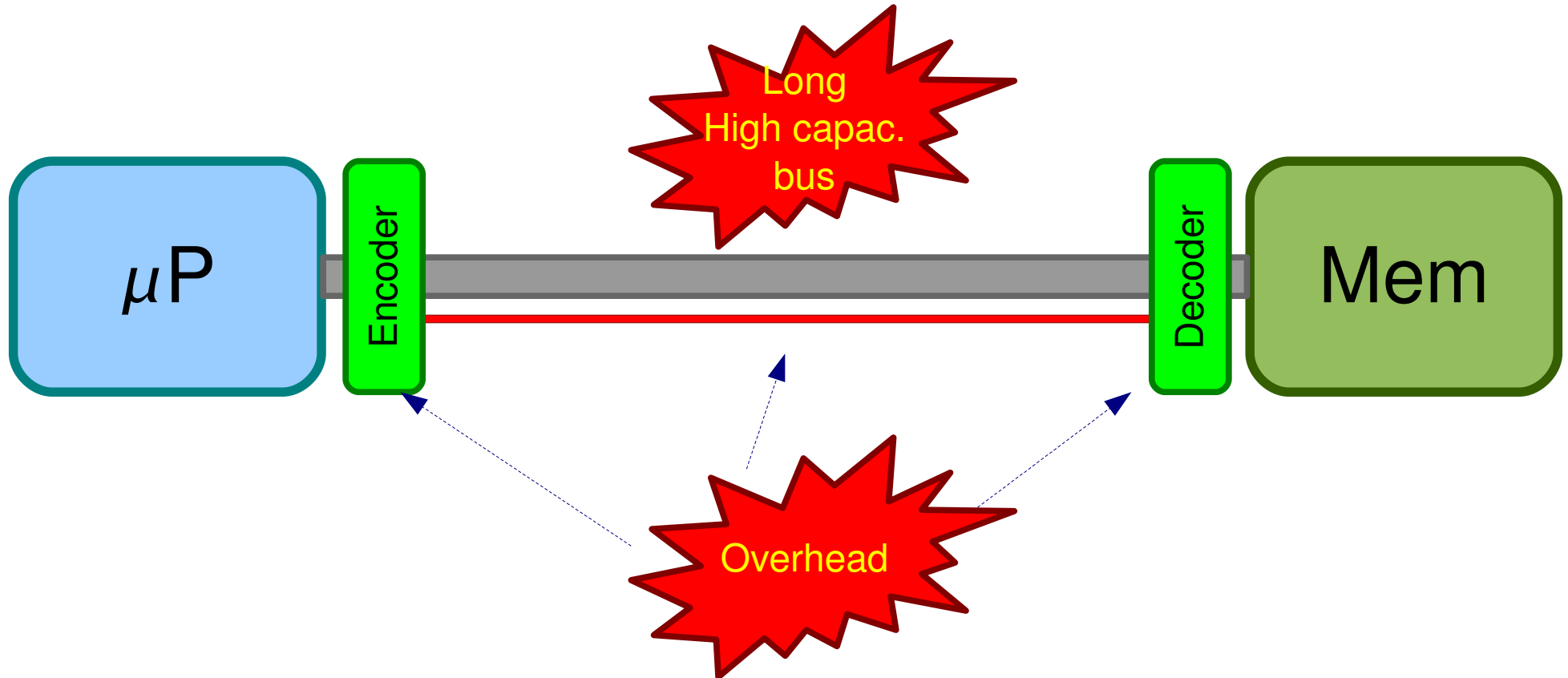
- The interconnect system is one of the main elements which characterizes the system in terms of both power dissipation and energy consumption
  - Approx 28% in the Intel's 80-tiles TeraFLOPS
- As technology shrinks, the power ratio between NoC links and routers increases
  - Links are becoming more power hungry than routers

