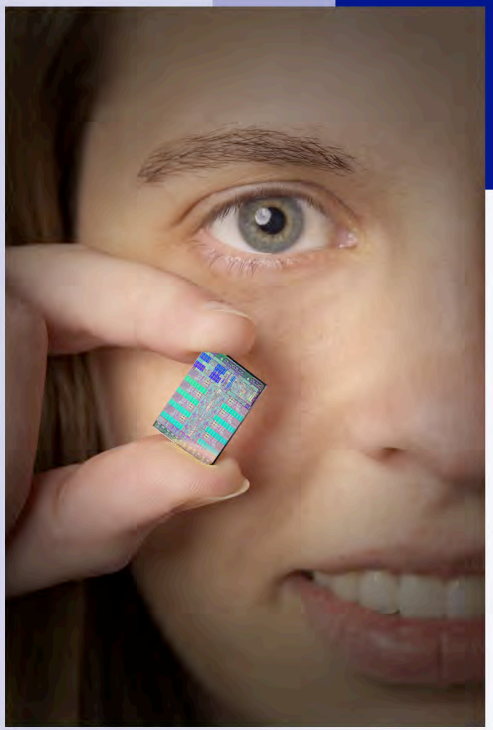
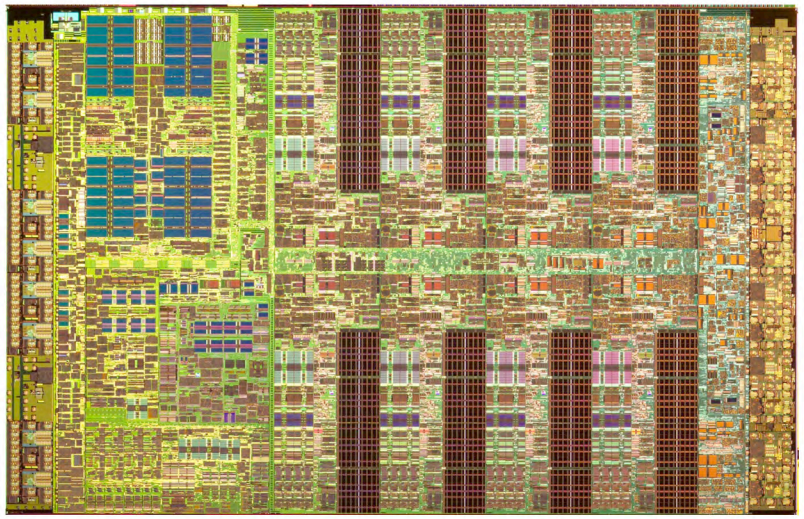
# *The revolution of the new millennium…*



Images Courtesy of IBM

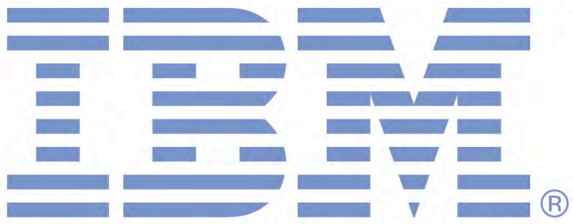


# The Road to Multicore

# The Road to Multicore

2001: IBM Releases its first commercial Dual-Core CPU, the **IBM POWER4**

2002: **POWER4+** is released



# The Road to Multicore

2001: IBM Releases its first commercial Dual-Core CPU, the **IBM POWER4**


2002: **POWER4+** is released

2003: HP Releases the **HP PA-RISC 8800** Dual Core CPU

# The Road to Multicore

2001: IBM Releases its first commercial Dual-Core CPU, the **IBM POWER4**


2002: **POWER4+** is released

2003: HP Releases the **HP PA-RISC 8800** Dual Core CPU

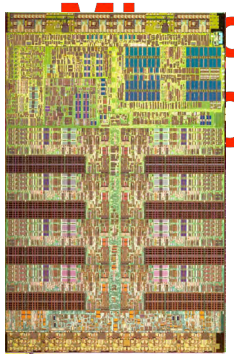

2004: Sun Microsystems Releases the **Sun UltraSPARC IV** Dual Core CPU

# More Multicores

# More Multicores

2004: IBM Releases the Dual-Core **IBM POWER5**

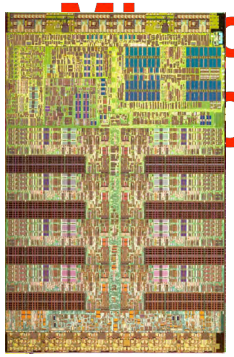
2005: **PowerPC 970MP** (used in the Apple PowerMac G5), **Cell Microprocessor** (8 Cores), **Xbox CPU** (3 Cores)

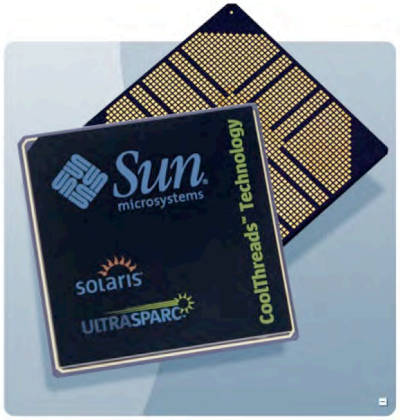


# More Multicores

2004: IBM Releases the Dual-Core **IBM POWER5**

2005: **PowerPC 970MP** (used in the Apple PowerMac G5), **Cell Microprocessor** (8 Cores), **Xbox CPU** (3 Cores)




2005: Sun Microsystems Releases the **Sun UltraSPARC T1** (codename **Niagara**) 8-Core CPU

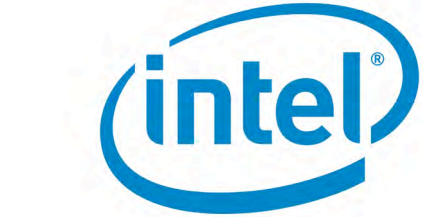
# *Multicore Becomes Mainstream*

# Multicore Becomes Mainstream

2005: **Intel Pentium D** (Smithfield, Presler)

2006: **Intel Core Duo** (Yonah), **Core 2 Duo** (Conroe, Merom, Woodcrest)

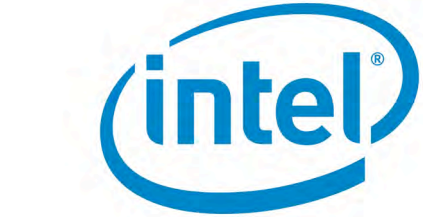
2006: **Intel** fabs **80-core teraflop** processor





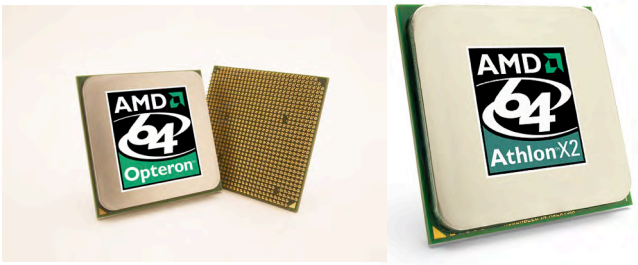
# Multicore Becomes Mainstream

2005: **Intel Pentium D** (Smithfield, Presler)
2006: **Intel Core Duo** (Yonah), **Core 2 Duo** (Conroe, Merom, Woodcrest)
2006: **Intel** fabs **80-core teraflop** processor






2005: **AMD Opteron** (Egypt, Italy, Denmark), **Athlon 64 X2** (Manchester, Toledo)
2006: AMD Athlon 64 X2 (Windsor), Athlon **Turion 64 X2** (Taylor)

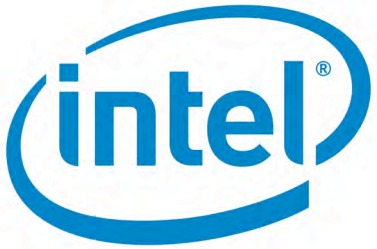



# Even More Cores in the Pipeline

# *Even More Cores in the Pipeline*

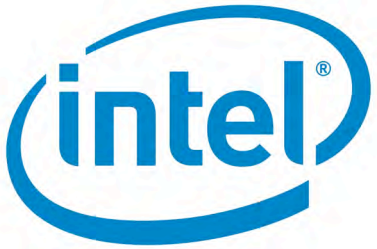
2007 : Intel Kentsfield, Clovertown **Quad-Core CPUs**

2009: Intel Yorkfield **8-Core CPU** (45 nm)



# *Even More Cores in the Pipeline*

2007 : Intel Kentsfield, Clovertown **Quad-Core CPUs**

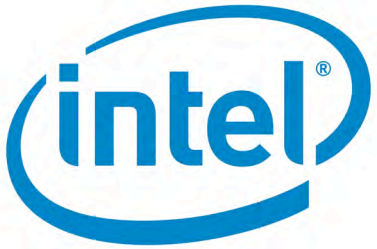



2009: Intel Yorkfield **8-Core CPU** (45 nm)

2007/8: AMD Deerhound, Greyhound, Zamora **Quad-Core CPUs**

# *Even More Cores in the Pipeline*

2007 : Intel Kentsfield, Clovertown **Quad-Core CPUs**



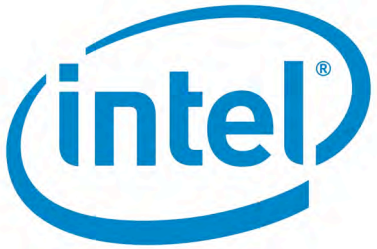

2009: Intel Yorkfield **8-Core CPU** (45 nm)

2007/8: AMD Deerhound, Greyhound, Zamora **Quad-Core CPUs**


2007: Sun Rock, Niagara 2 **8-Core CPUs**

# Even More Cores in the Pipeline

2007 : Intel Kentsfield, Clovertown **Quad-Core CPUs**




2009: Intel Yorkfield **8-Core CPU** (45 nm)

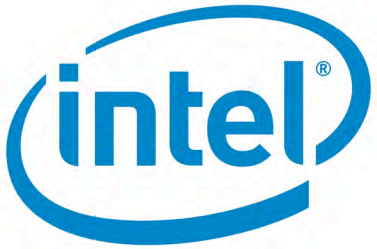
**Multi-Core Chips are the Way of the Future!!!**

2007/8: AMD Deerhound, Greyhound, Zamora **Quad-Core CPUs**


2007: Sun Rock, Niagara 2 **8-Core CPUs**

# *Even More Cores in the Pipeline*

2007 : Intel Kentsfield, Clovertown **Quad-Core CPUs**




2009: Intel Yorkfield **8-Core**

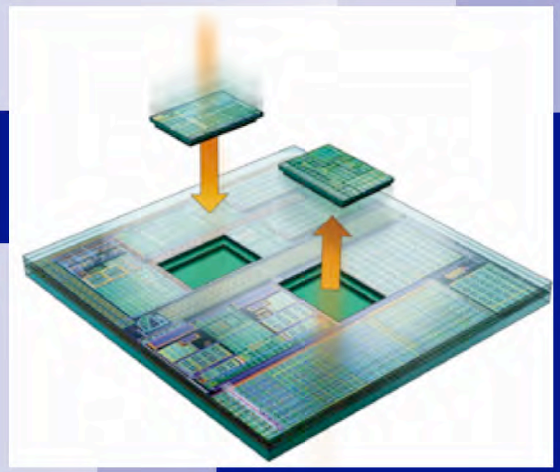
**BUT…could the INTERCONNECT stand in their way???**

Greyhound, Zamora **Quad-Core CPUs**


2007: Sun Rock, Niagara 2 **8-Core CPUs**

## 2006 Workshop on On- and Off-Chip Interconnection Networks for Multicore Systems

Dec 6, 2006



# Exploring NoC Architecture Design Space for Multicore Systems


Chita R. Das

PENNSTATE

The Pennsylvania State University

Department of Computer Science & Engineering

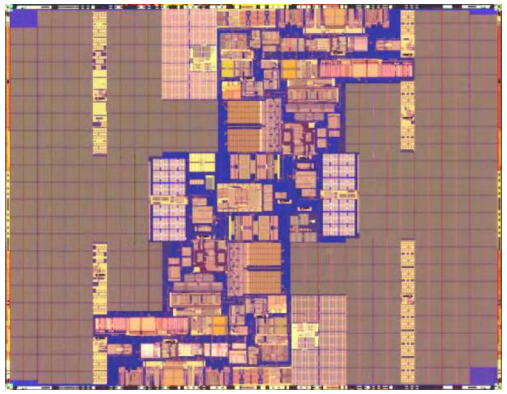
# Talk Outline


- Motivation
- A Typical NoC Router Architecture
- **The Row-Column (RoCo) Router**
- **Unified Buffer Organization: ViChaR**
- **3D NoC Architectures**
- **CNT-based  Interconnects**
- Conclusions

# The Billion Transistor Era

# The Billion Transistor Era

Photo by Intel

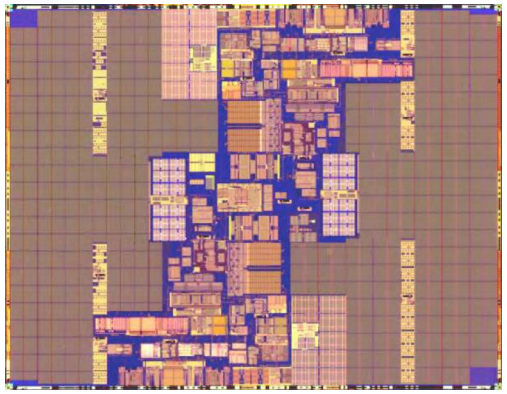


Intel Itanium 2
(Codename Montecito)
**1.7 BILLION**
**transistors per die!**

# The Billion Transistor Era

- Feature sizes **diminishing RAPIDLY** into the nanometer regime

Photo by Intel

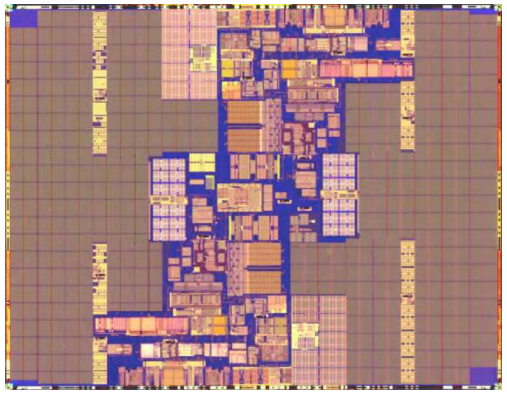


Intel Itanium 2
(Codename Montecito)
**1.7 BILLION transistors per die!**

# The Billion Transistor Era

- Feature sizes **diminishing RAPIDLY** into the nanometer regime
- Transistor densities skyrocketing

Photo by Intel

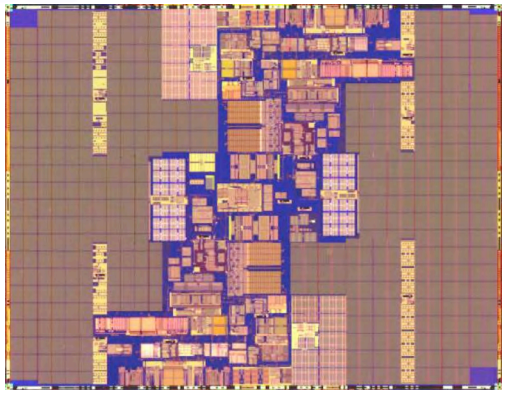


Intel Itanium 2
(Codename Montecito)
**1.7 BILLION transistors per die!**

# The Billion Transistor Era

- Feature sizes **diminishing RAPIDLY** into the nanometer regime
- Transistor densities skyrocketing
- Gate delays are scaling down

Photo by Intel

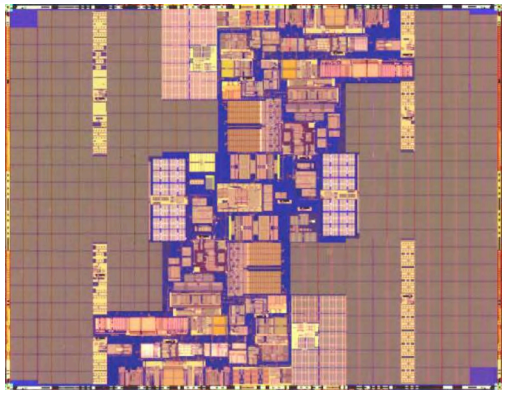


Intel Itanium 2
(Codename Montecito)
**1.7 BILLION transistors per die!**

# The Billion Transistor Era

- Feature sizes **diminishing RAPIDLY** into the nanometer regime
- Transistor densities skyrocketing
- Gate delays are scaling down
- What about Global Wiring delays?

Photo by Intel

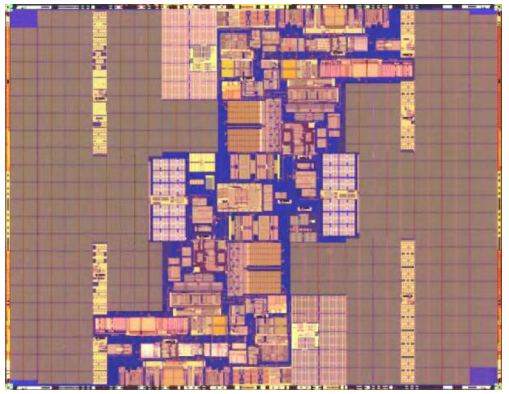


Intel Itanium 2
(Codename Montecito)
**1.7 BILLION transistors per die!**

# The Billion Transistor Era

- Feature sizes **diminishing RAPIDLY** into the nanometer regime
- Transistor densities skyrocketing
- Gate delays are scaling down
- What about Global Wiring delays?
  - As wire cross-sections decrease, resistance INCREASES!