# General Scheme – bus-based Archs

---



# General Scheme – bus-based Archs

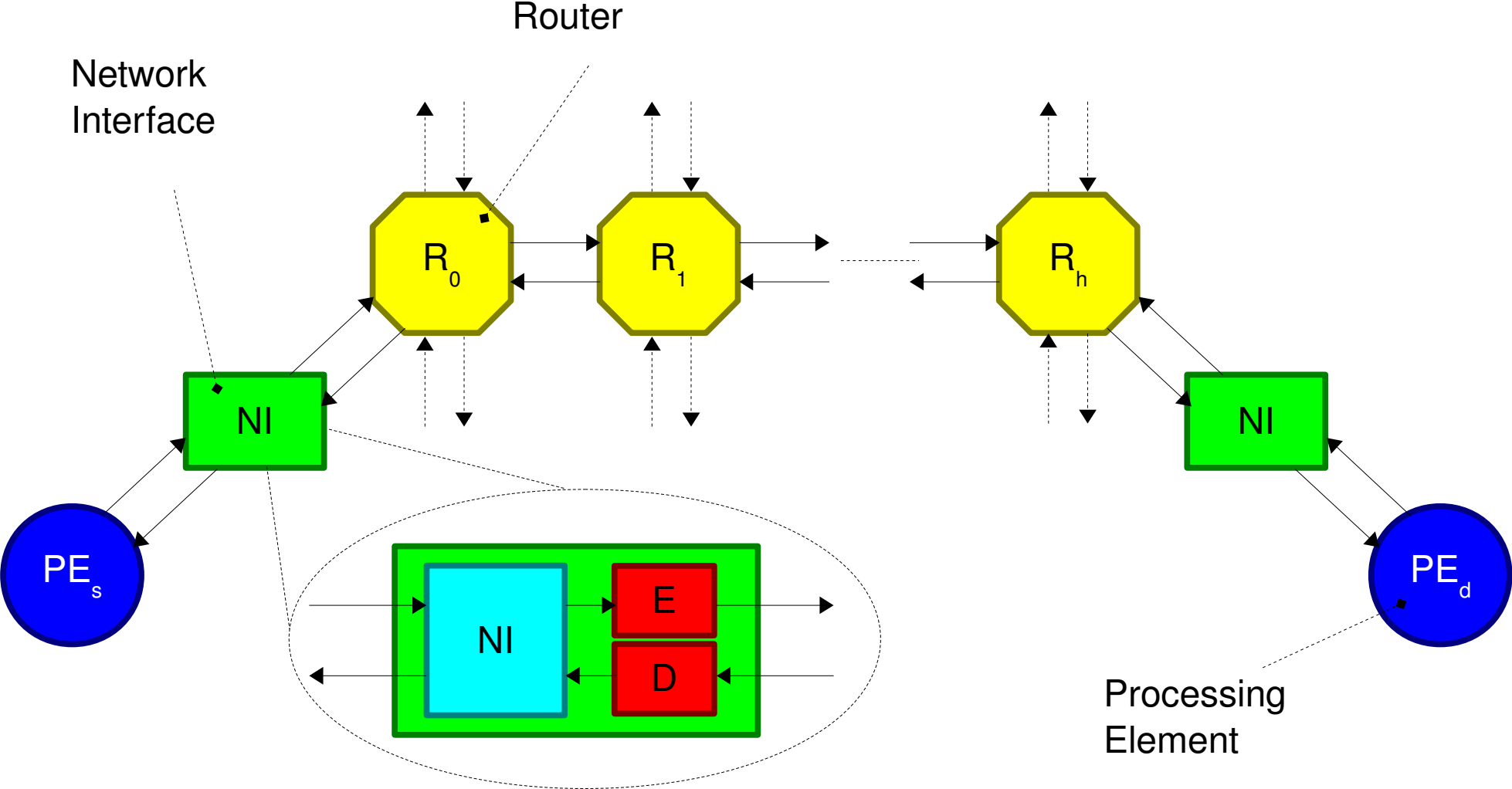


# Encoding Schemes

---

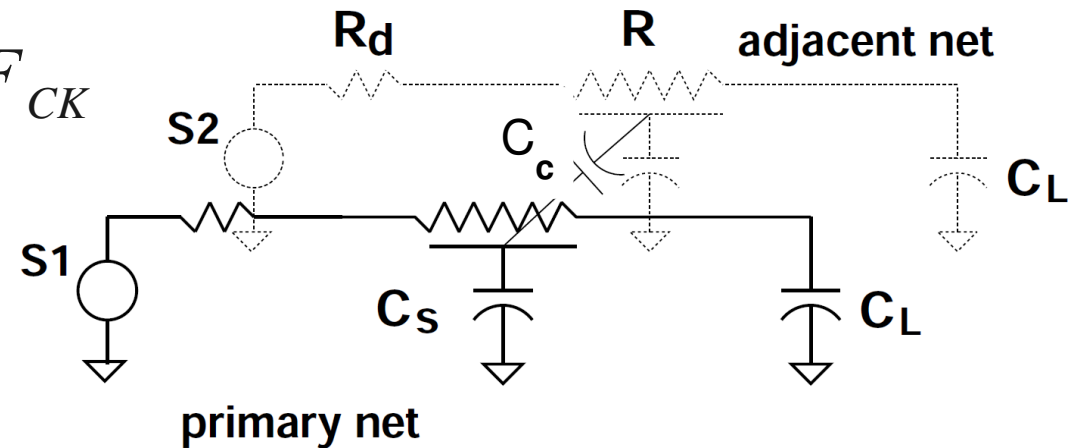
- Applied in the context of bus-based architectures
- Bus-invert method [Stan and Burleson, TVLSI'95]
- Gray code [Su *et al.*, D&T'94]
- T0 method [Benini *et al.*, GLS-VLSI'97]
- Working-zone encoding [Mussoll *et al.*, PDAW'98]
- ...
- Do not take into account coupling effects
  - Dominant in DSM regime

# General Scheme – NoC Archs

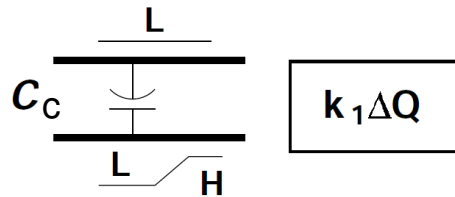


# Power Model

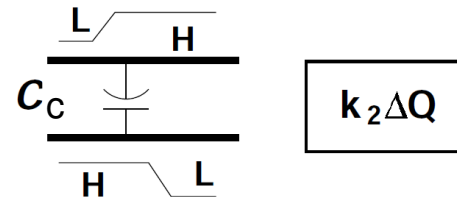
$$P = [T_{0 \rightarrow 1} (C_s + C_l) + T_c C_c] V_{dd}^2 F_{CK}$$



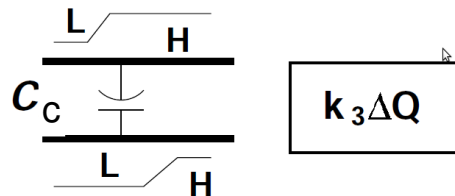
$$T_c = k_1 T_1 + k_2 T_2 + k_3 T_3 + k_4 T_4$$