Photo by Intel

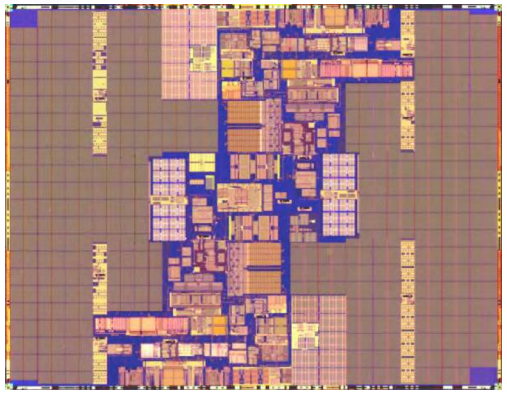


Intel Itanium 2
(Codename Montecito)
**1.7 BILLION transistors per die!**

# The Billion Transistor Era

- Feature sizes **diminishing RAPIDLY** into the nanometer regime
- Transistor densities skyrocketing
- Gate delays are scaling down
- What about Global Wiring delays?
  - As wire cross-sections decrease, resistance INCREASES!
- Interconnects are also an issue in terms of **AREA**, **POWER**, and **RELIABILITY**

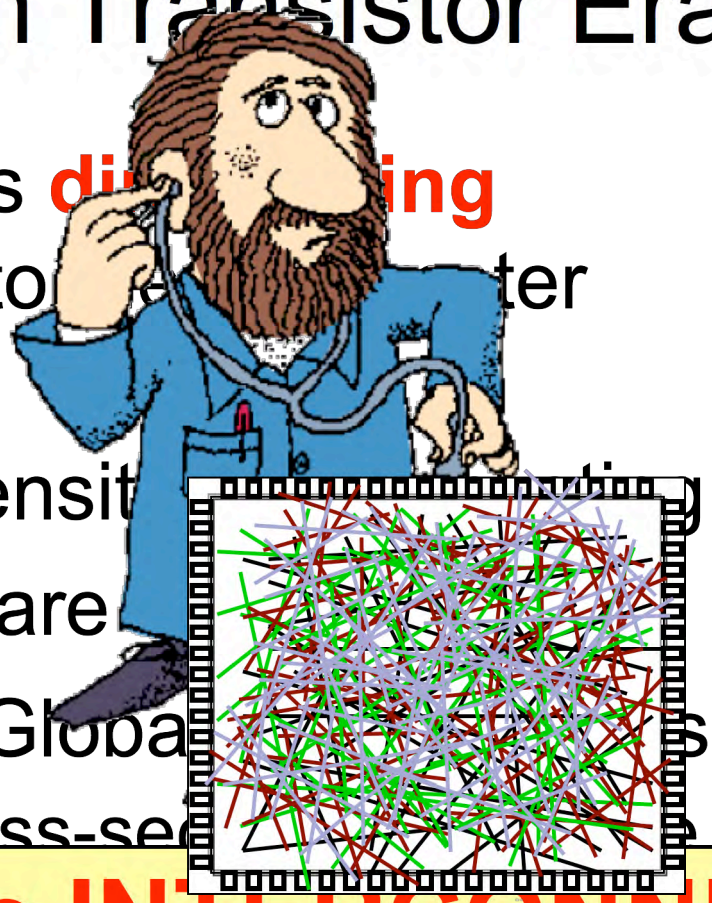
Photo by Intel



Intel Itanium 2
(Codename Montecito)
**1.7 BILLION transistors per die!**

# The Billion Transistor Era

- Feature sizes di ing **RAPIDLY** into ter regime
- Transistor densit g
- Gate delays are
- What about Globa s?
  - As wire cross-se resistance



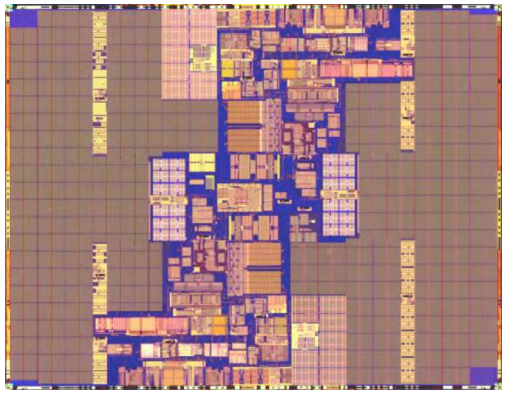


Photo by Intel

Intel Itanium 2
(Codename Montecito)
**1.7 BILLION transistors per die!**

**The INTERCONNECT can no longer be ignored!**

AREA, POWER, and RELIABILITY

# *Old Wine in a New Bottle…*

**Network Research is Old…**

# Old Wine in a New Bottle…

**Network Research is Old…**

**But, Comes With Different Flavors…**

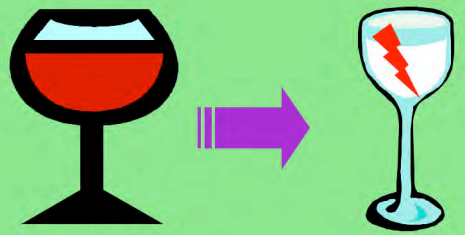
# Old Wine in a New Bottle…

**Network Research is Old…**

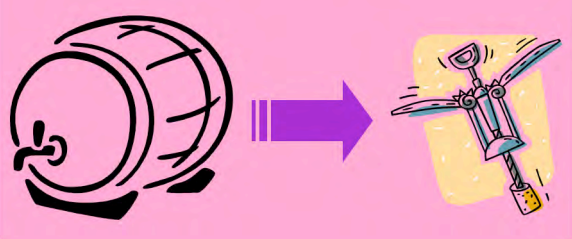
**But, Comes With Different Flavors…**

**Submicron Design:**

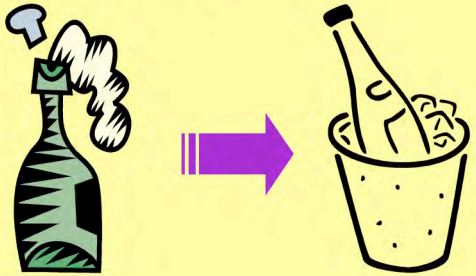
Going Thinner…
Less Reliable…



Power Consumption
Going Higher…



Higher Integration…
Getting Hotter…



# Enter the Network-on-Chip (NoC)!

# Enter the Network-on-Chip (NoC)!

- Replace Global Wires with a Resource-Constrained Network

# Enter the Network-on-Chip (NoC)!

- Replace Global Wires with a Resource-Constrained Network
- Structured Interconnect Layout

# Enter the Network-on-Chip (NoC)!

- Replace Global Wires with a Resource-Constrained Network
- Structured Interconnect Layout
- Electrical Properties OPTIMIZED and WELL CONTROLLED

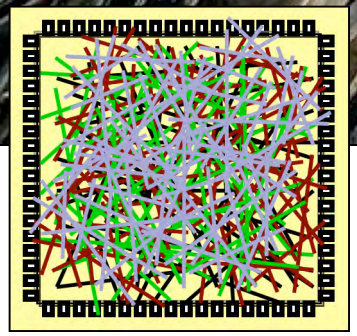
# Enter the Network-on-Chip (NoC)!

- Replace Global Wires with a Resource-Constrained Network
- Structured Interconnect Layout
- Electrical Properties OPTIMIZED and WELL CONTROLLED
- **NoCs are like IP Blocks for Wiring!**

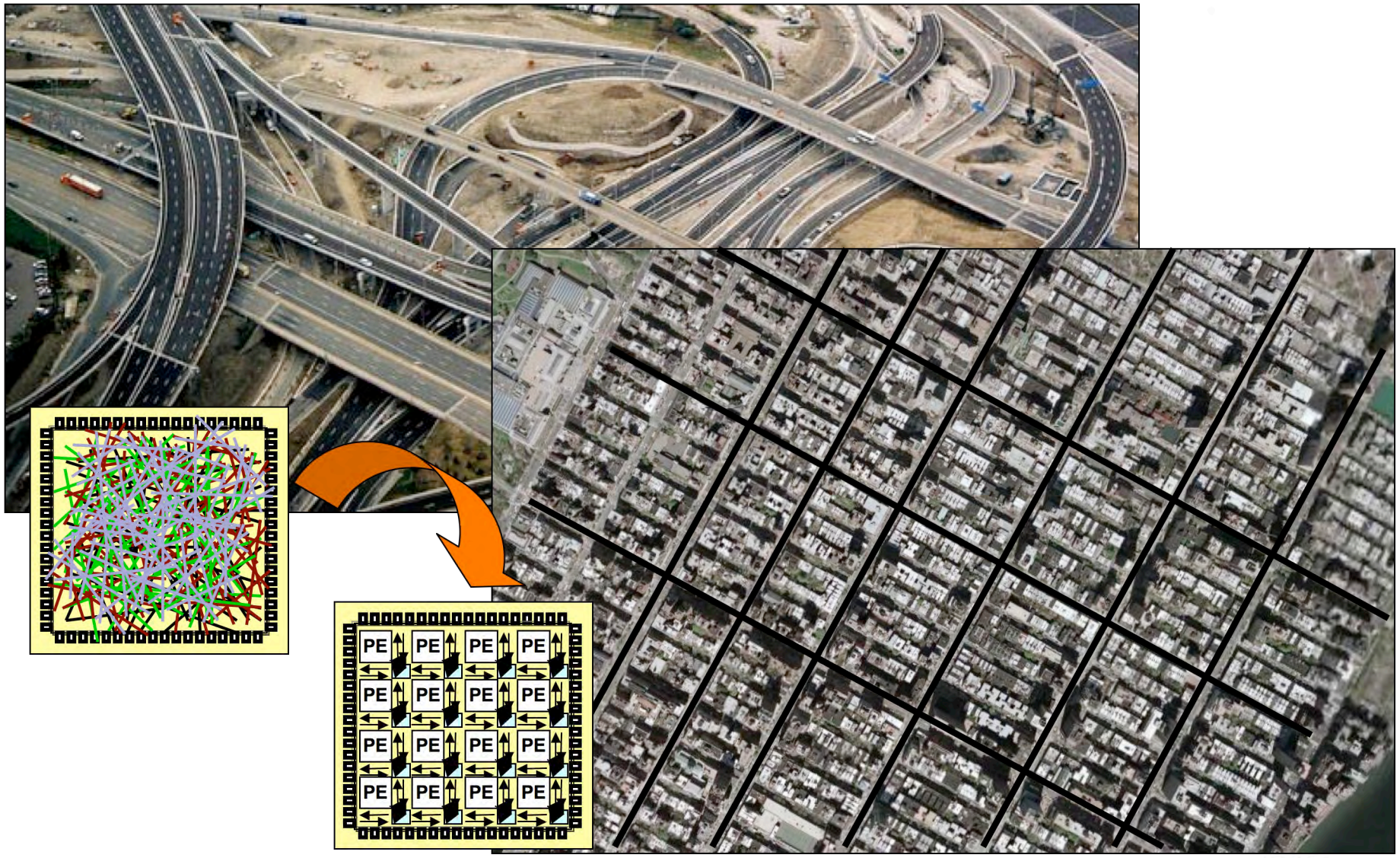
# Enter the Network-on-Chip (NoC)!

LL

!



# Enter the Network-on-Chip (NoC)!



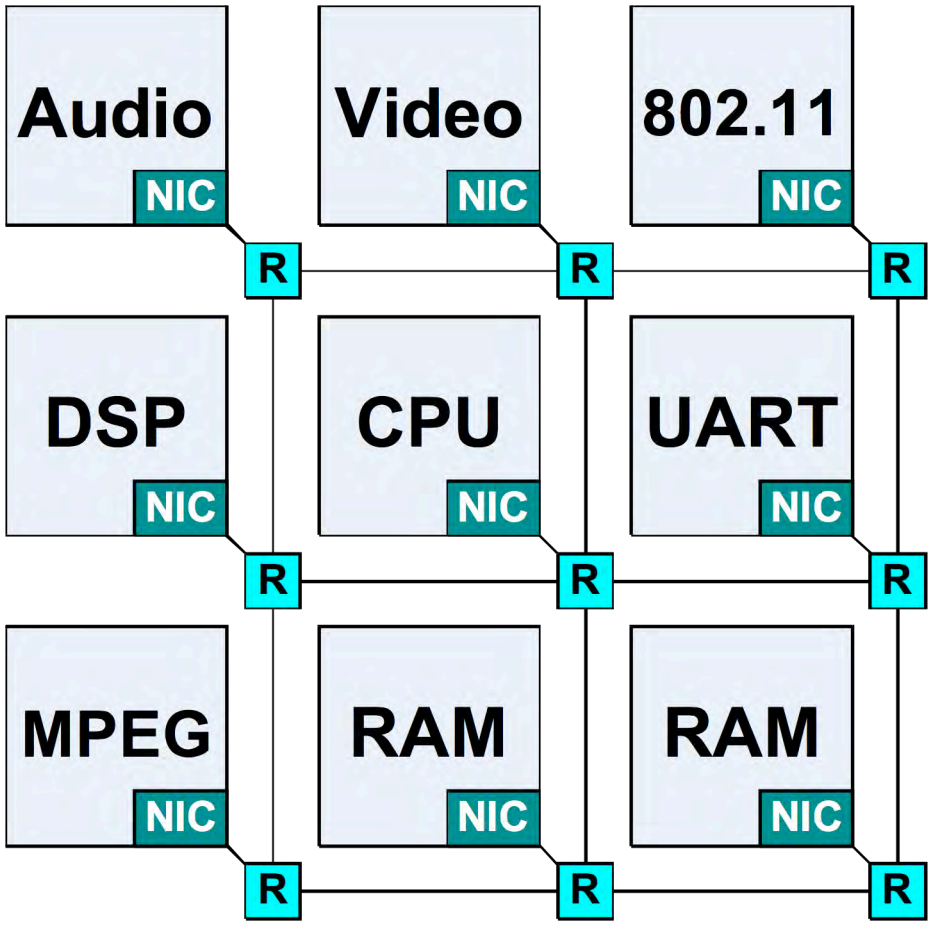
# ***What*** are Networks-on-Chip (NoC)?

# ***What*** are Networks-on-Chip (NoC)?

**Processing Elements** (PEs) interconnected via a ***packet-based*** network

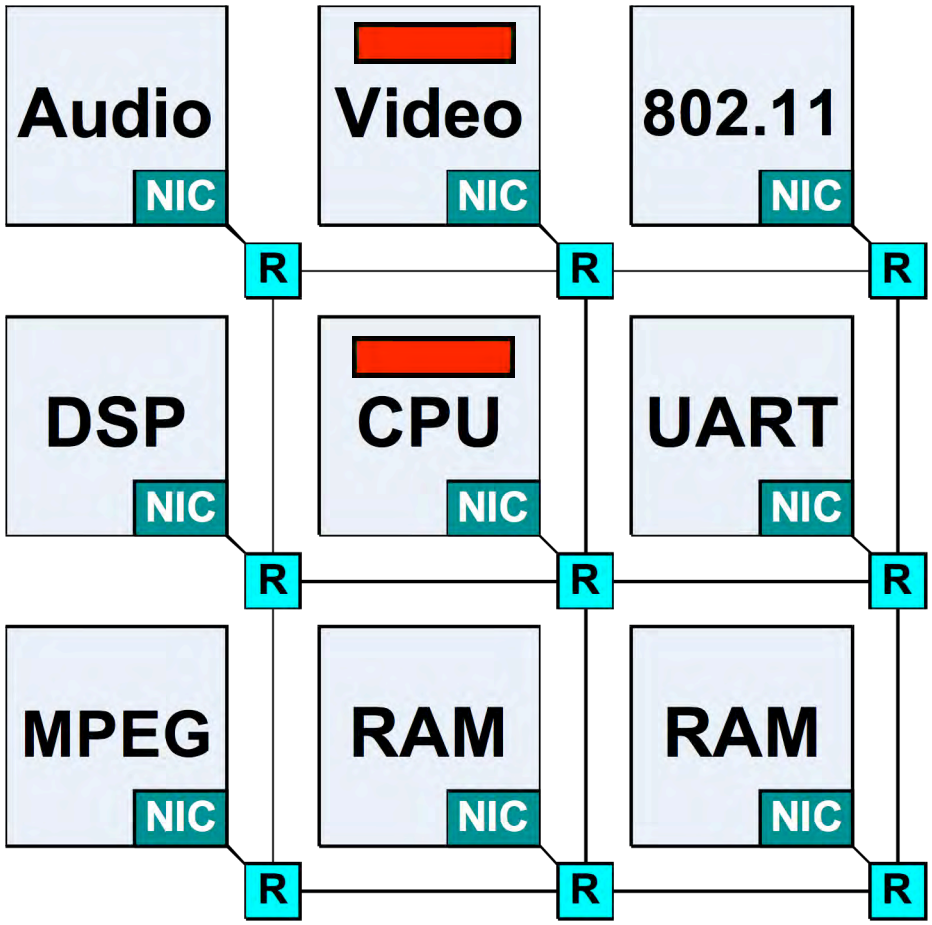
# *What* are Networks-on-Chip (NoC)?

**Processing Elements** (PEs) interconnected via a ***packet-based*** network



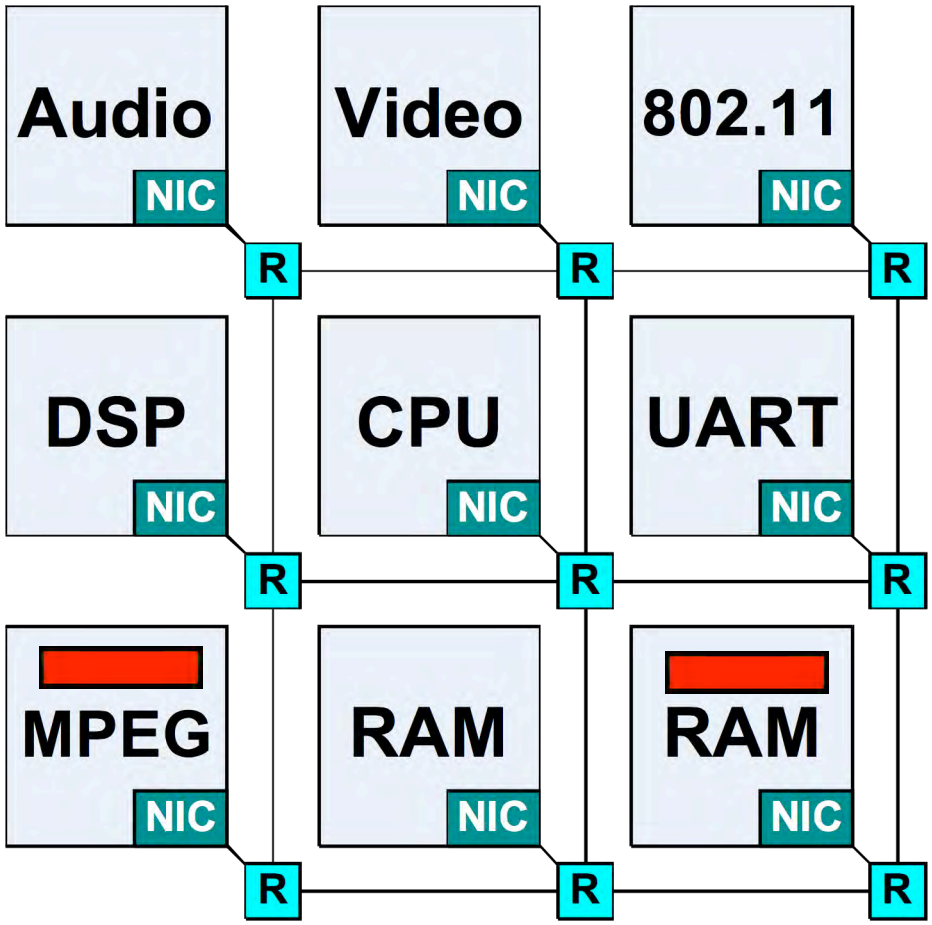
# *What* are Networks-on-Chip (NoC)?

**Processing Elements** (PEs) interconnected via a ***packet-based*** network



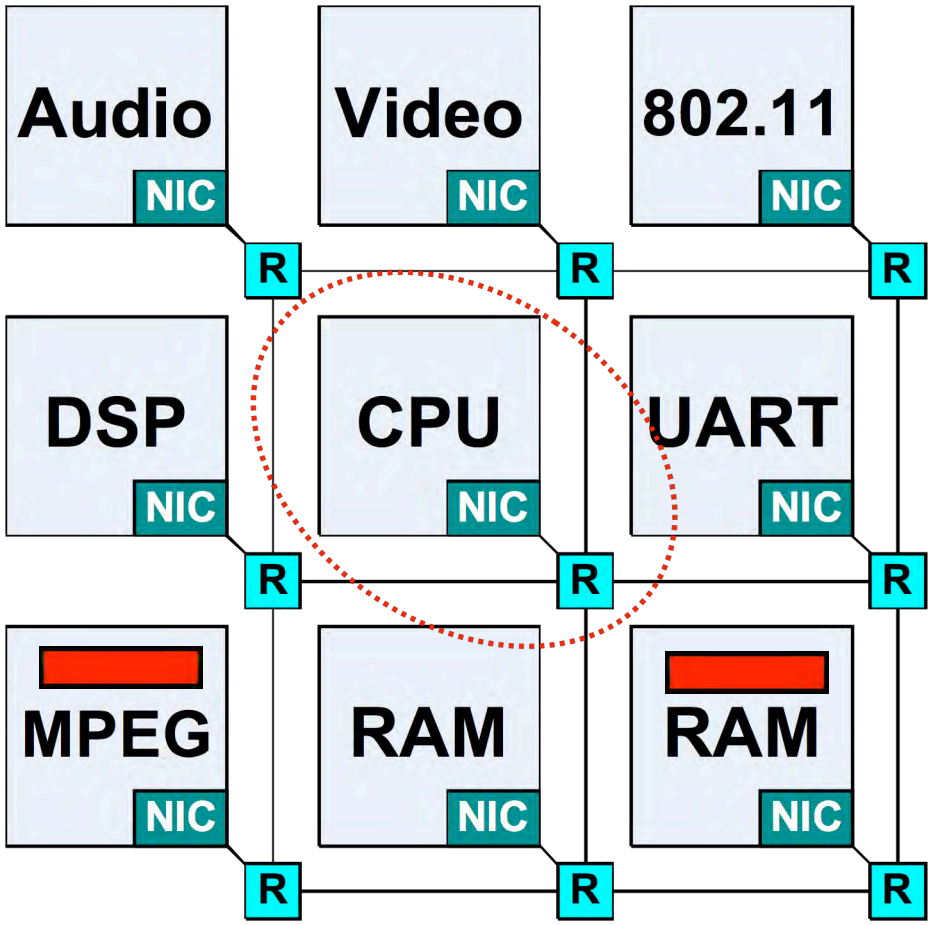
# *What* are Networks-on-Chip (NoC)?

**Processing Elements** (PEs) interconnected via a ***packet-based*** network



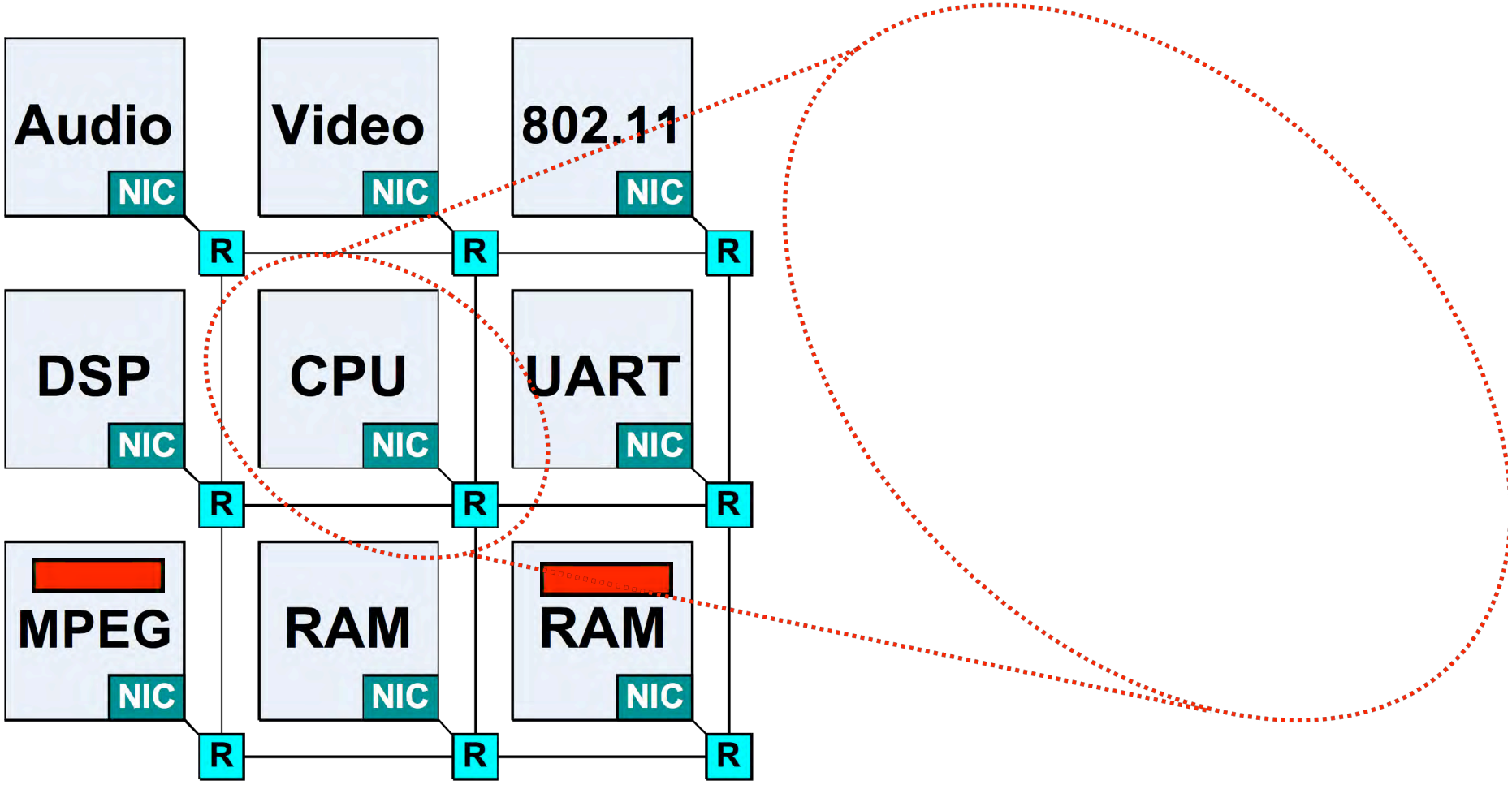
# *What* are Networks-on-Chip (NoC)?

**Processing Elements** (PEs) interconnected via a ***packet-based*** network



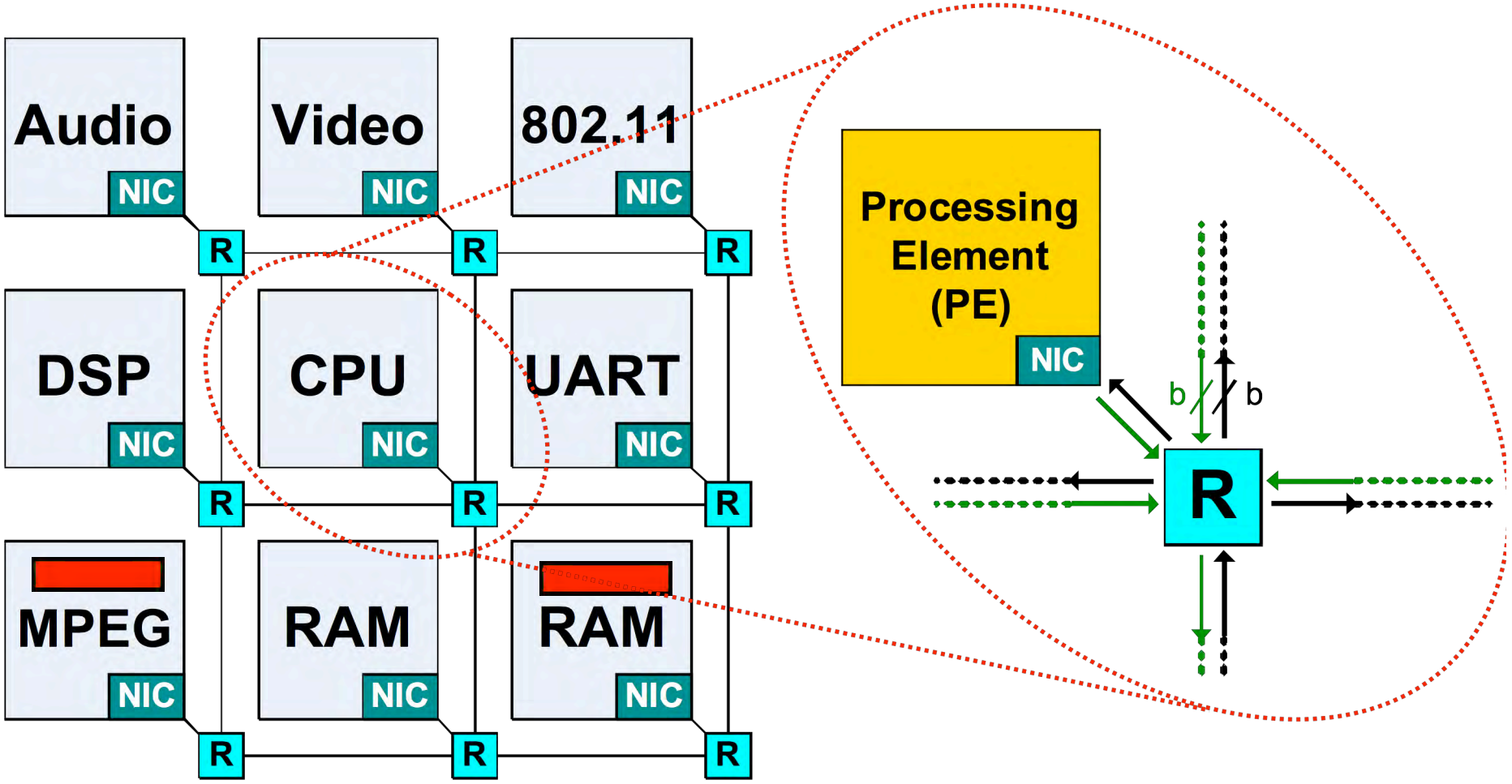
# *What* are Networks-on-Chip (NoC)?

**Processing Elements** (PEs) interconnected via a ***packet-based*** network



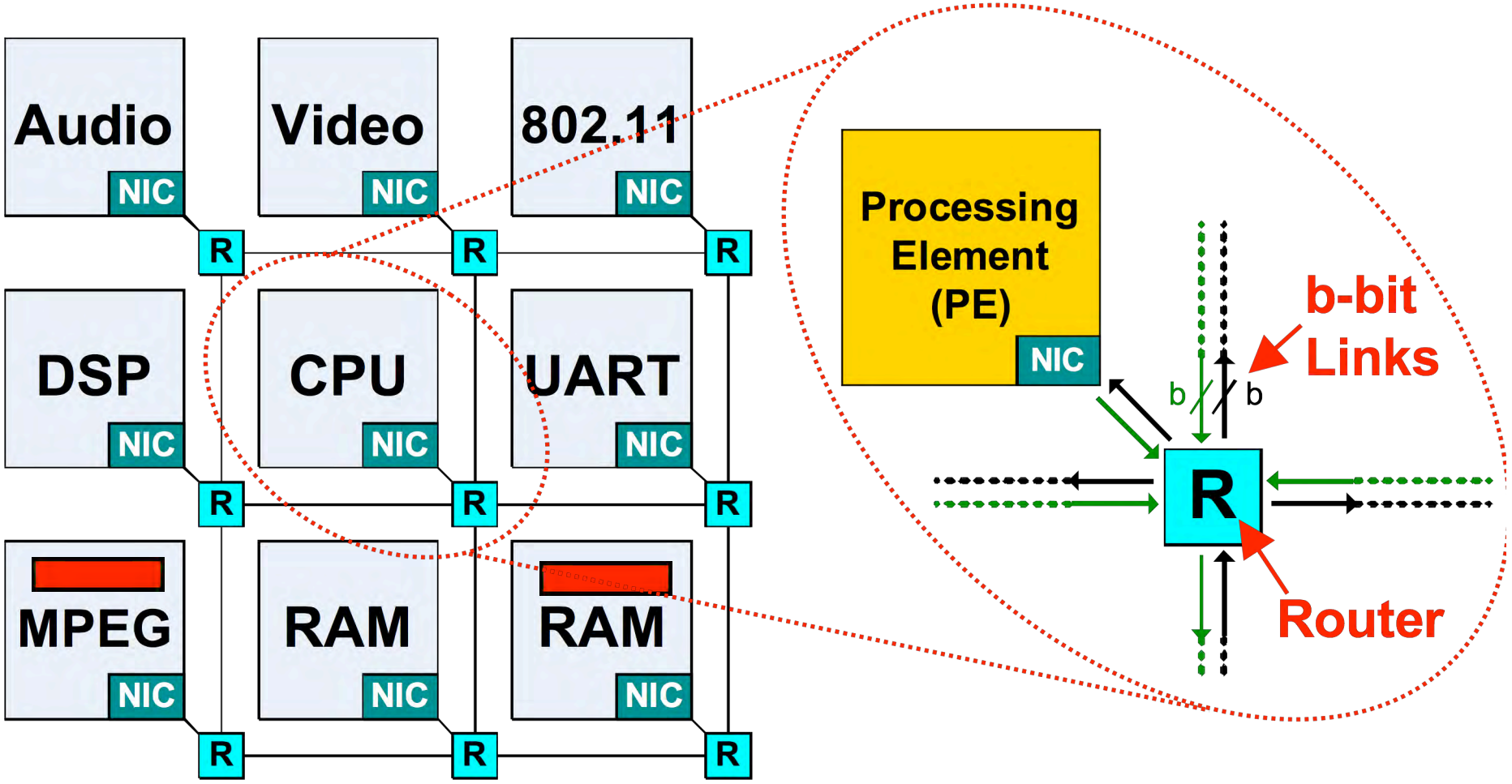
# *What* are Networks-on-Chip (NoC)?

**Processing Elements** (PEs) interconnected via a ***packet-based*** network



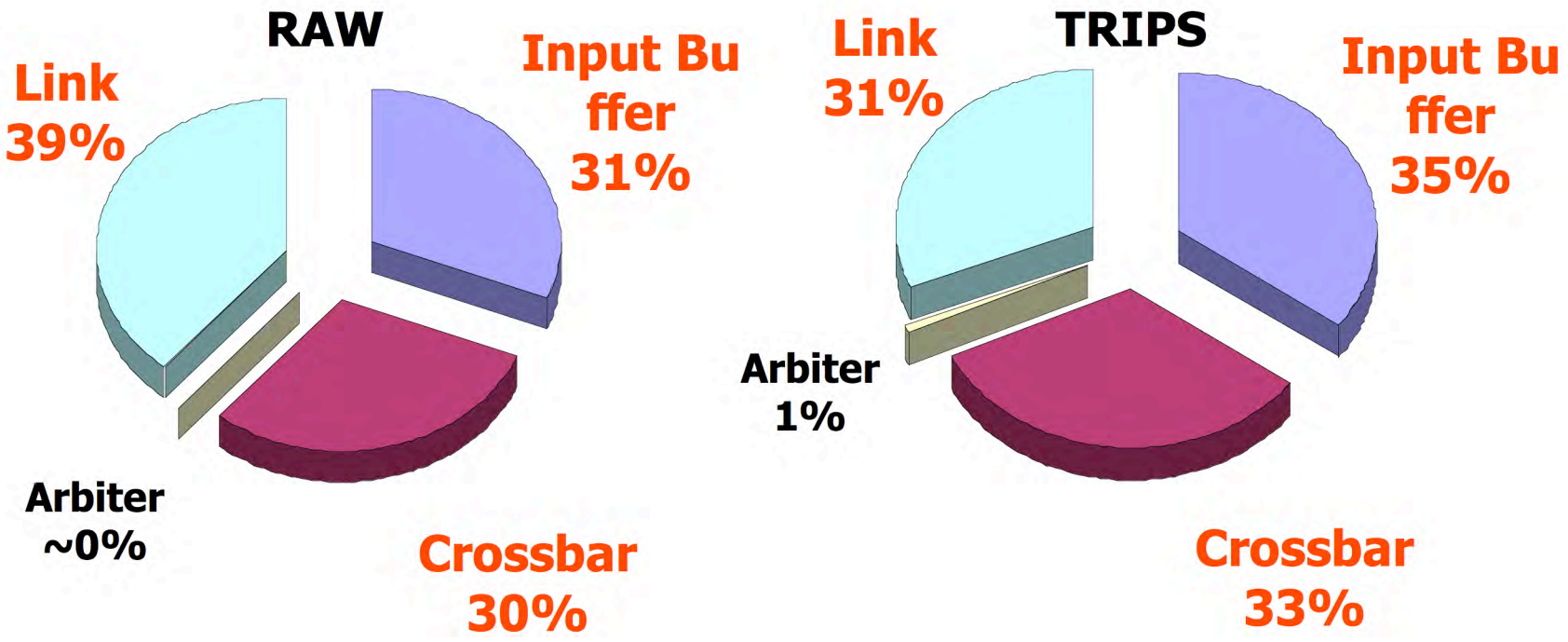
# *What* are Networks-on-Chip (NoC)?

**Processing Elements** (PEs) interconnected via a ***packet-based*** network



# Average Power Consumption in NoC Architectures

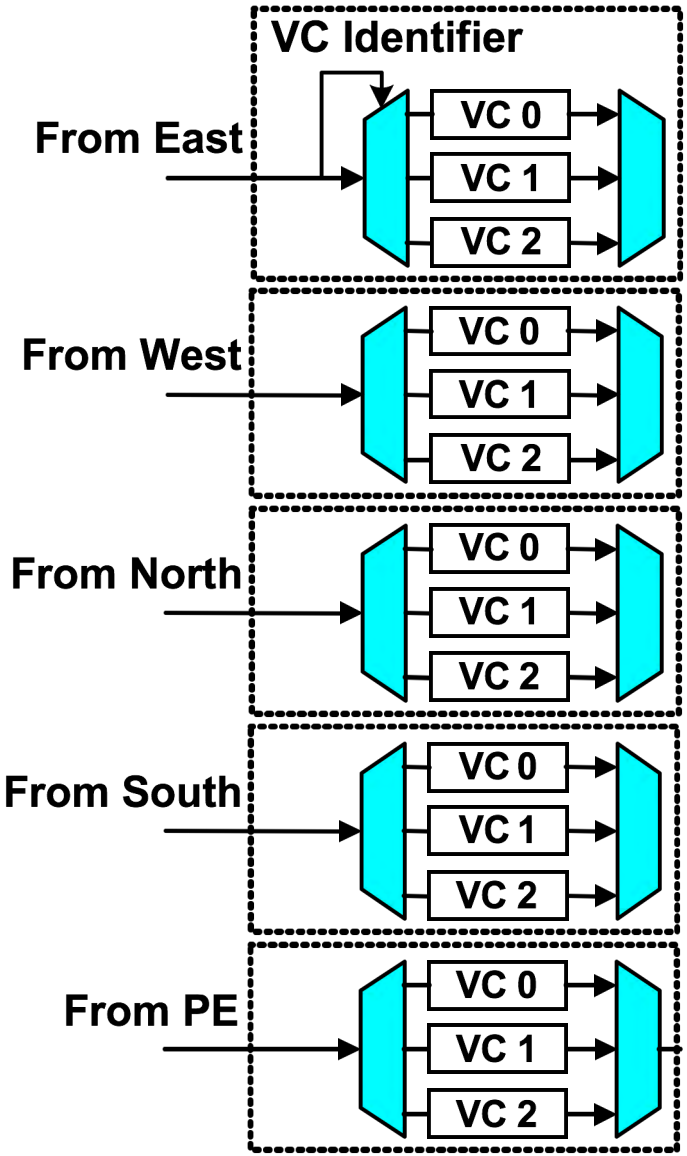
Note the contributions of: **(1) Crossbar, (2) Buffers, (3) Links**