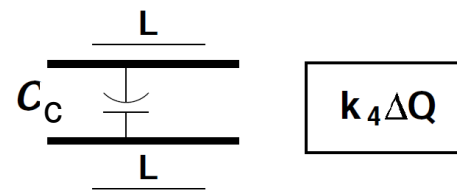
(a) Type I



(b) Type II



(c) Type III

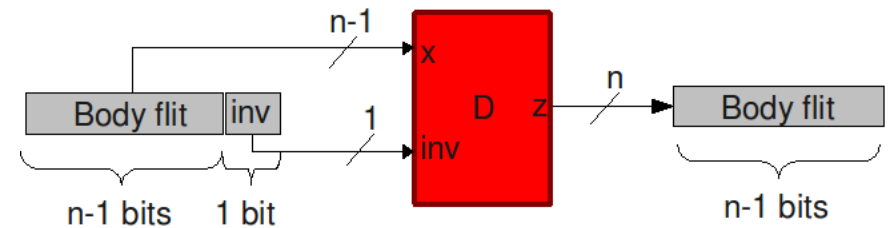
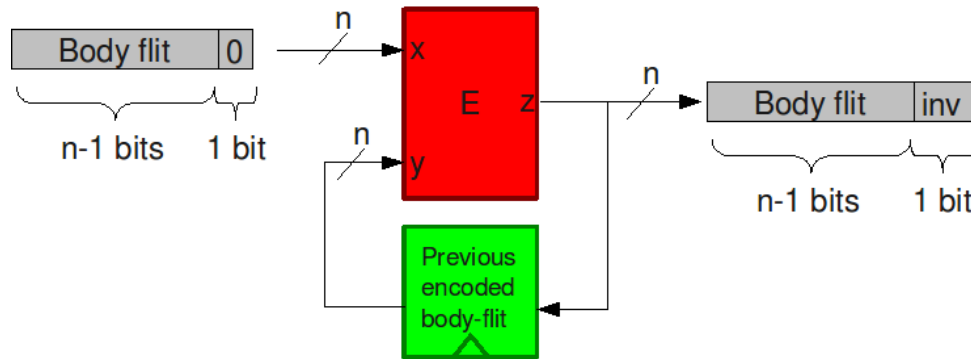


(d) Type IV

# Proposed Scheme (1/2)

$$P \propto T_{0 \rightarrow 1} C_s (k_1 T_1 + k_2 T_2 + k_3 T_3 + k_4 T_4) C_c$$

$$P' \propto T'_{0 \rightarrow 1} C_s (k_1 T'_1 + k_2 T'_2 + k_3 T'_3 + k_4 T'_4) C_c$$



■ Invert if  $P > P'$

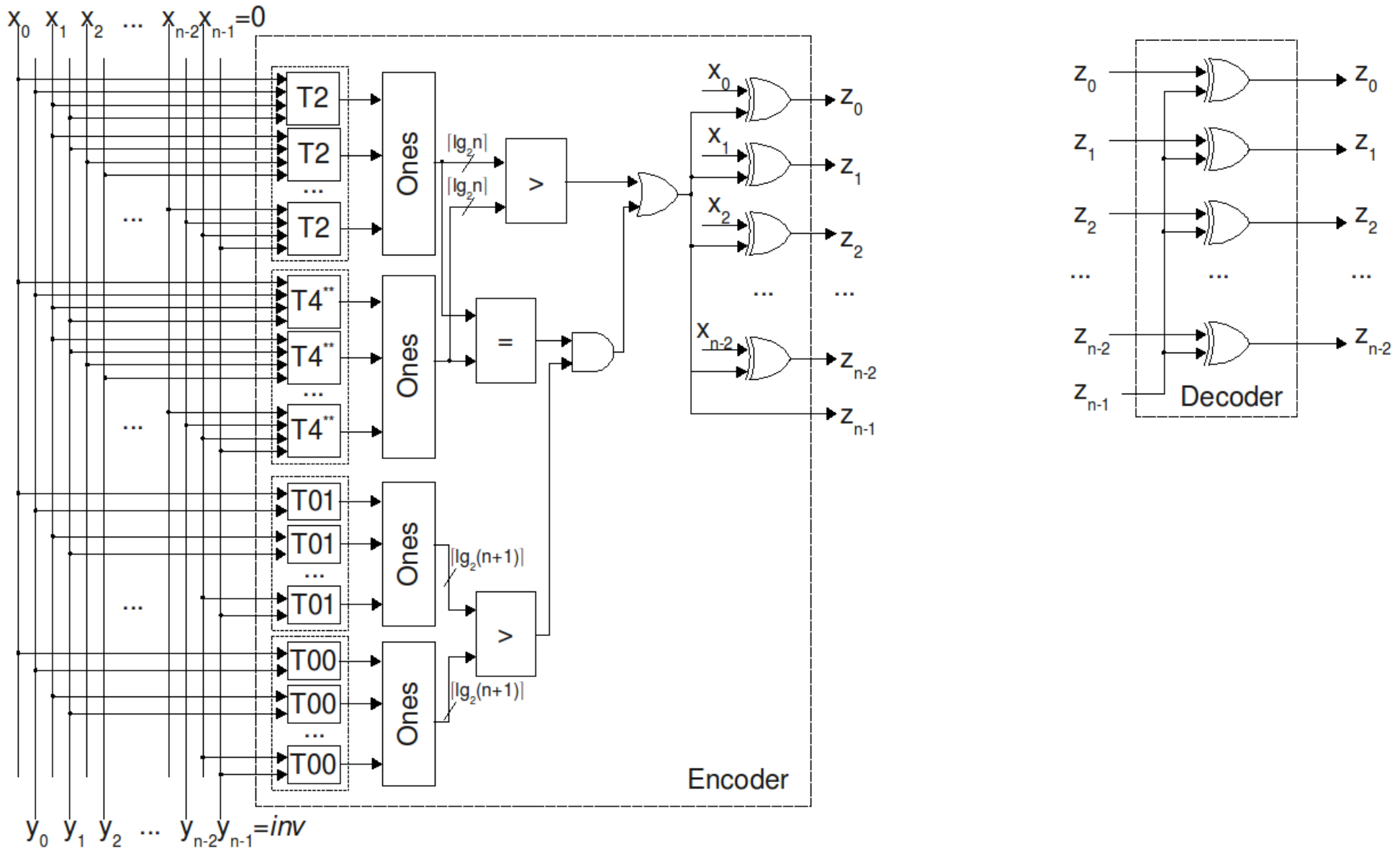
# Proposed Scheme (2/2)