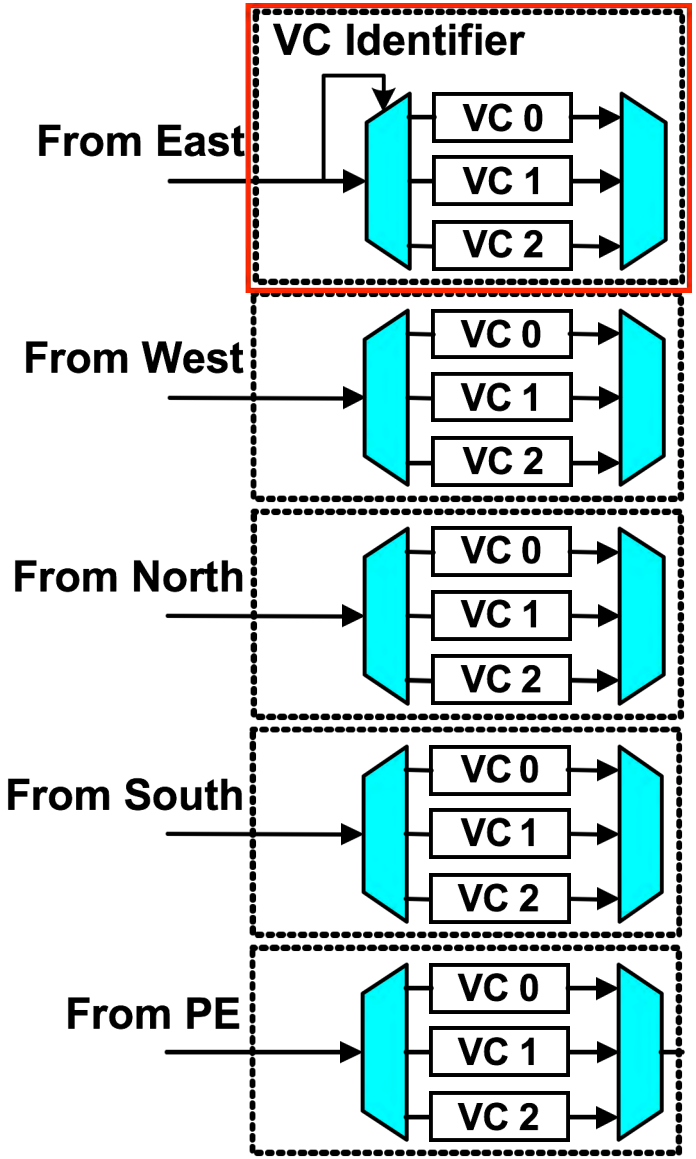

H.S.Wang & L.S.Peh, MICRO 2003

# A Conventional NoC Router

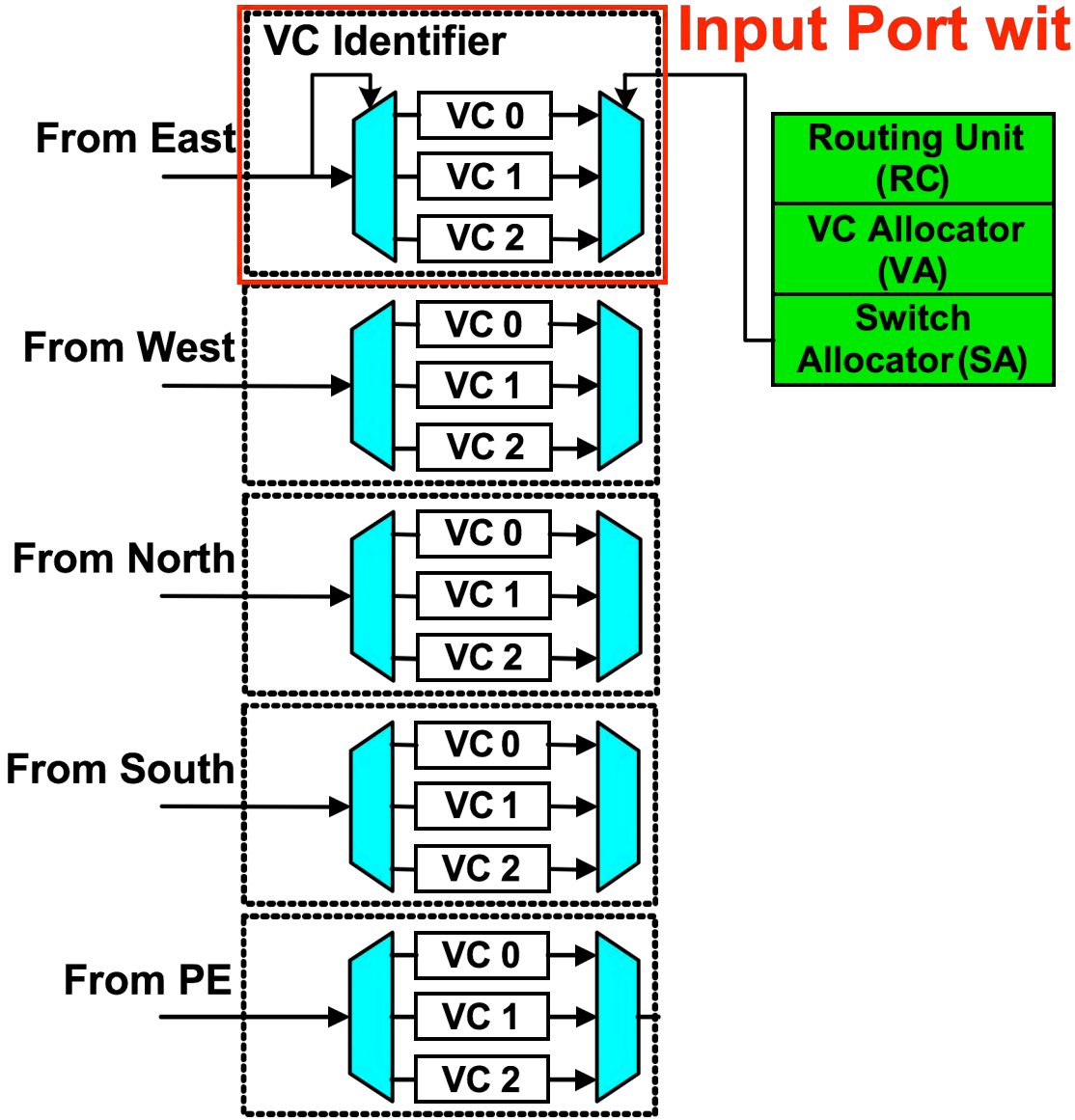
# A Conventional NoC Router



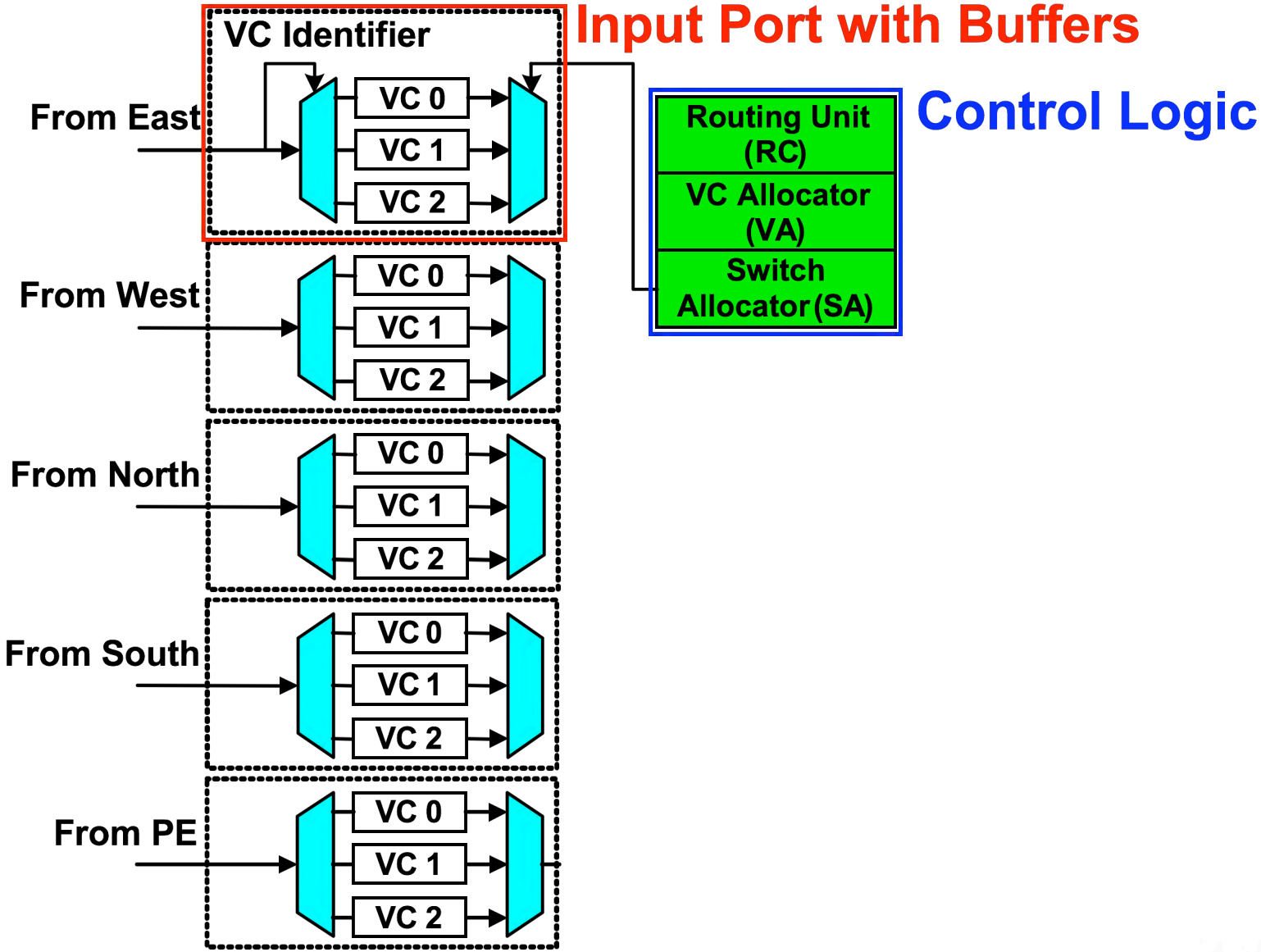
# A Conventional NoC Router



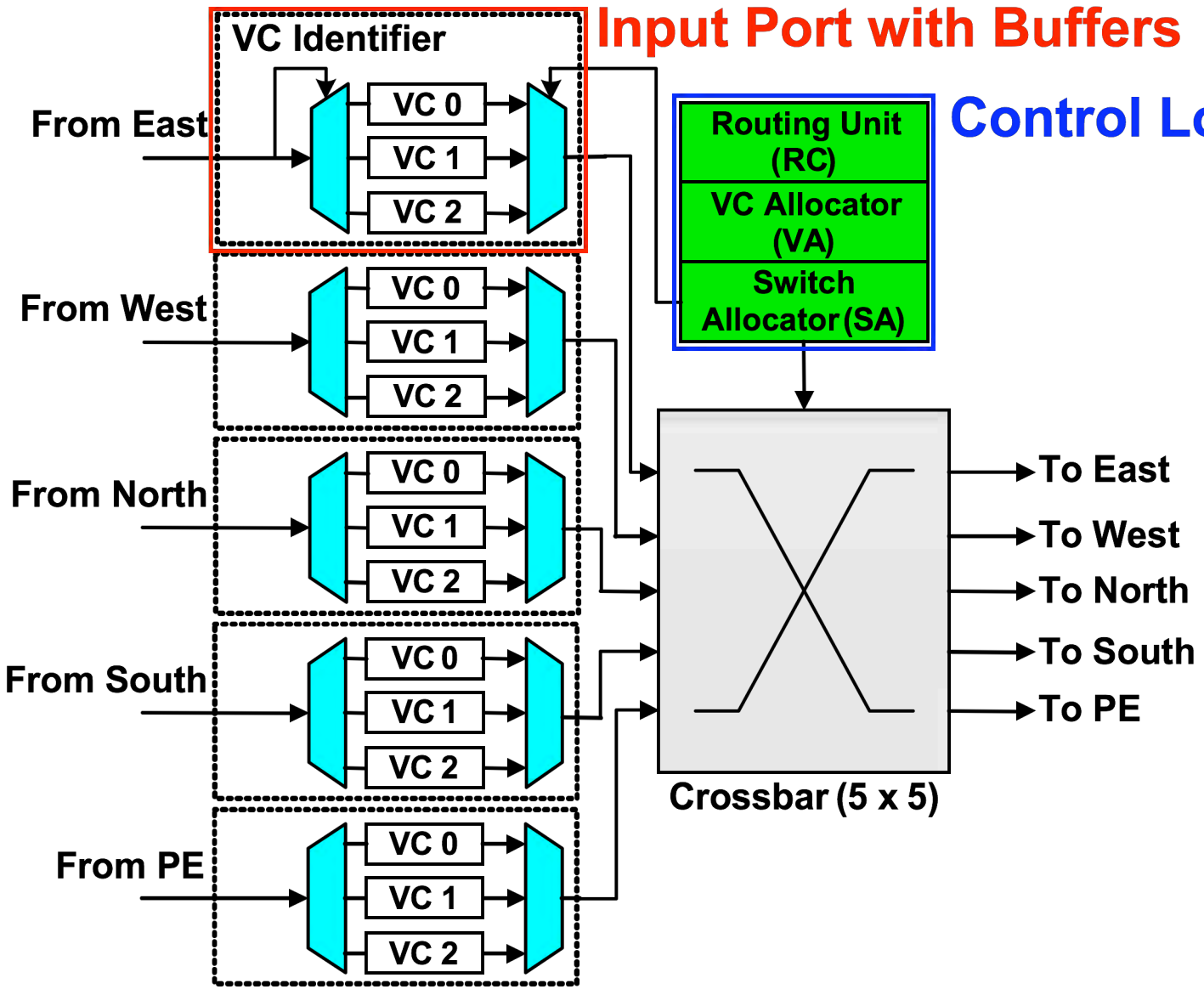
# A Conventional NoC Router



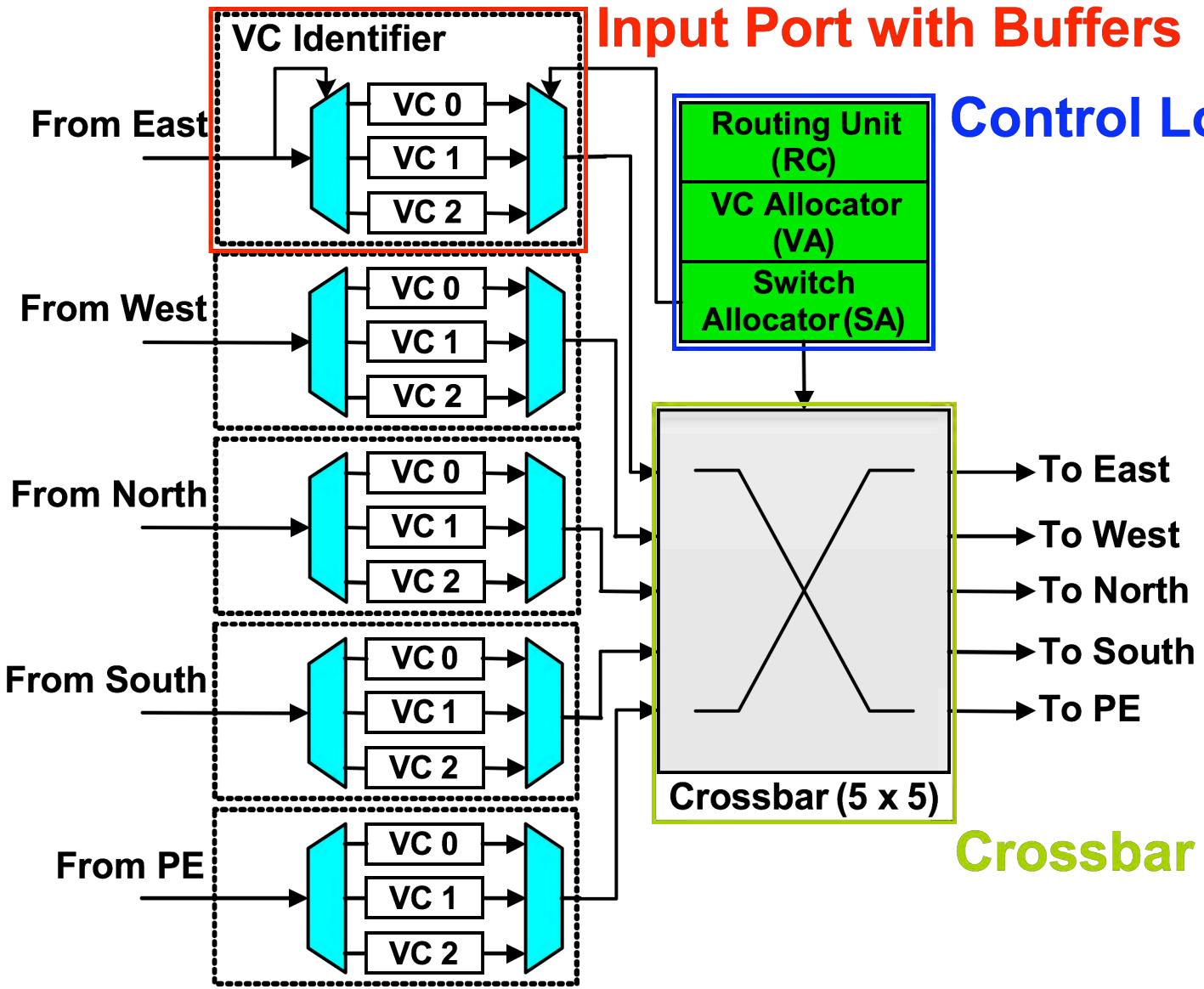
# A Conventional NoC Router



# A Conventional NoC Router



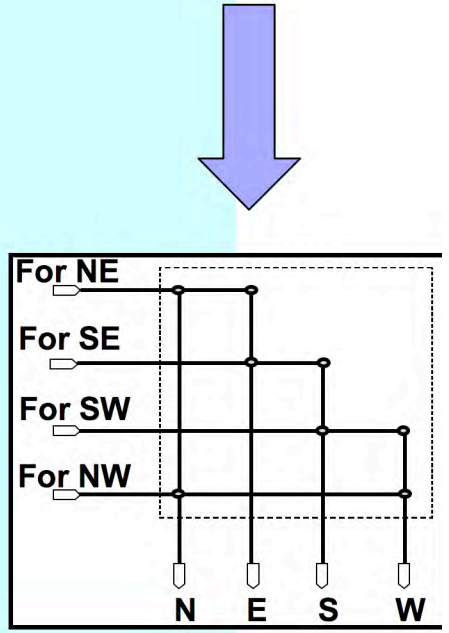
# A Conventional NoC Router



# The First Design Challenge: Reduce Crossbar Complexity: Path-Sensitive Router [DAC-05]

- Direction vector indicates destination quadrant (SW, SE, NW, NE) including required direction (S, N, E, W).
- *Previous node* sets which of the two quadrants (NE, SE) or PE it will go to
- Pre-Selection Mechanism in *current node* determines output port (N or E, S or E)

# The Path-Sensitive Router [DAC-05]



# The Row-Column (RoCo) Router [ISCA-06]

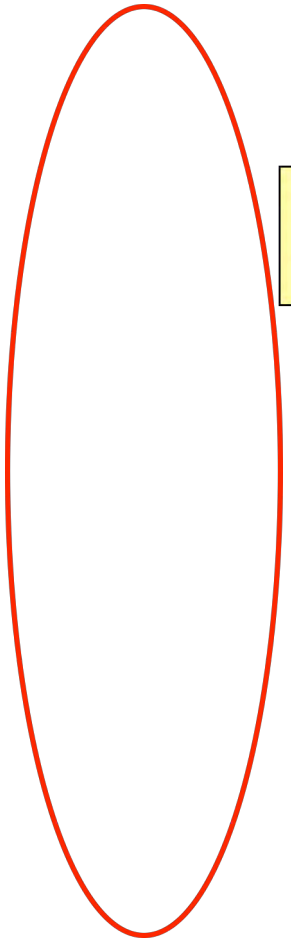
# The Row-Column (RoCo) Router [ISCA-06]

- Two **Smaller**, **Distinct** and **Independent** modules
- Smaller Crossbars (2 2x2 instead of 1 5x5)
- Partitioned Virtual Channels
- Guided Flit Queuing
- Early Ejection Mechanism
- Maximal Matching through Mirroring-Effect
- Inherent Fault-Tolerance
- Hardware Recycling Mechanism

# The RoCo Router Architecture

# The RoCo Router Architecture

# The RoCo Router Architecture



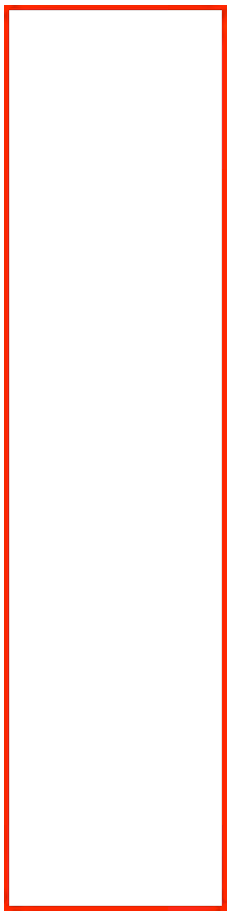
**Guided Flit Queuing**

# The RoCo Router Architecture

# The RoCo Router Architecture

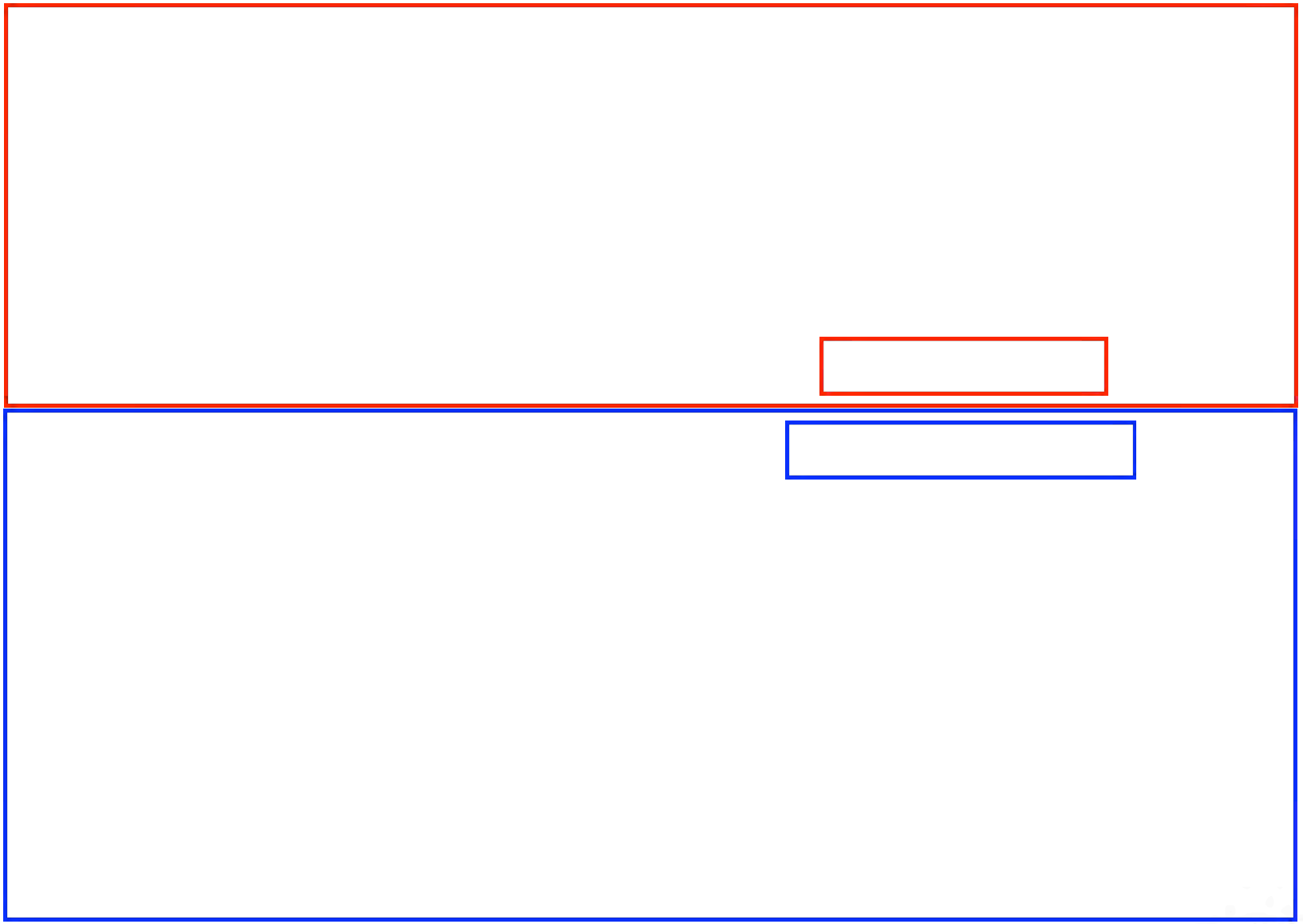
**Partitioned VCs**

$d_x$ = Continue on X
$d_y$ = Continue on Y
$t_{yx}$ = Turn from Y to X
$t_{xy}$ = Turn from X to Y
$Inj_{xy}$ = Injection into X
$Inj_{yx}$ = Injection into Y



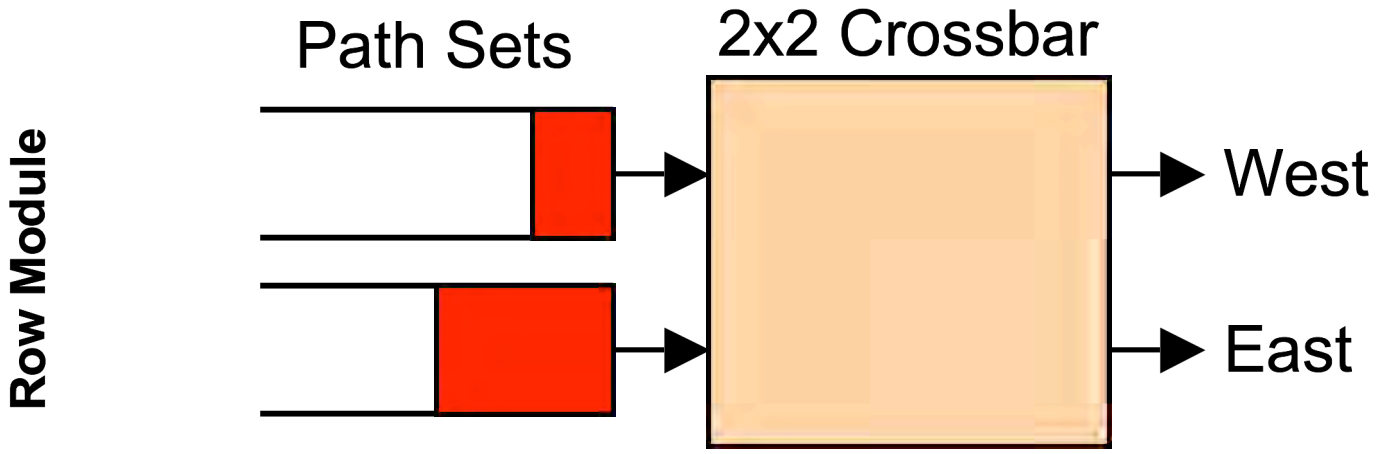
# The RoCo Router Architecture

# The RoCo Router Architecture

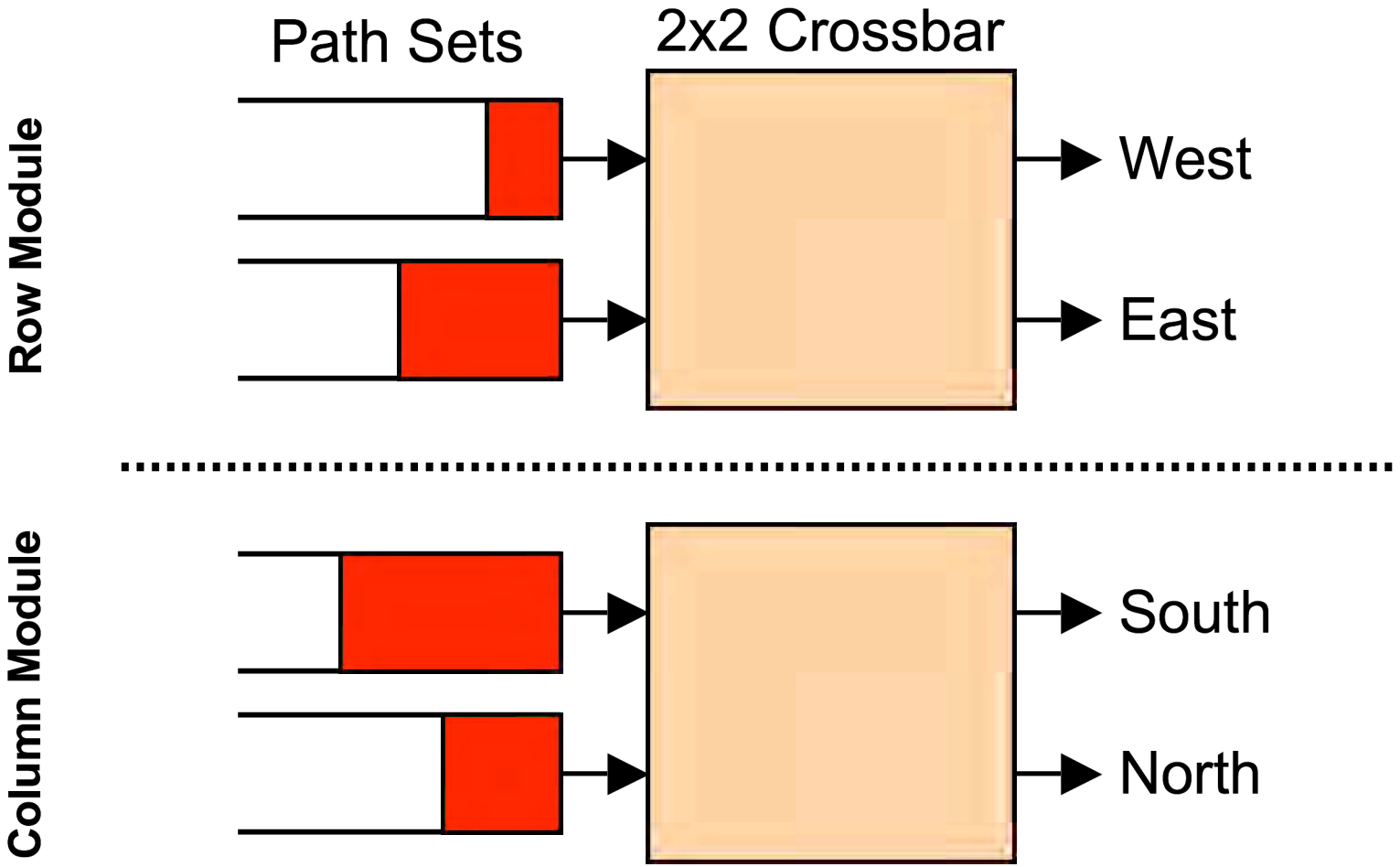


# Maximal Matching with the Mirroring Effect

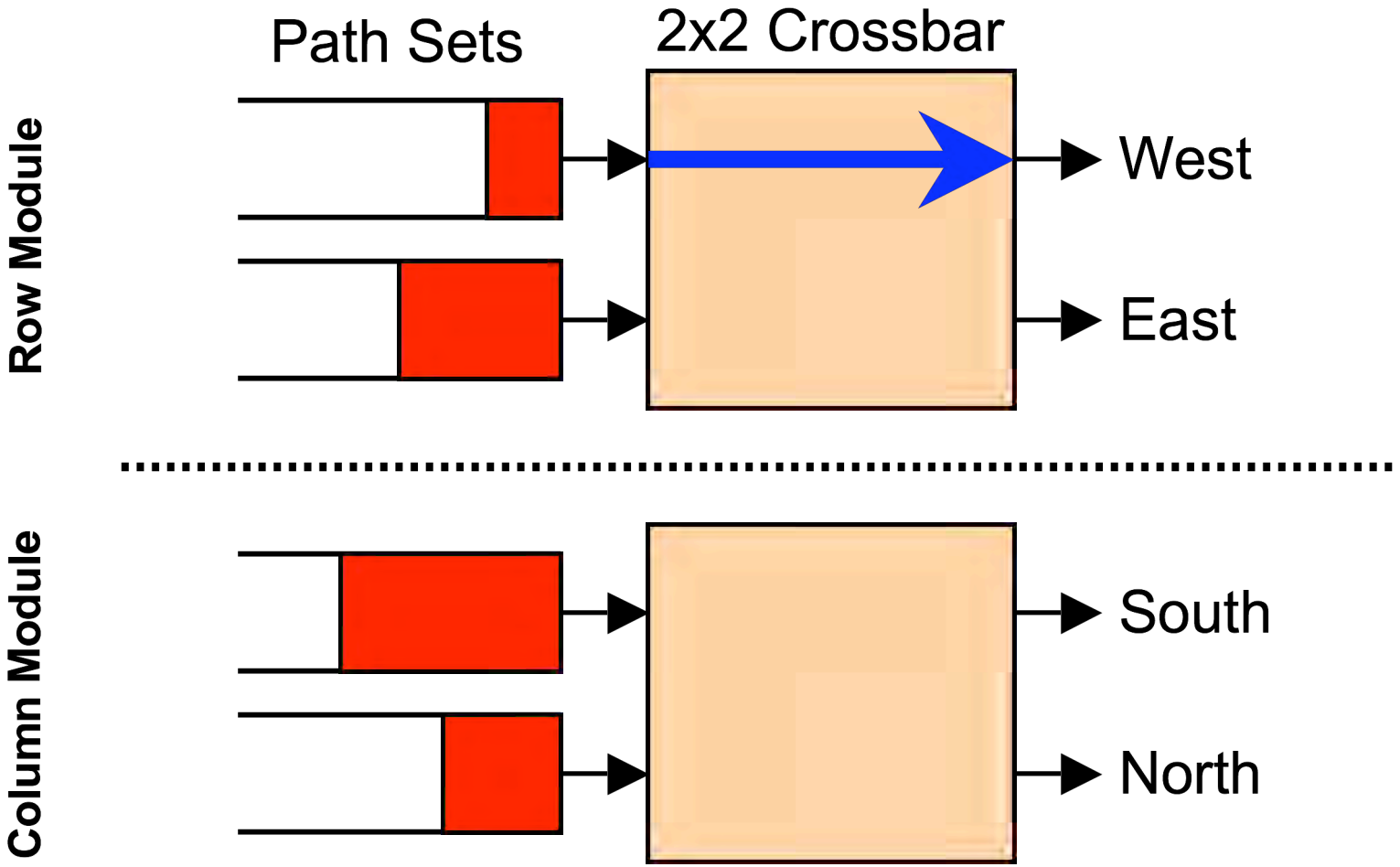
# Maximal Matching with the Mirroring Effect



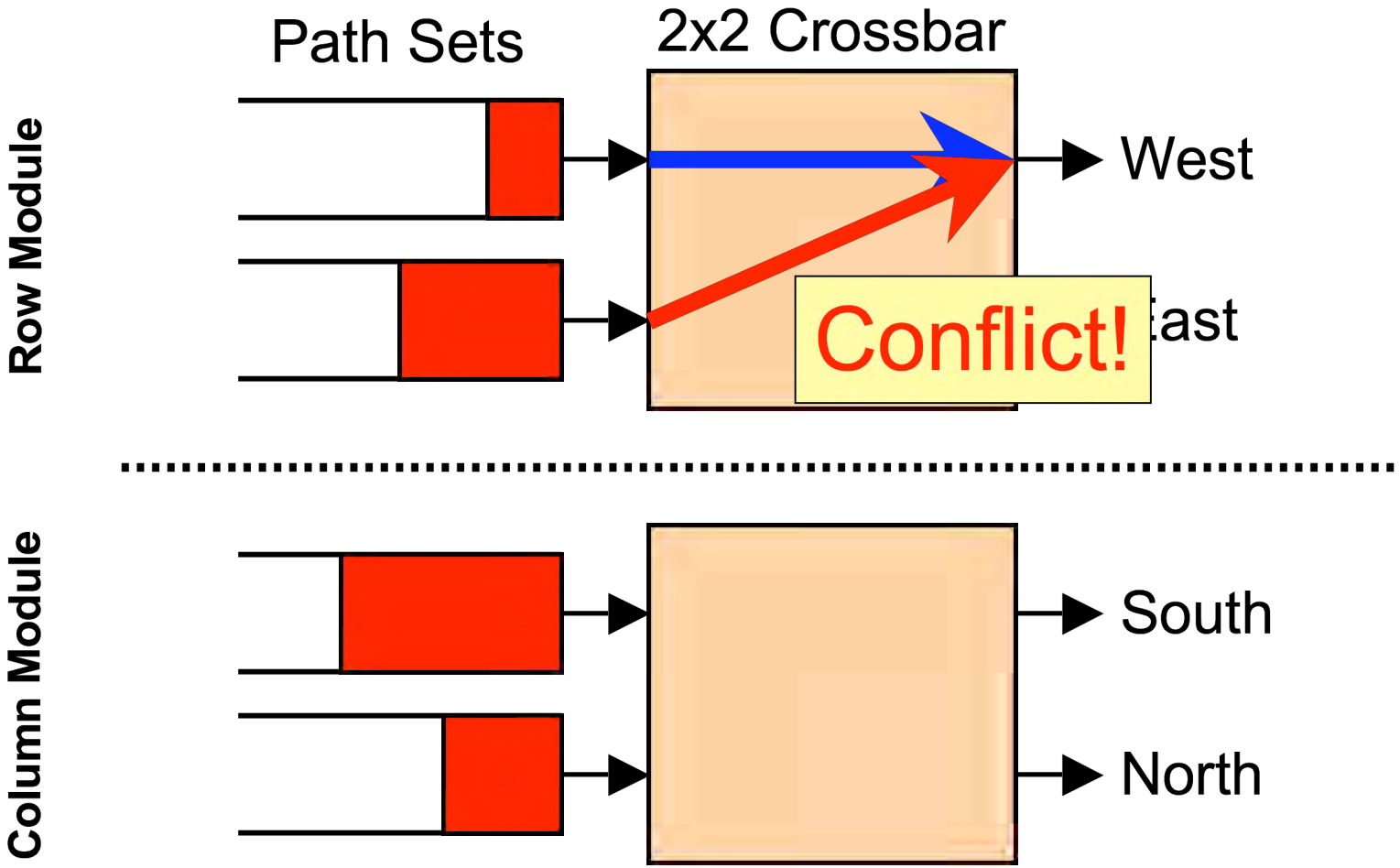
# Maximal Matching with the Mirroring Effect