Time	Normal				Inverted			
	Type I				Type I			
$t-1$	00	00	11	11	00	00	11	11
$t$	01	10	01	10	10	01	10	01
	Type II				Type IV			
$t-1$	01	10			01	10		
$t$	10	01			01	10		
	Type III				Type IV			
$t-1$	00	11			00	11		
$t$	11	00			00	11		
	Type IV				Type II and III			
$t-1$	00	11	01	10	00	11	01	10
$t$	00	11	01	10	11	00	10	01
	$T_4^*$		$T_4^{**}$		$T_3$		$T_2$	

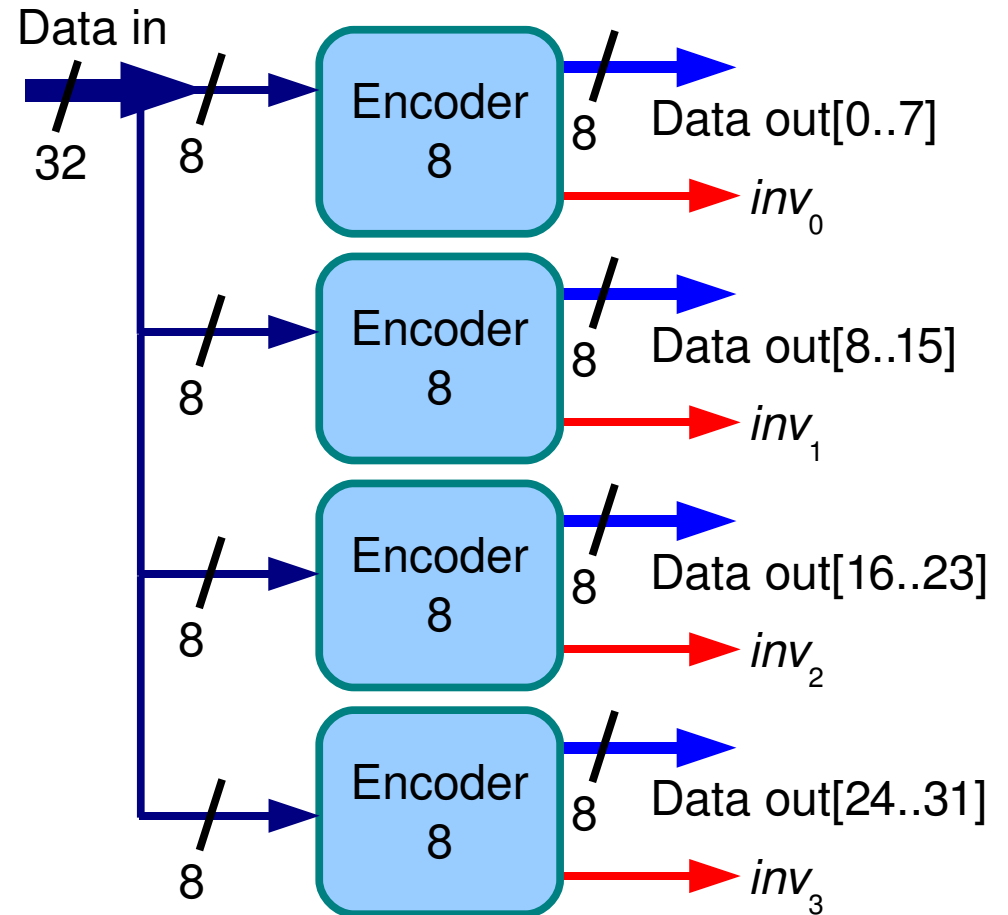
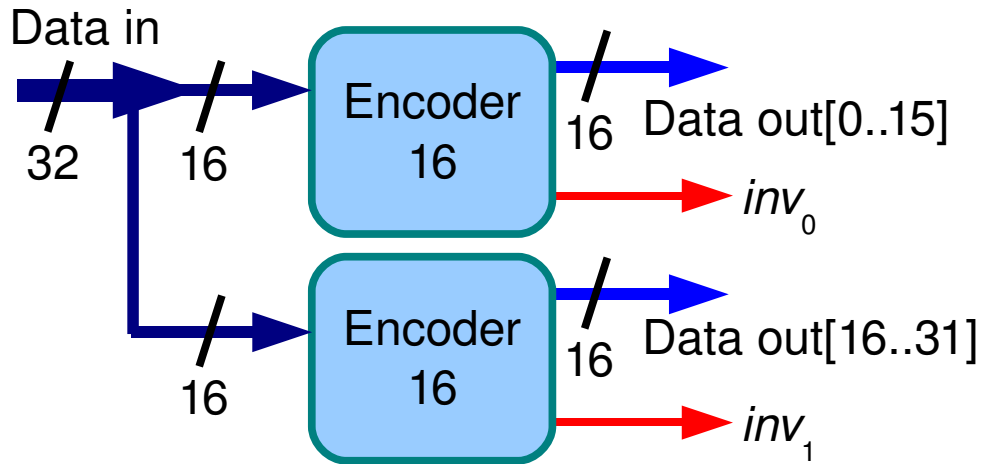
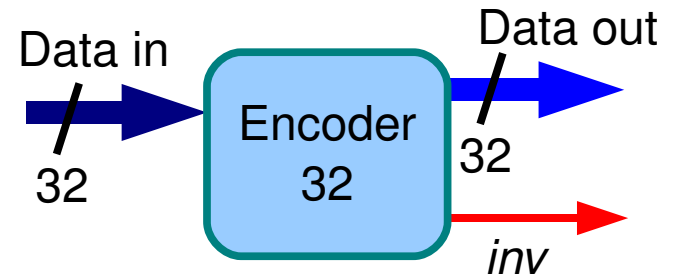
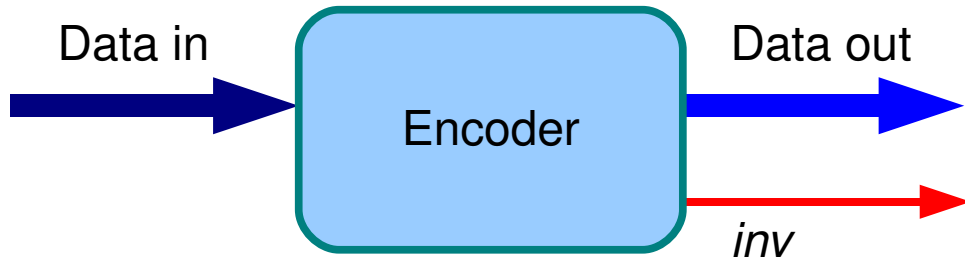
■ Invert if  $P > P'$

$$T_{0 \rightarrow 1} + 8T_2 > T_{0 \rightarrow 0} + 8T_4^{**}$$

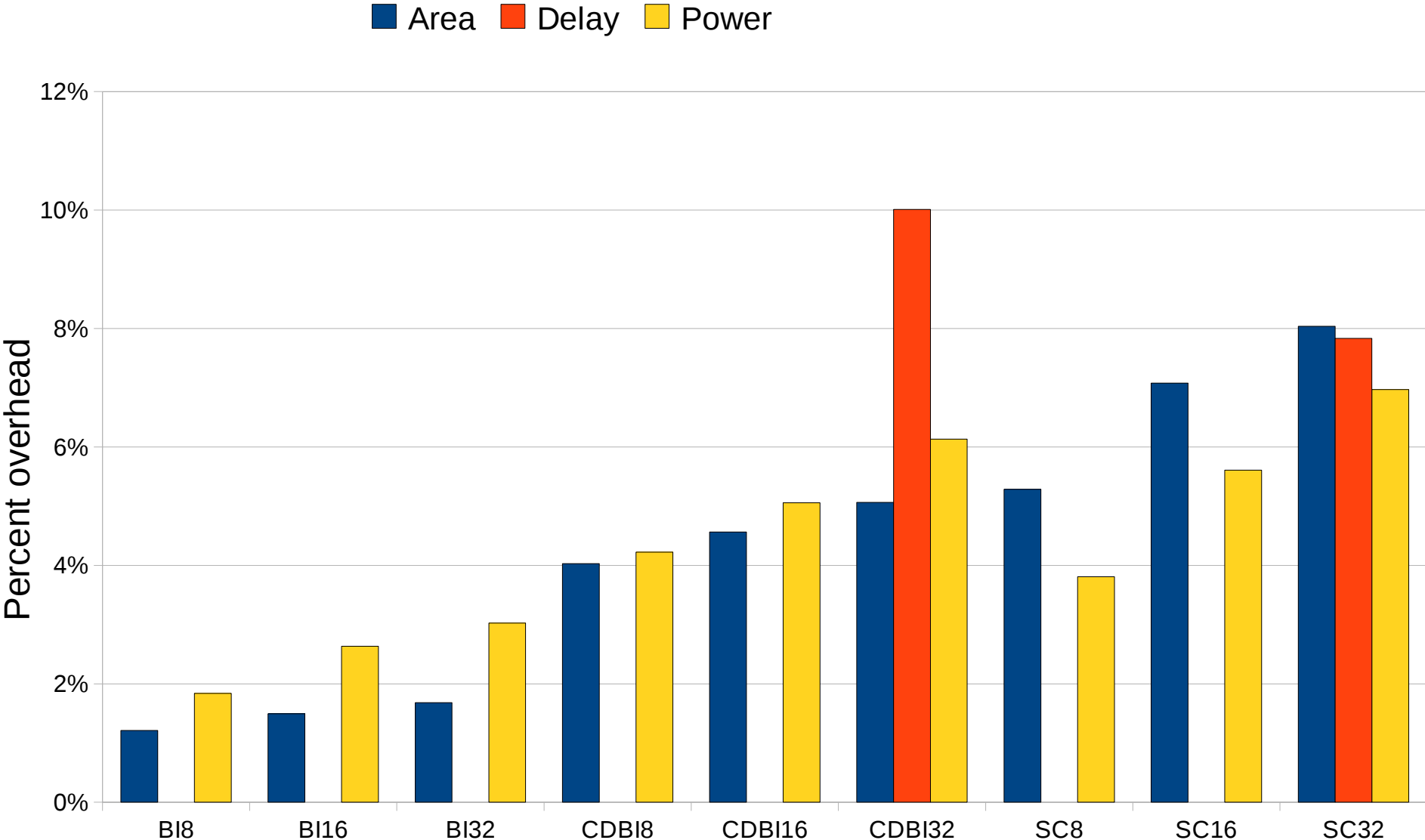
# Encoder and Decoder



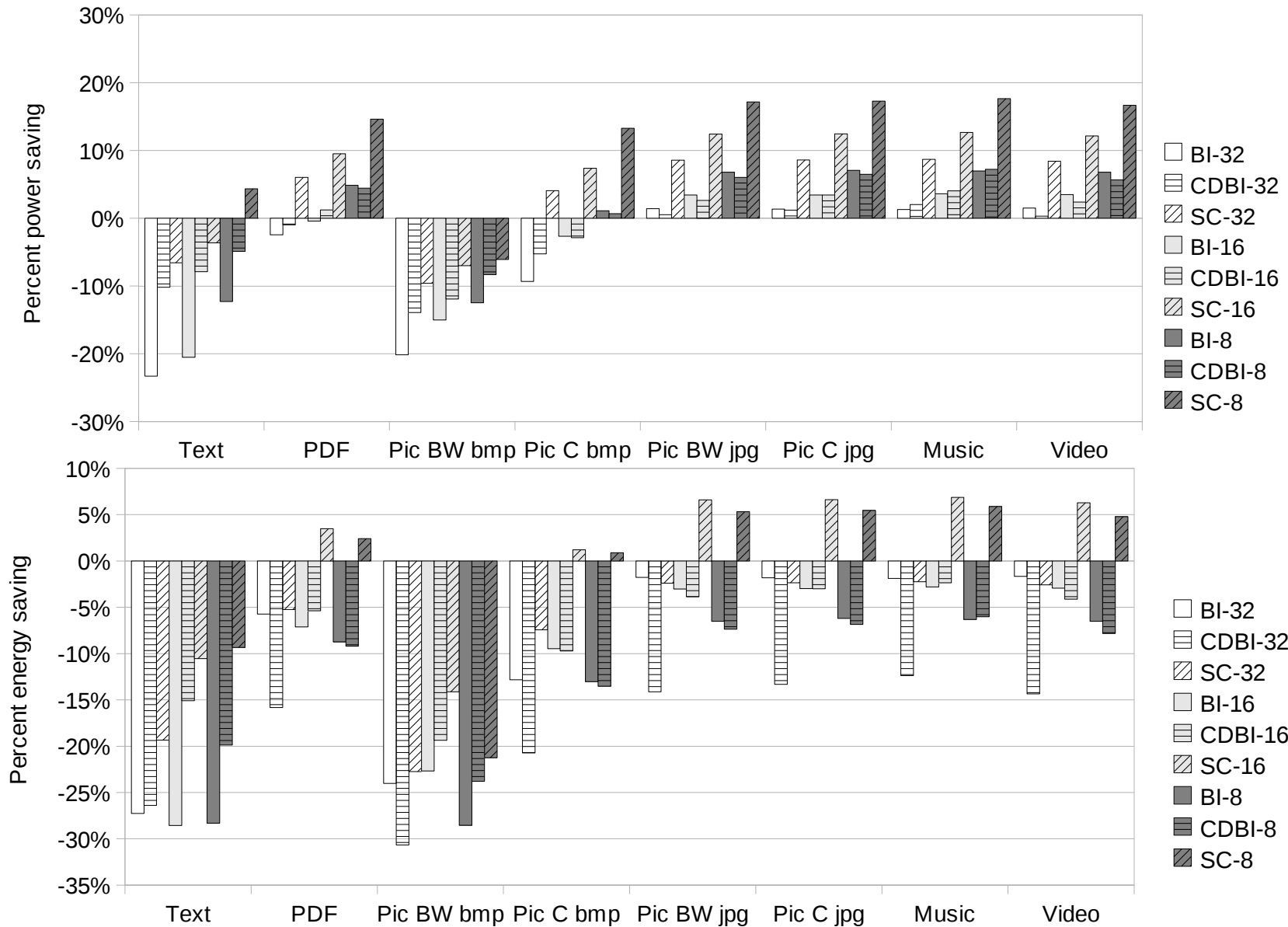
# Partitions



# Overhead

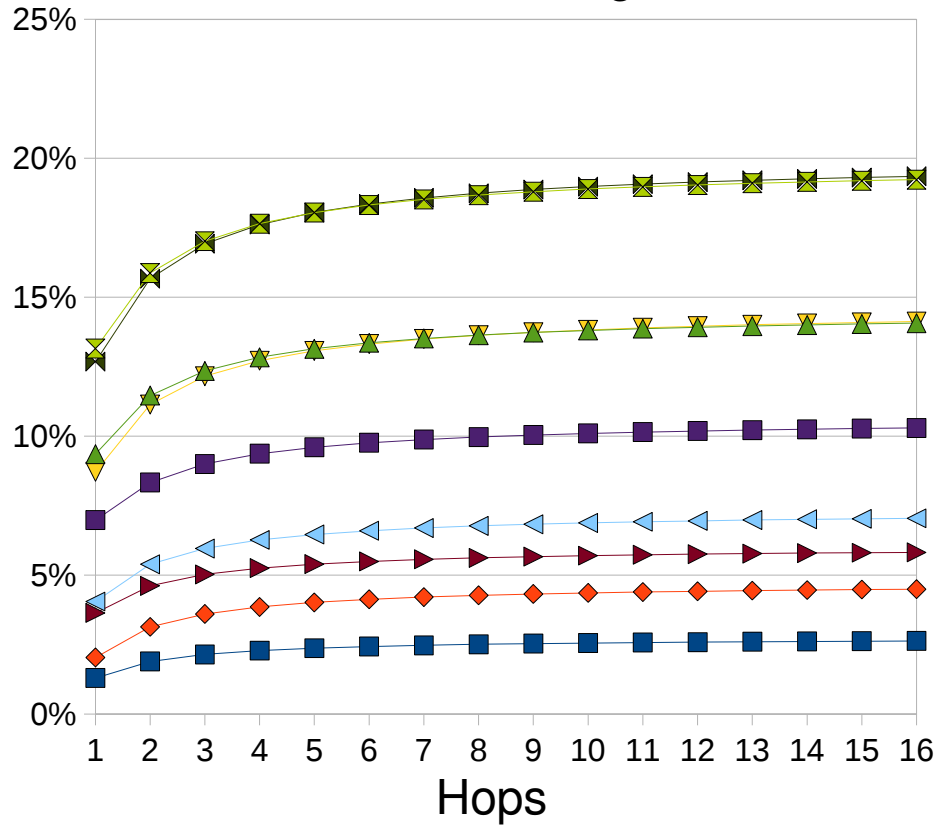


# Power and Energy Saving

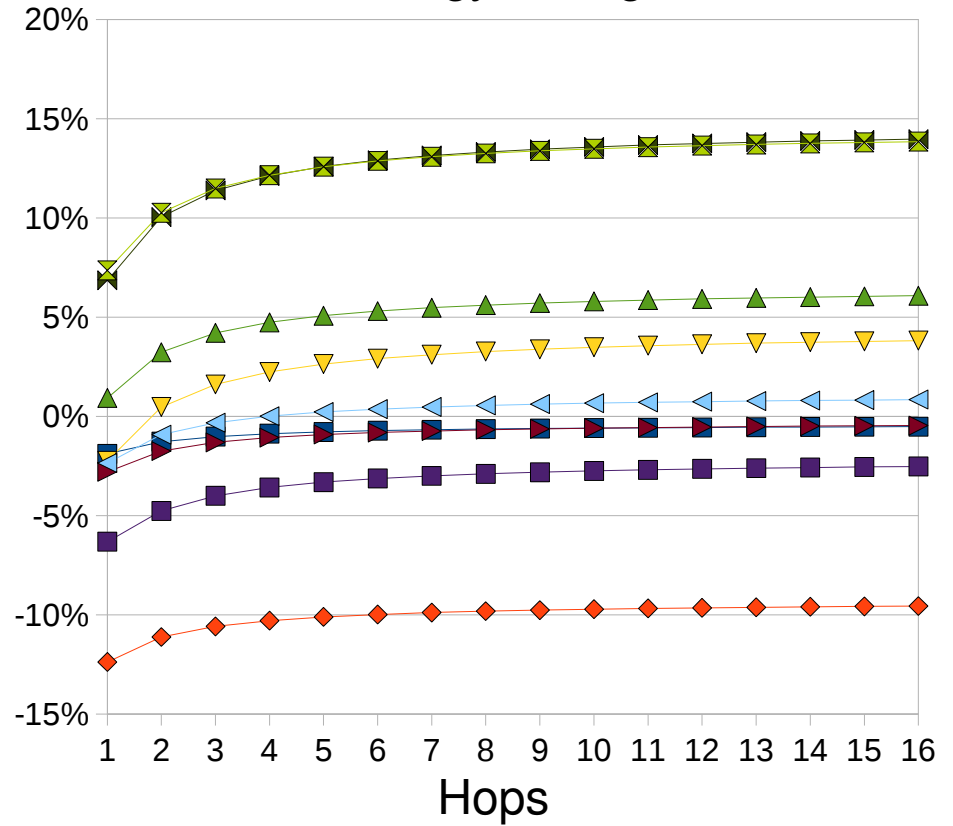


# Saving vs. Path Length

Power saving



Energy saving



■ BI-32  
 ◆ CDBI-32  
 ▼ SC-32  
 ▲ Scs-32  
 ▶ BI-16  
 ◀ CDBI-16  
 ✱ SC-16  
 ✱ Scs-16  
 ■ BI-8

# Outline

---

- Application Specific Routing Algorithms ✓
- Concurrent Mapping and Routing ✓
- Dealing with Manufacturing Defects ✓
- Encoding Scheme for Low Power ✓

# References (1/3)

---

- M. Palesi, S. Kumar, V. Catania. Leveraging Partially Faulty Links Usage for Enhancing Yield and Performance in Networks on Chip. Accepted for publication in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.