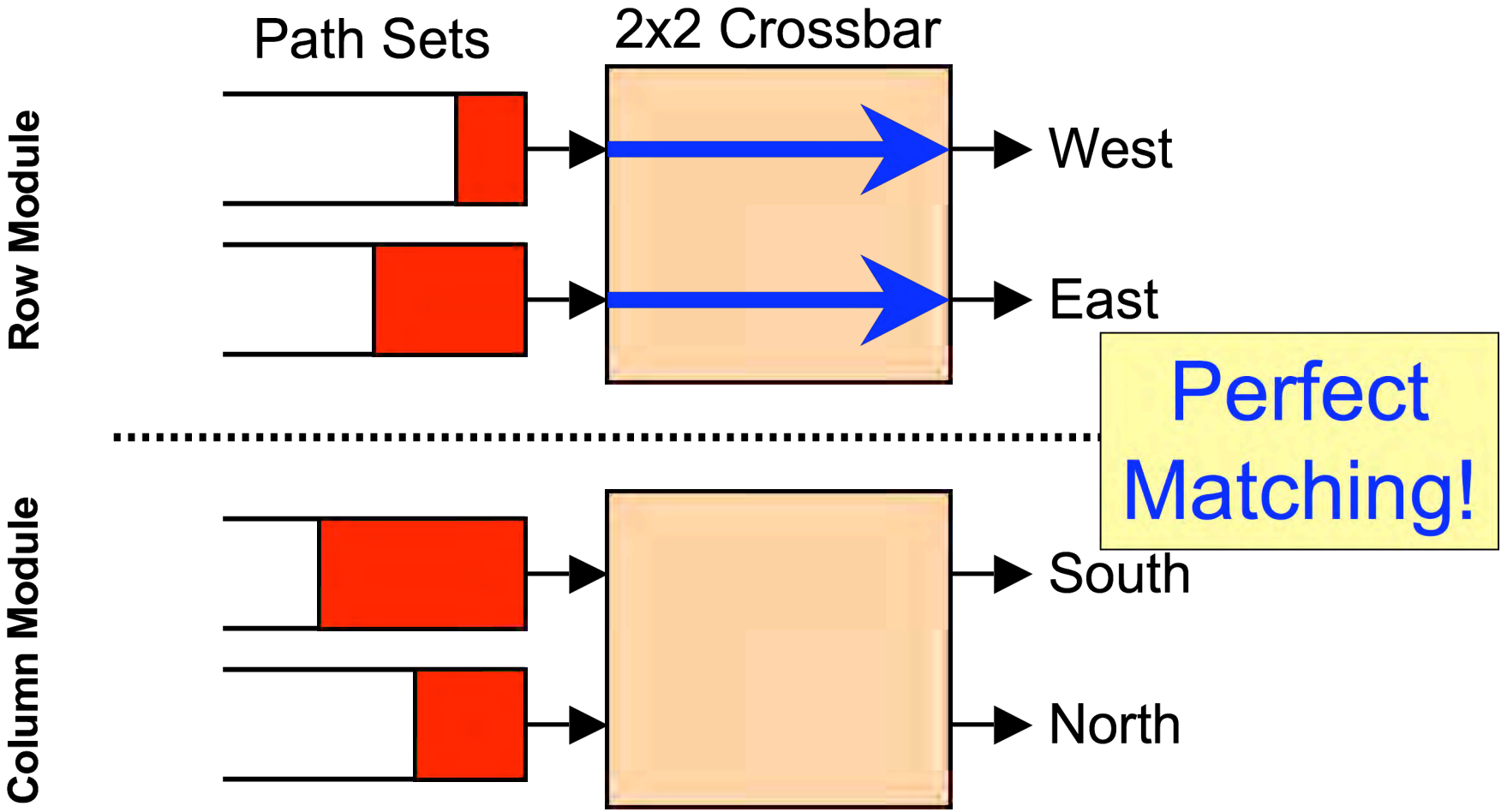
# Maximal Matching with the Mirroring Effect



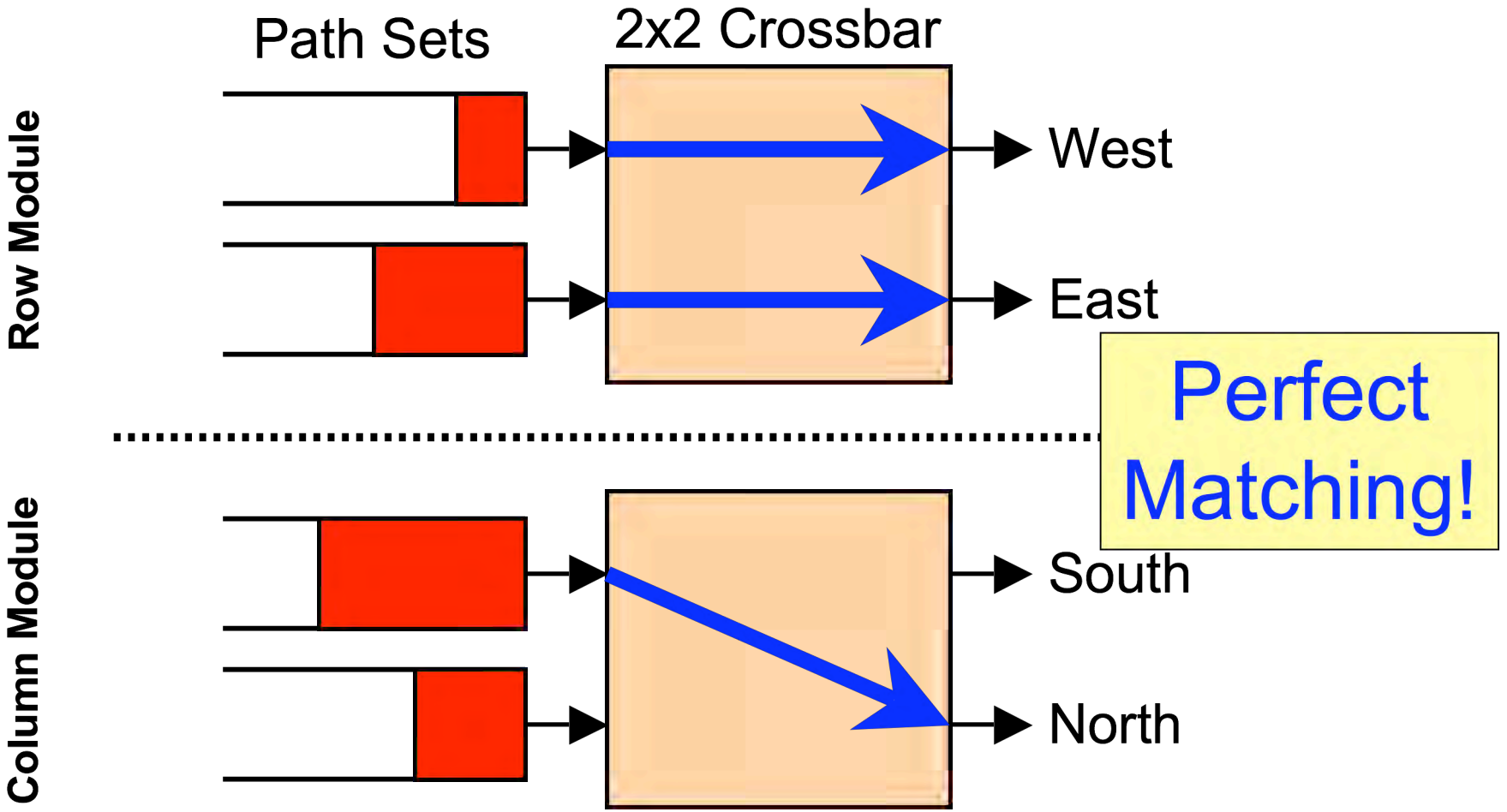
# Maximal Matching with the Mirroring Effect



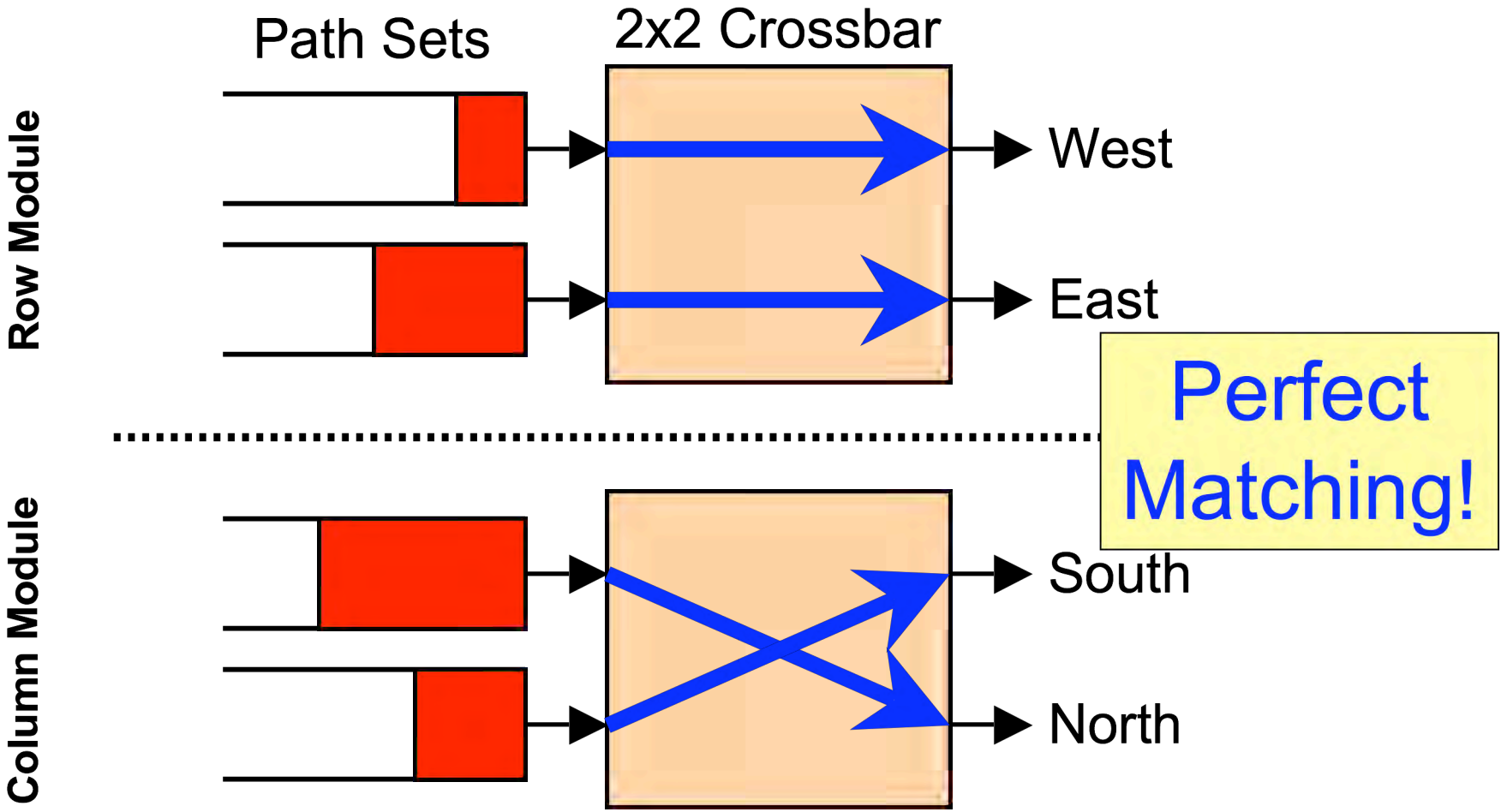
# Maximal Matching with the Mirroring Effect



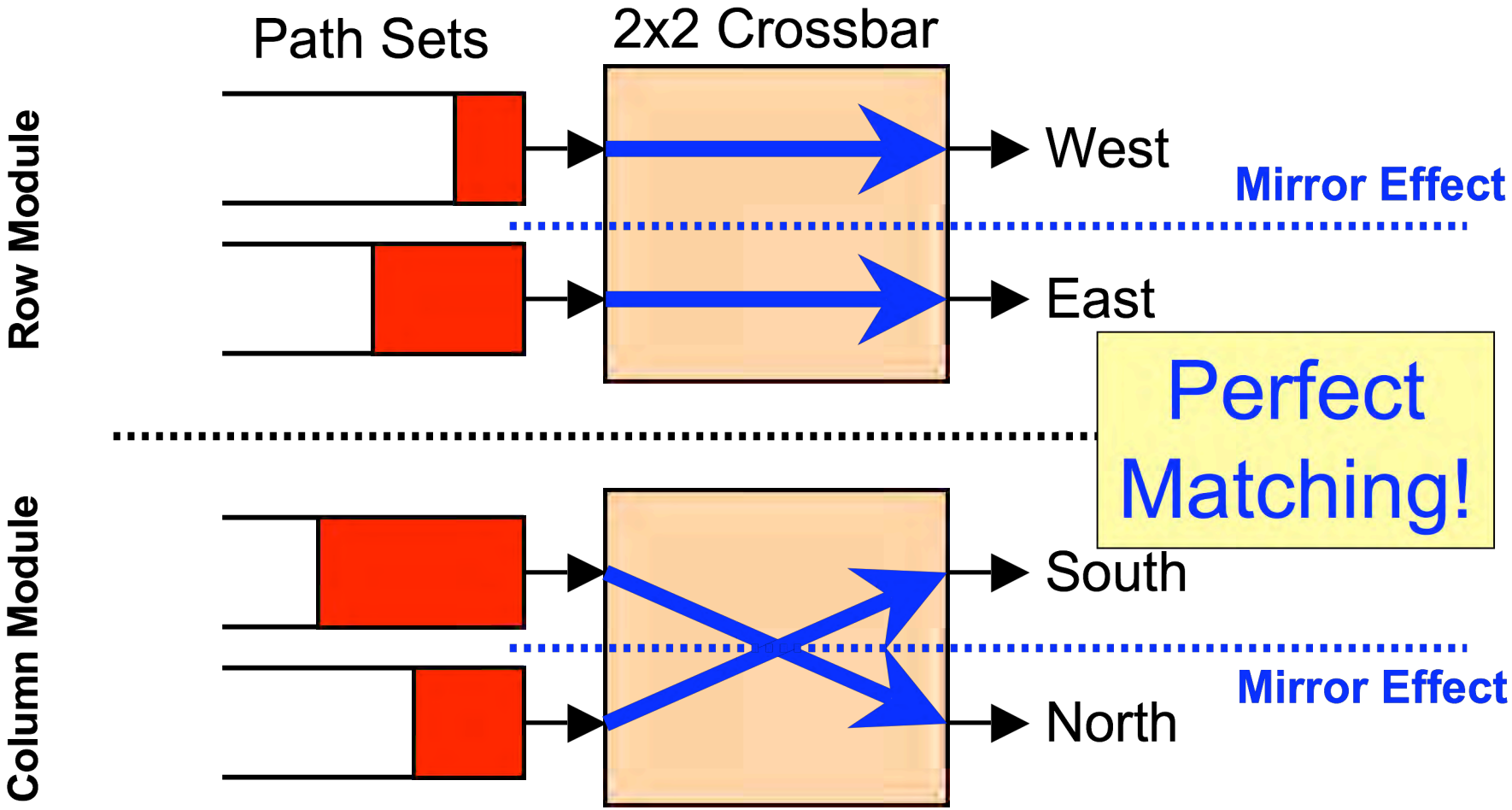
# Maximal Matching with the Mirroring Effect



# Maximal Matching with the Mirroring Effect



# Maximal Matching with the Mirroring Effect



# What about Fault-Tolerance?

# What about Fault-Tolerance?

- The RoCo Router has inherent fault-tolerant attributes due to its **DECOUPLED** operation

# What about Fault-Tolerance?

- The RoCo Router has inherent fault-tolerant attributes due to its **DECOUPLED** operation
- Two **SEPARATE** and **INDEPENDENT** modules

# What about Fault-Tolerance?

- The RoCo Router has inherent fault-tolerant attributes due to its **DECOUPLED** operation
- Two **SEPARATE** and **INDEPENDENT** modules
- Only faulty module is isolated

# What about Fault-Tolerance?

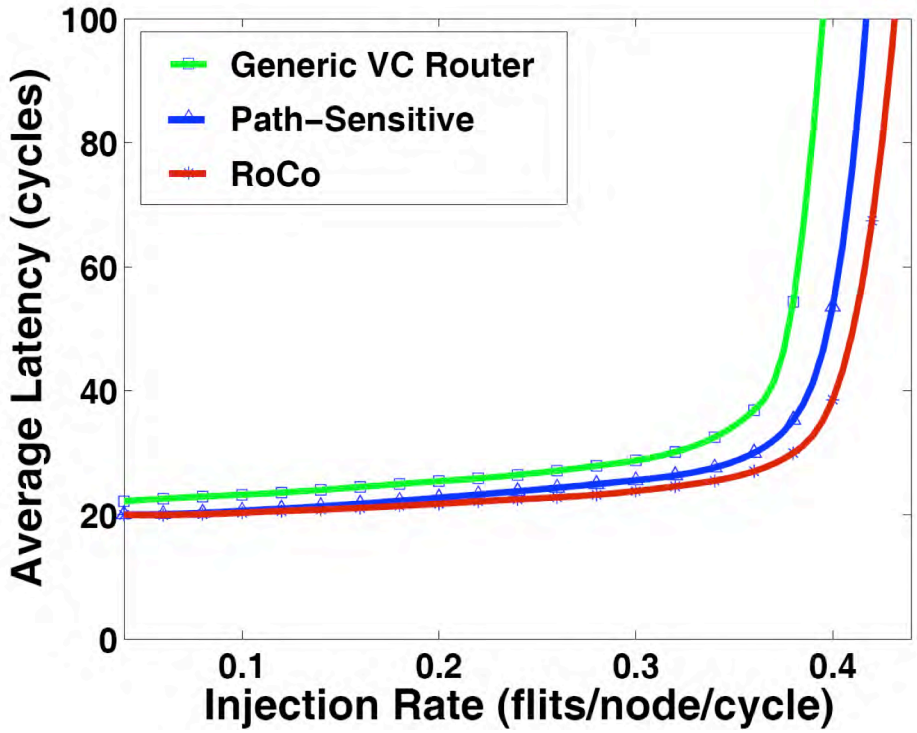
- The RoCo Router has inherent fault-tolerant attributes due to its **DECOUPLED** operation
- Two **SEPARATE** and **INDEPENDENT** modules
- Only faulty module is isolated
- Afflicted router can still handle limited network traffic

# What about Fault-Tolerance?

- The RoCo Router has inherent fault-tolerant attributes due to its **DECOUPLED** operation
- Two **SEPARATE** and **INDEPENDENT** modules
- Only faulty module is isolated
- Afflicted router can still handle limited network traffic
- Supports hardware recycling

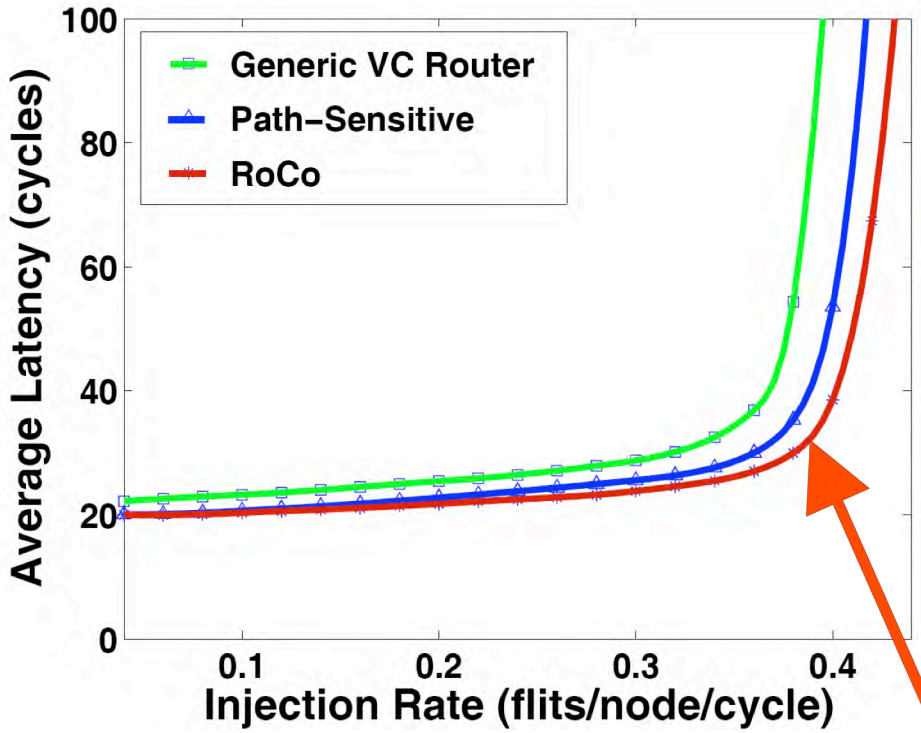
# Performance Analysis:
# Average Latency under Uniform Random Traffic