- M. Palesi, S. Kumar, V. Catania. Bandwidth Aware Routing Algorithms for Networks-on-Chip Platforms. Computers & Digital Techniques, IET, Vol. 3, No. 5. (11 August 2009), pp. 413-429.
- M. Palesi, R. Holsmark, S. Kumar, V. Catania. Application Specific Routing Algorithms for Networks on Chip. IEEE Transactions on Parallel and Distributed Systems, 20(3), pp. 316-330, March 2009.
- A. Mejia, M. Palesi, J. Flich, S. Kumar, P. Lopez, R. Holsmark and J. Duato. Region-Based Routing: A Mechanism to Support Efficient Routing Algorithms in NoCs IEEE Transactions on on Very Large Scale Integration Systems, 17(3), pp. 356-369, March 2009.
- G. Ascia, V. Catania, M. Palesi, D. Patti. Implementation and Analysis of a New Selection Strategy for Adaptive Routing in Networks-on-Chip. IEEE Transactions on Computers, 57(6), pp. 809-820, June 2008.
- R. Holsmark, M. Palesi, S. Kumar. Deadlock free Routing Algorithms for Irregular Mesh Topology NoC Systems with Rectangular Regions. Journal of Systems Architecture, 54/3-4 (2008) pp. 427-440.
- D. Bertozzi, S. Kumar, M. Palesi. Networks-on-Chip: Emerging Research Topics and Novel Ideas. VLSI Design, vol. 2007, Article ID 26454, doi:10.1155/2007/26454.
- G. Ascia, V. Catania, M. Palesi. A Multi-objective Genetic Approach to Mapping Problem on Network-on-Chip. Journal of Universal Computer Science, 12(4):370--394, 2006.
- G. Ascia, V. Catania, M. Palesi. Mapping Cores on Network-on-Chip. International Journal of Computational Intelligence Research (IJCIR), ISSN 0972-9836, 1(1-2):109--126, 2005.
- M. Palesi, F. Fazzino, G. Ascia, V. Catania. Data Encoding for Low-Power in Wormhole-Switched Networks-on-Chip. To appear in The 12th Euromicro Conference on Digital System Design (DSD), 27-29 Aug 2009, Patras, Greece.

# References (2/3)

- R. Tornero, V. Sterrantino, M. Palesi, J. M. Orduna. A Multi-objective Strategy for Concurrent Mapping and Routing in Networks on Chip. To appear in The 12th International Workshop on Nature Inspired Distributed Computing held in conjunction with The 23th IEEE/ACM International Parallel and Distributed Processing, May 25-28, 2009, Rome, Italy.
- R. Holsmark, M. Palesi, S. Kumar, A. Mejia. HiRA: A Methodology for Deadlock Free Routing in Hierarchical Networks on Chip. 3rd ACM/IEEE International Symposium on Networks on Chip. May 10-13, 2009, San Diego, CA
- D. Frazzetta, G. Dimartino, M. Palesi, S. Kumar, V. Catania. Efficient Application Specific Routing Algorithms for NoC Systems utilizing Partially Faulty Links. 11th EUROMICRO Conference on Digital System Design, Architectures, Methods and Tools, pp. 18-25, Sep. 3-5, 2008, Parma, Italy.
- R. Tornero, J. M. Orduna, M. Palesi, J. Duato. A Communication-Aware Topological Mapping Technique for NoCs. International Conference on Parallel and Distributed Computing, pp. 910-919, August 26-29th, 2008, Las Palmas de Gran Canaria, Spain.
- M. Palesi, G. Longo, S. Signorino, S. Kumar, R. Holsmark, V. Catania. Design of Bandwidth Aware and Congestion Avoiding Efficient Routing Algorithms for Networks-on-Chip Platforms. IEEE International Symposium on Networks-on-Chip, pp. 97-106, 7th-11th April 2008, Newcastle University, UK.
- G. Longo, S. Signorino, M. Palesi, S. Kumar, R. Holsmark, V. Catania. Bandwidth Aware Routing Algorithms for Networks-on-Chip. 2nd Workshop on Interconnection Network Architectures: On-Chip, Multi-Chip. Goteborg, Sweden, January 27, 2008.
- R. Tornero, J. M. Orduna, M. Palesi, J. Duato. A Communication-Aware Task Mapping Technique for NoCs. 2nd Workshop on Interconnection Network Architectures: On-Chip, Multi-Chip. Goteborg, Sweden, January 27, 2008.
- M. Palesi, S. Kumar, R. Holsmark, V. Catania. Exploiting Communication Concurrency for Efficient Deadlock Free Routing in Reconfigurable NoC Platforms. IEEE International Parallel and Distributed Processing Symposium, pp. 1-8, Long Beach, CA, March 2007.
- G. Ascia, V. Catania, M. Palesi, D. Patti. Neighbors-on-Path: A New Selection Strategy for On-Chip Networks. Fourth IEEE Workshop on Embedded Systems for Real Time Multimedia, pp. 79-84. Seoul, Korea, October 26-27, 2006.

# References (3/3)

---

- M. Palesi, R. Holsmark, S. Kumar, V. Catania. A Methodology for Design of Application Specific Deadlock-free Routing Algorithms for NoC Systems. International Conference on Hardware-Software Codesign and System Synthesis, pp. 142-147. Seoul, Korea, October 22-25, 2006.
- R. Holsmark, M. Palesi, S. Kumar. Deadlock Free Routing Algorithms for Mesh Topology NoC Systems with Regions. DSD 2006, 9th EUROMICRO Conference on Digital System Design, Architectures, Methods and Tools, pp. 696-703. Croatia, Sept 2006.
- M. Palesi, S. Kumar, R. Holsmark. A Method for Router Table Compression for Application Specific Routing in Mesh Topology NoC Architectures. SAMOS VI Workshop: Embedded Computer Systems: Architectures, Modeling, and Simulation, pp. 373-384. Samos, Greece, July 17-20, 2006.
- G. Ascia, V. Catania, M. Palesi, D. Patti. A New Selection Policy for Adaptive Routing in Network on Chip. International Conference on Electronics, Hardware, Wireless and Optical Communications. Madrid, Spain, February 15-17, 2006.
- G. Ascia, V. Catania, M. Palesi. An Evolutionary Approach to Network-on-Chip Mapping Problem. IEEE Congress on Evolutionary Computation. Edinburgh, UK, September 2nd-5th, 2005.
- G. Ascia, V. Catania, M. Palesi. Multi-objective Mapping for Mesh-based NoC Architectures. In Second IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, pages 182-187, Stockholm, Sweden, Sept. 8-10, 2004.