# Performance Analysis:
# Average Latency under Uniform Random Traffic




**Deterministic Routing**

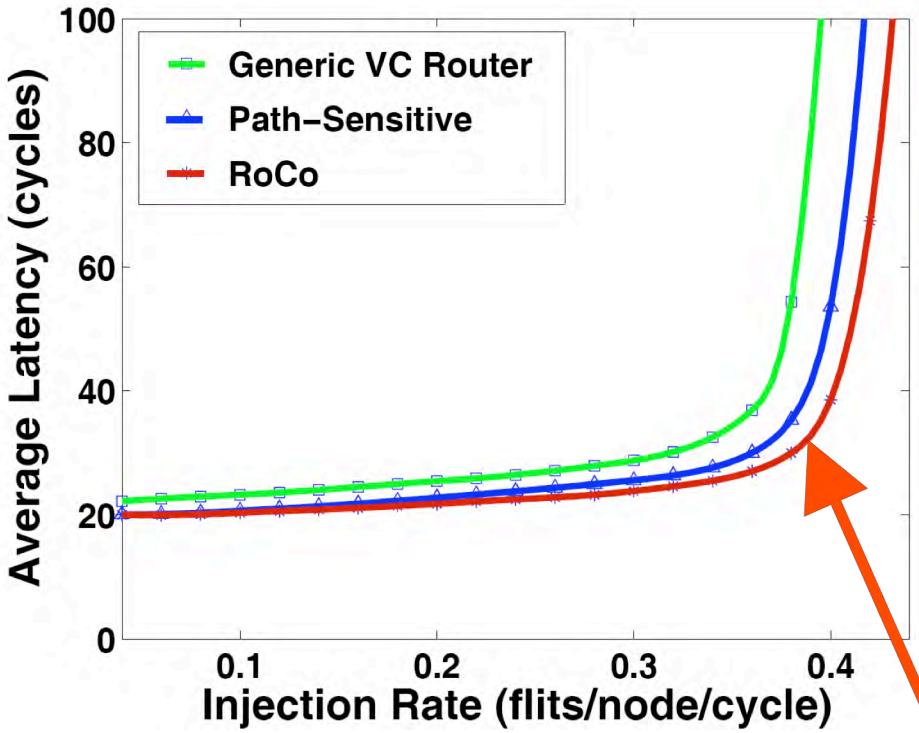
# Performance Analysis:
# Average Latency under Uniform Random Traffic

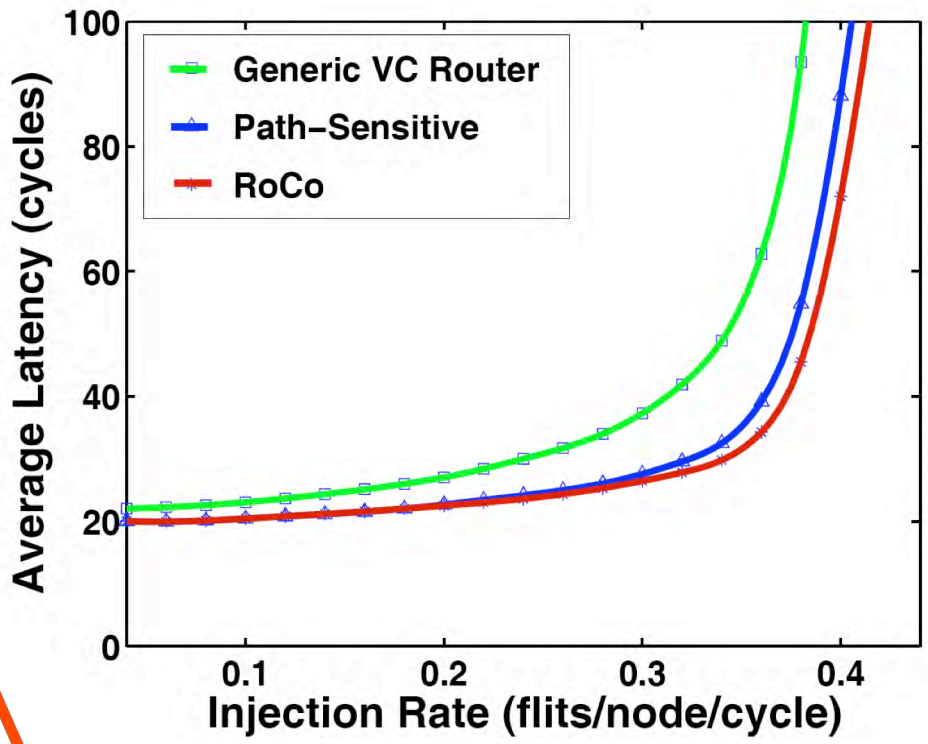



Deterministic Routing

RoCo

# Performance Analysis:
# Average Latency under Uniform Random Traffic



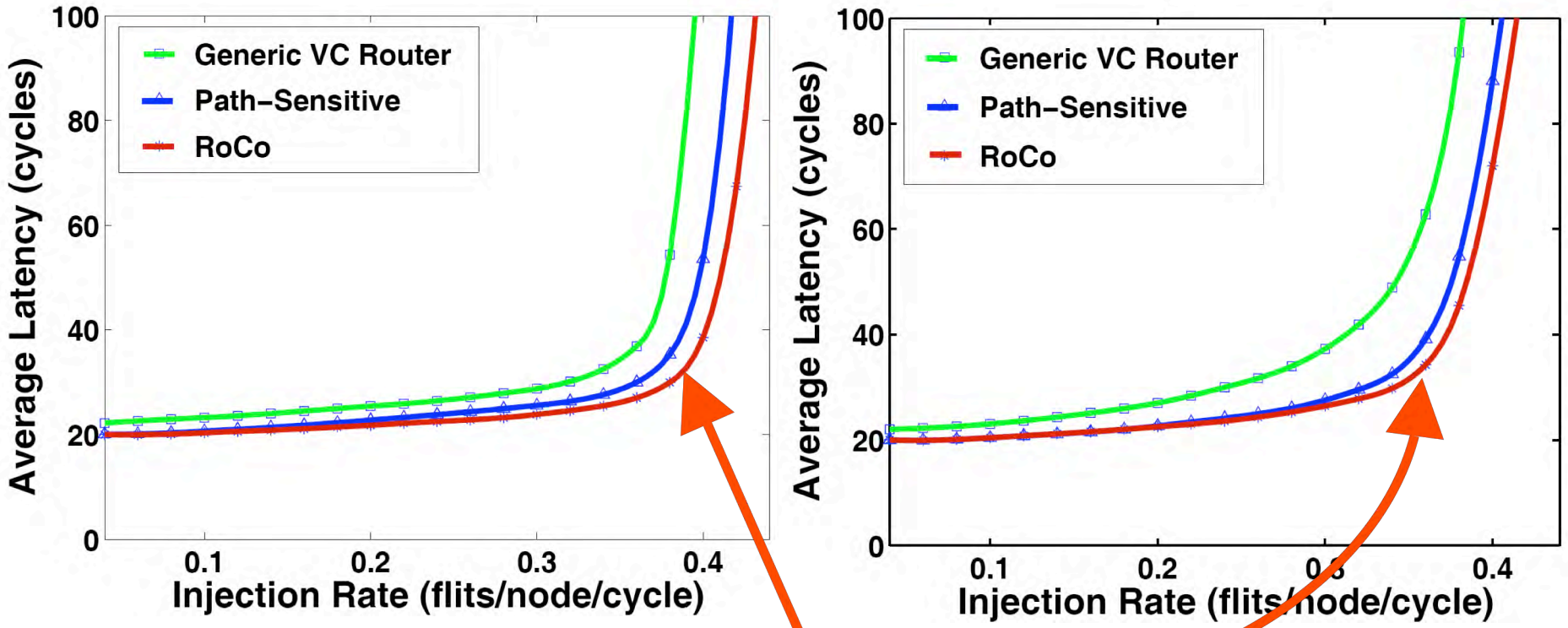




Deterministic Routing

RoCo

Adaptive Routing

# Performance Analysis:
# Average Latency under Uniform Random Traffic

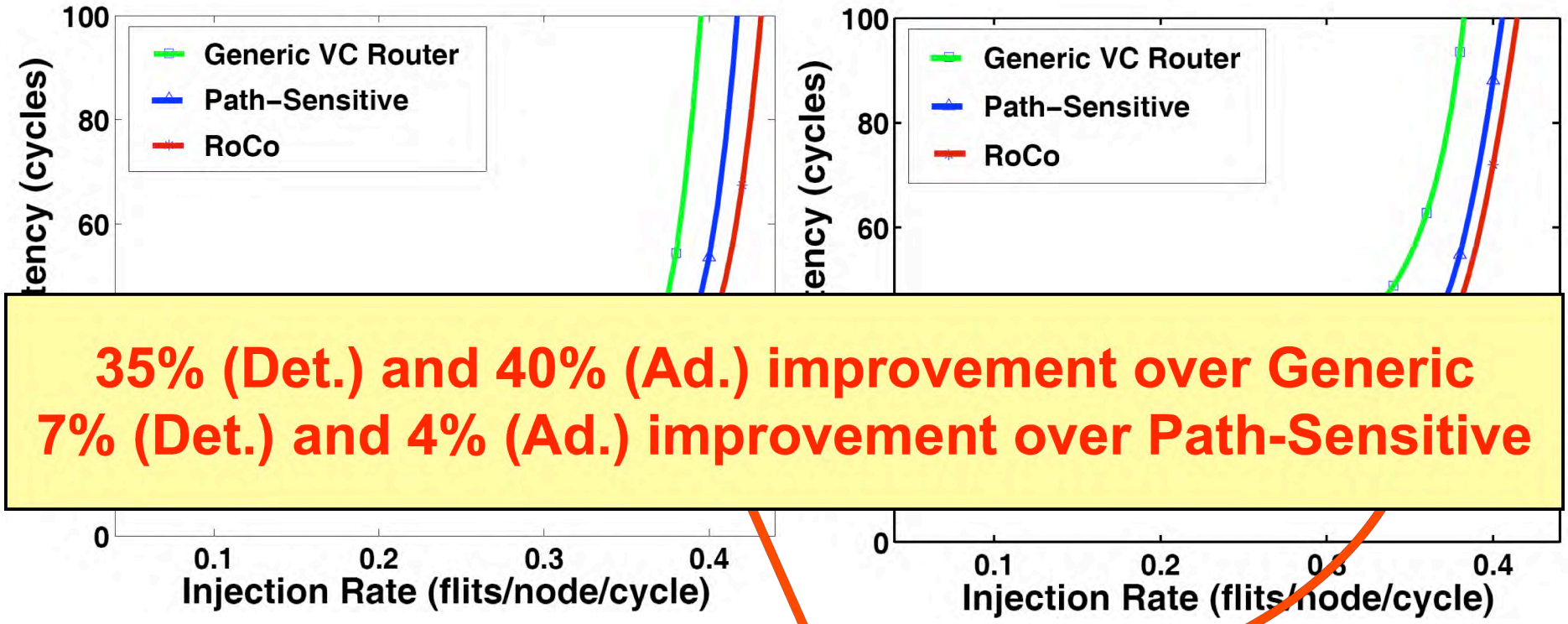



Deterministic Routing

RoCo

Adaptive Routing

# Performance Analysis:
# Average Latency under Uniform Random Traffic




**35% (Det.) and 40% (Ad.) improvement over Generic**
**7% (Det.) and 4% (Ad.) improvement over Path-Sensitive**

Deterministic Routing

RoCo

Adaptive Routing

# Performance Analysis :
# Packet Completion Probabilities in the Presence of Faults

# Performance Analysis : Packet Completion Probabilities in the Presence of Faults

- Faults injected randomly
  - Generic and Path-Sensitive: Entire node blocked
  - RoCo: Only one module blocked

# Performance Analysis : Packet Completion Probabilities in the Presence of Faults

- Faults injected randomly
  - Generic and Path-Sensitive: Entire node blocked
  - RoCo: Only one module blocked

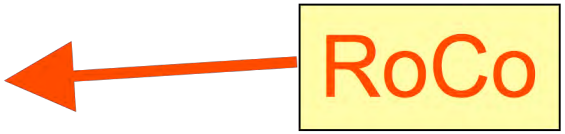
**Deterministic Routing**

# Performance Analysis : Packet Completion Probabilities in the Presence of Faults

- Faults injected randomly
  - Generic and Path-Sensitive: Entire node blocked
  - RoCo: Only one module blocked

Higher is better!

Injection Rate in faulty network = 30%

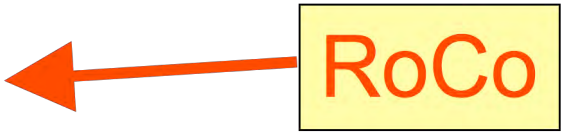



**Deterministic Routing**

# Performance Analysis : Packet Completion Probabilities in the Presence of Faults

- Faults injected randomly
  - Generic and Path-Sensitive: Entire node blocked
  - RoCo: Only one module blocked

Higher is better!




Injection Rate in faulty network = 30%

Deterministic Routing

Adaptive Routing

# Performance Analysis : Packet Completion Probabilities in the Presence of Faults

- Faults injected randomly
  - Generic and Path-Sensitive: Entire node blocked
  - RoCo: Only one module blocked

Higher is better!

Injection Rate in faulty network = 30%


**70% improvement with Det. Routing**
**7% improvement with Adapt. Routing**

**Deterministic Routing** **Adaptive Routing**

# Performance Analysis:
# Energy and PEF Metric Results

# Performance Analysis: Energy and PEF Metric Results

Lower is better!

Injection Rate in network = 30%

**Energy Per Packet**

# Performance Analysis: Energy and PEF Metric Results

Lower is better!

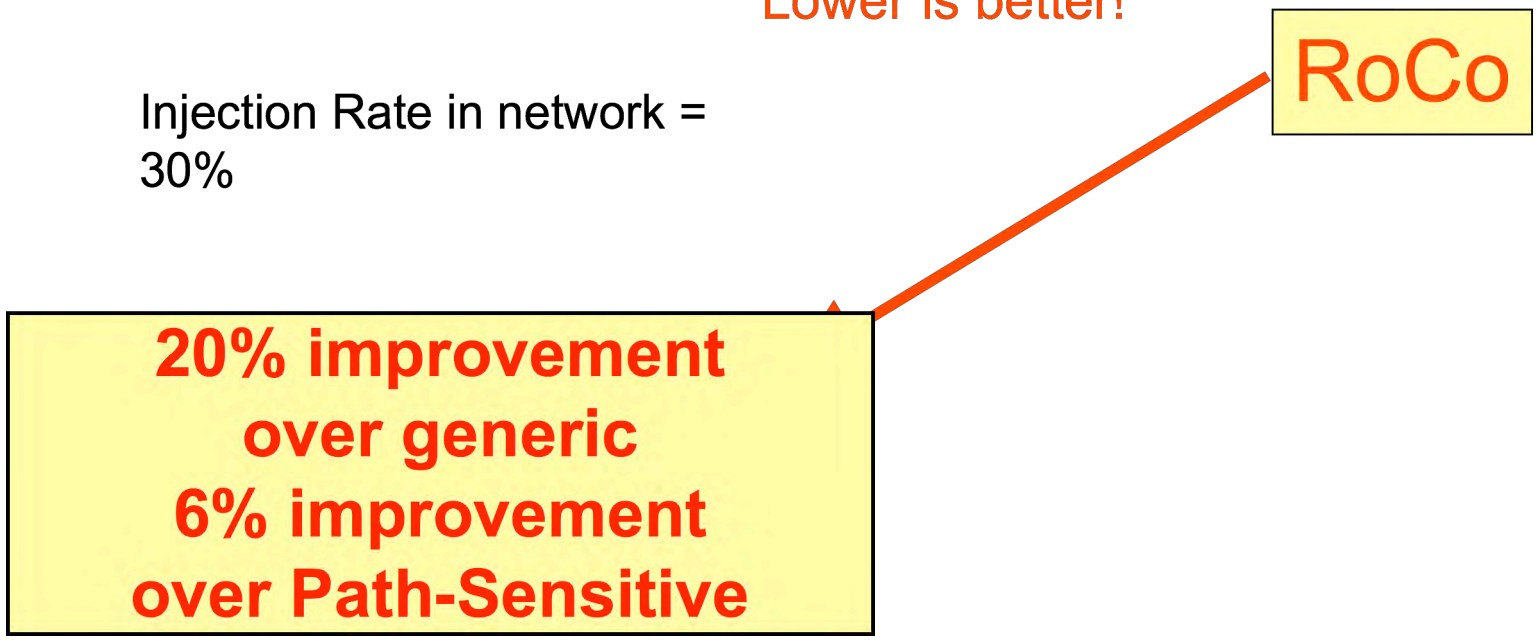
Injection Rate in network = 30%


**Energy Per Packet**

# Performance Analysis: Energy and PEF Metric Results

Lower is better!

Injection Rate in network = 30%

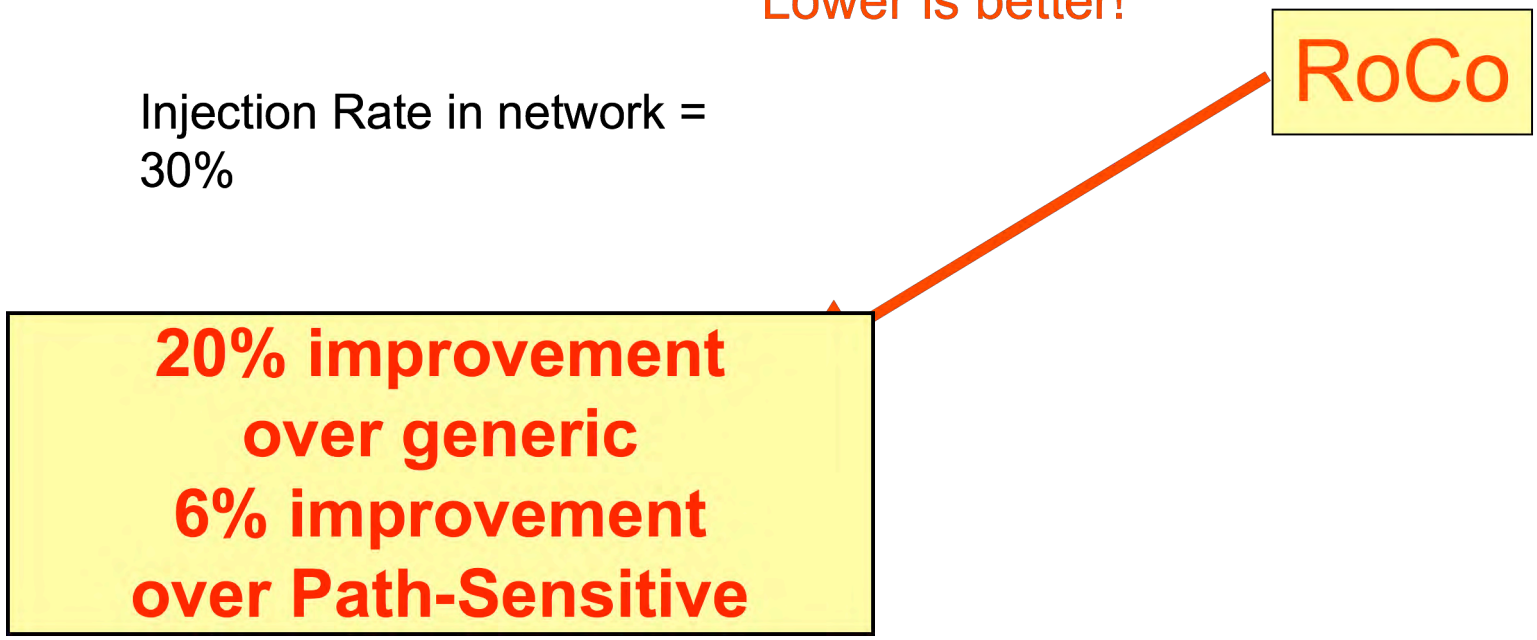



Energy Per Packet

# Performance Analysis: Energy and PEF Metric Results

Lower is better!

Injection Rate in network = 30%



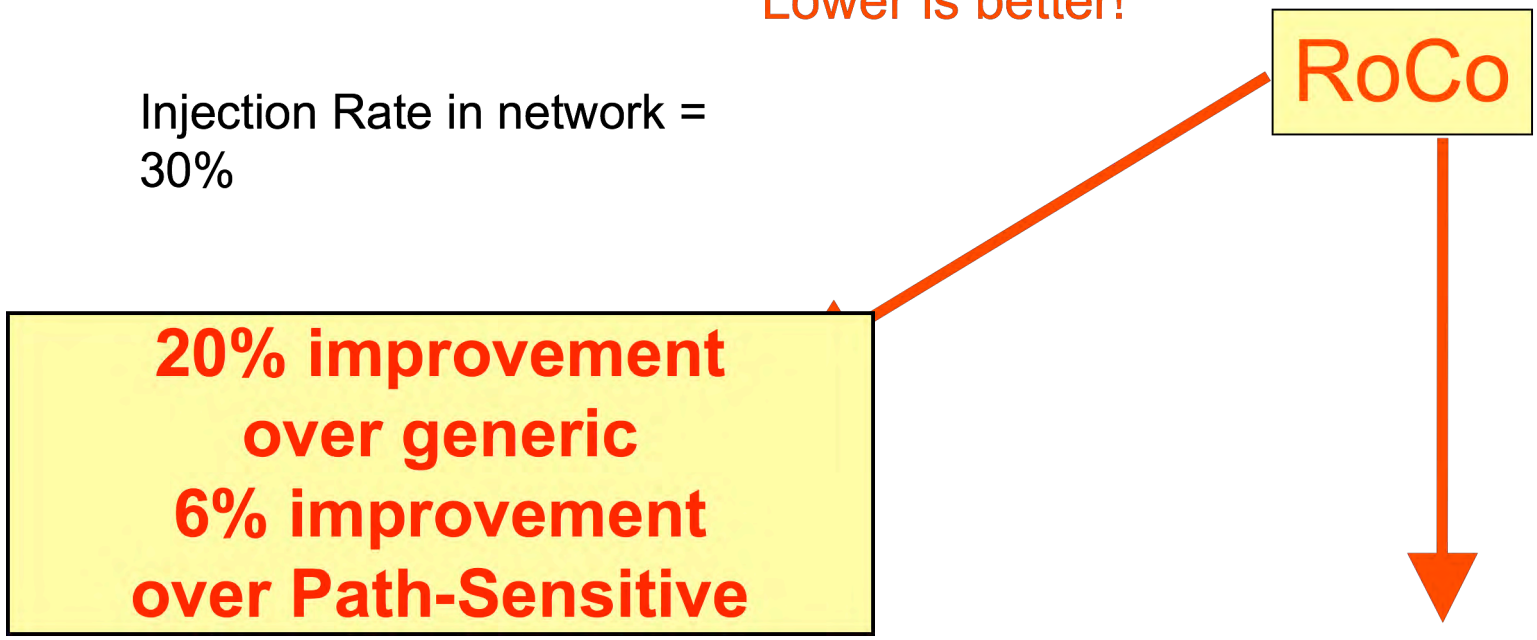

Energy Per Packet

Performance-Energy-Fault (PEF) Metric

# Performance Analysis: Energy and PEF Metric Results

Lower is better!

Injection Rate in network = 30%



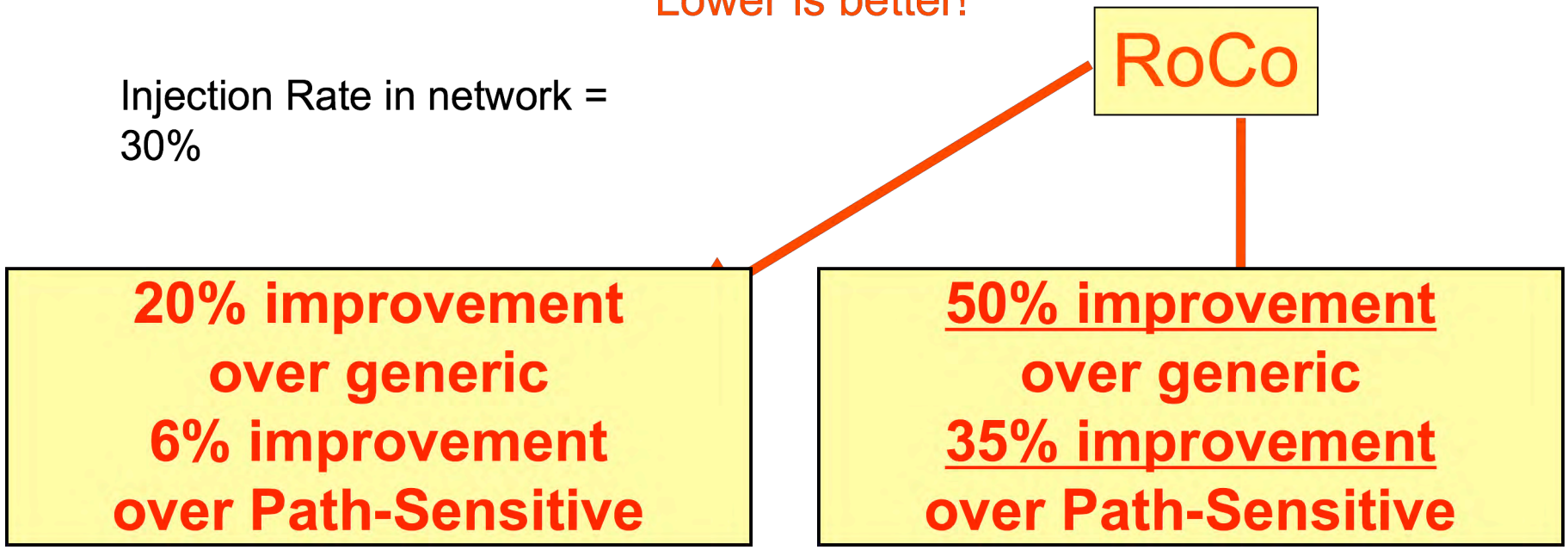

Energy Per Packet

Performance-Energy-Fault (PEF) Metric

# Performance Analysis: Energy and PEF Metric Results

Lower is better!

Injection Rate in network = 30%




Energy Per Packet

Performance-Energy-Fault (PEF) Metric

# The Second Design Challenge: **The NoC buffers**

- The NoC Buffers **DOMINATE** the Area and Power budgets of the router!
- Any improvements in the on-chip buffers will yield significant benefits in the overall interconnect system.
- Existing on-chip buffer solutions suffer from a number of crippling limitations (such as Head-of-Line Blocking and Underutilization)

# The Second Design Challenge:
# The NoC buffers

Solution:

**ViChaR: A Dynamic Virtual Channel Regulator for Network-on-Chip Routers**

To appear at the

**39th Annual International Symposium on Microarchitecture (MICRO)**

December 2006

# **ViChaR**: A Dynamic Virtual Channel Regulator for NoC Routers

- ViChaR's operation revolves around two fundamental concepts:
  - **ViChaR uses a Unified Buffer Structure (UBS)**
  - **ViChaR provides each individual router port with a variable number of VCs**

# **ViChaR**: A Dynamic Virtual Channel Regulator for NoC Routers

- ViChaR's operation revolves around two fundamental concepts:
  - **ViChaR uses a Unified Buffer Structure (UBS)**
  - **ViChaR provides each individual router port with a variable number of VCs**

# **ViChaR** – Average Latency

# ViChaR – Average Latency

GEN = Generic Router
**ViC = ViChaR Router**
NR = Normal Random (Source-Destination Selection)
TN = Tornado (Source-Destination Selection)
UR = Uniform Random Traffic

**Average Latency (Deterministic Routing)**

# ViChaR – Average Latency

GEN = Generic Router
**ViC = ViChaR Router**
NR = Normal Random (Source-Destination Selection)
TN = Tornado (Source-Destination Selection)
UR = Uniform Random Traffic


**Average Latency (Deterministic Routing)**

# ViChaR – Average Latency

GEN = Generic Router
**ViC = ViChaR Router**
NR = Normal Random (Source-Destination Selection)
TN = Tornado (Source-Destination Selection)
UR = Uniform Random Traffic


**~30% Improvement (Deterministic Routing)**
**~25% Improvement (Minimal Adaptive Routing)**
over a Conventional Router

# **ViChaR** – Average Latency

# **ViChaR** – Average Latency

**Average Latency (Deterministic Routing)**

# **ViChaR** – Average Latency

**Average Latency (Deterministic Routing)**

# **ViChaR** – Average Latency

Average Latency (Deterministic Routing)

- **ViChaR's efficiency allows us to *halve* the buffer resources with no discernible effect on performance.**

# **ViChaR** – Power Improvements

# **ViChaR** – Power Improvements

# **ViChaR** – Power Improvements

- For equally sized configurations, ViChaR consumes slightly more power than a conventional buffer structure (GEN-16 vs. ViC-16).

# **ViChaR** – Power Improvements

- For equally sized configurations, ViChaR consumes slightly more power than a conventional buffer structure (GEN-16 vs. ViC-16).
- However, since **ViChaR's efficiency allows us to *halve* the buffer resources with no discernible effect on performance**, the **overall power drops by about 34%** (GEN-16 vs. ViC-8) for equivalent performance.

# The Third Design Challenge:
# Reducing Wire Length

# The Third Design Challenge: Reducing Wire Length

- **Three-dimensional integration** (3D IC) is an attractive option for interconnect scaling.

# The Third Design Challenge: Reducing Wire Length

- **Three-dimensional integration** (3D IC) is an attractive option for interconnect scaling.
- **Direct vertical tunnel** reduces global interconnects.

# The Third Design Challenge: Reducing Wire Length

- **Three-dimensional integration** (3D IC) is an attractive option for interconnect scaling.
- **Direct vertical tunnel** reduces global interconnects.
- **A combination of the 3D technology and NoC** design can be used to provide scalable and efficient on-chip networks.

# A 3D Symmetric NoC Architecture

Simplest Extension to the Baseline NoC Router to facilitate a 3D layout:

- 3D Symmetric NoC Architecture
- **Hop-by-Hop traversal**: implemented by **2D Crossbar**.

# 3D NoC-Bus Hybrid Architecture

**Hybridized with a bus link** in the vertical dimension:

Given the **very small inter-strata distance**, **single hop communication** is feasible.

**NoC-Bus Hybrid Architecture**

**Inter-Layer Via Structure In a 3D NoC-Bus Hybrid**

# A Full 3D NoC Router

Vertical Links are embedded in 3D crossbar switch:

- **Seamless integration** of the vertical links in a single router operation.
- **Multiple internal paths** and no intermediate buffers – go through a couple of crossbar switching points and directly connect to the output port of the destination layer.

# Inter-Layer Via Structure in a 3D Crossbar Technology

3D connection box can facilitate linkage between vertical and horizontal channels, and vertical pillars are segmented for flexible flit traversal.

Connection Box

Inter-Layer Via Layout
(Vertical Link Segmentation)

A 3D 3x3x3 Crossbar Concep

# DimDe: A Dimensionally-Decomposed 3D NoC Router Architecture

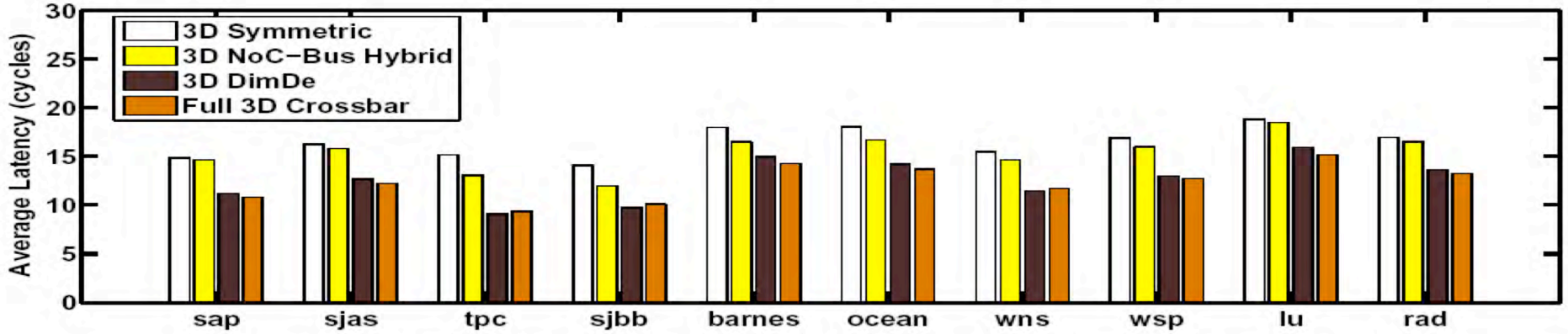
Full 3D crossbar requires complex arbitration
and enormous number of vertical links and control signals.

# DimDe Router (Contd.)

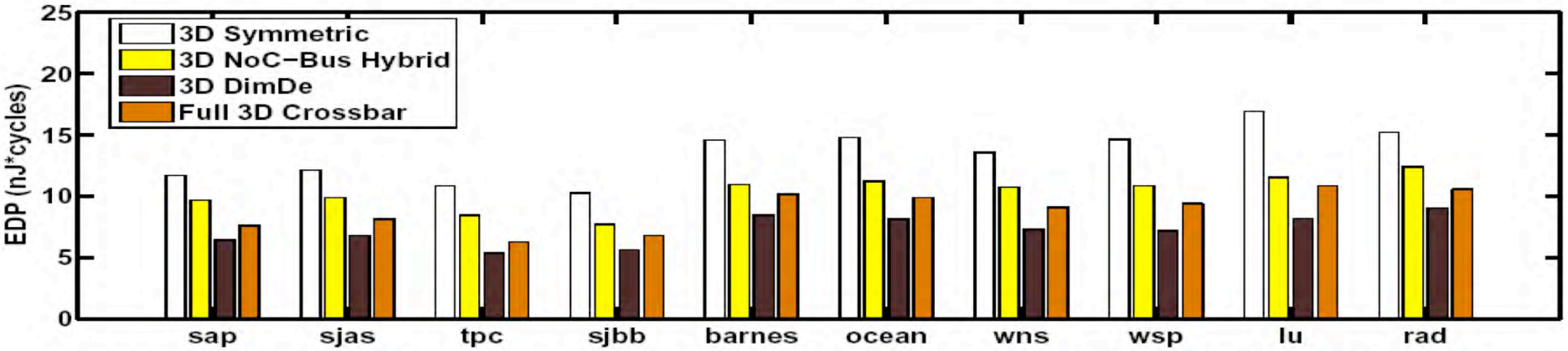
Partially-connected 3D crossbar structure (two pillars)

*** Top View**

# Performance Evaluation with Workload Traces

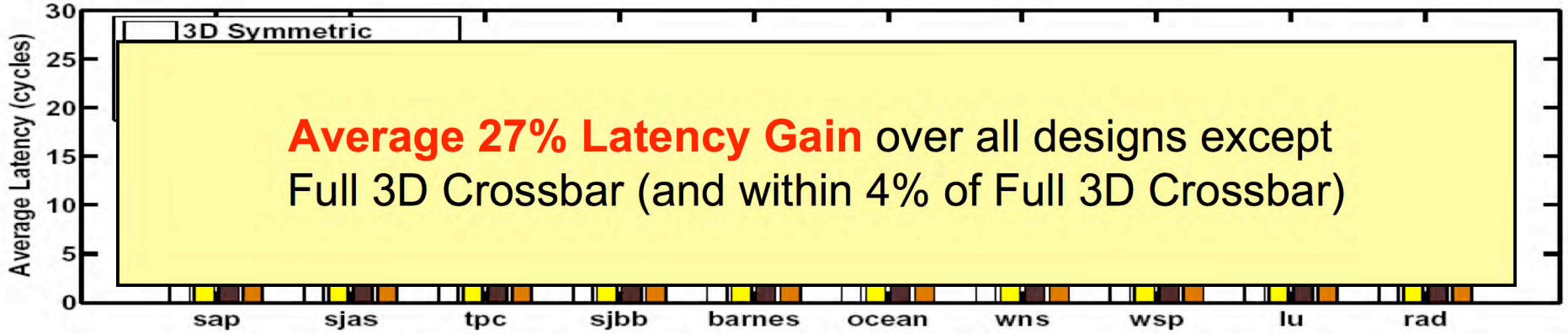



Average Latency with various Commercial and Scientific Workloads

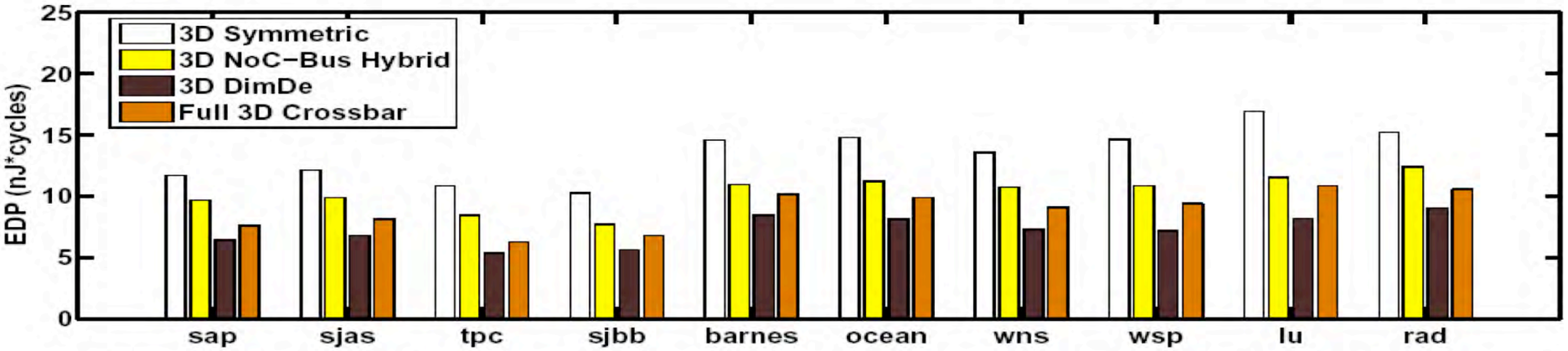



Energy-Delay Product (EDP) with various Commercial and Scientific Workloads

# Performance Evaluation with Workload Traces

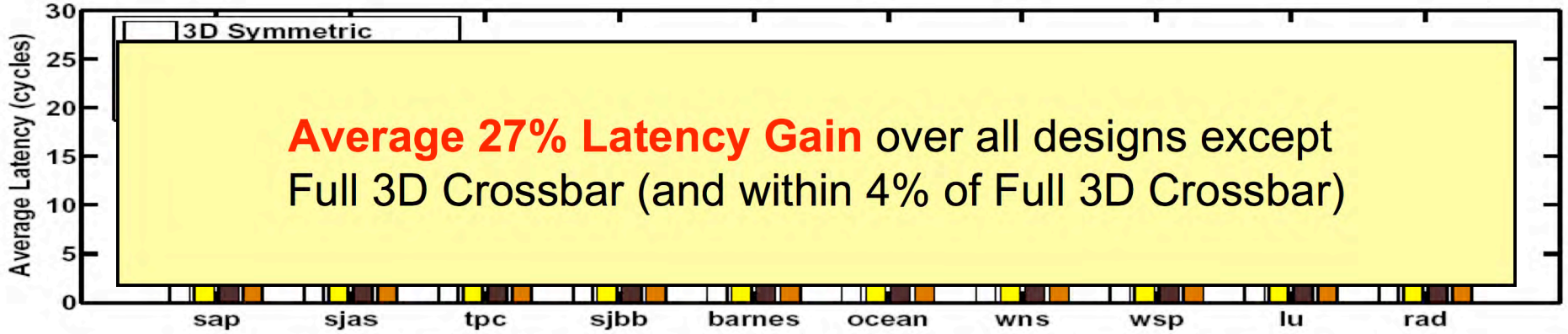



Average Latency with various Commercial and Scientific Workloads

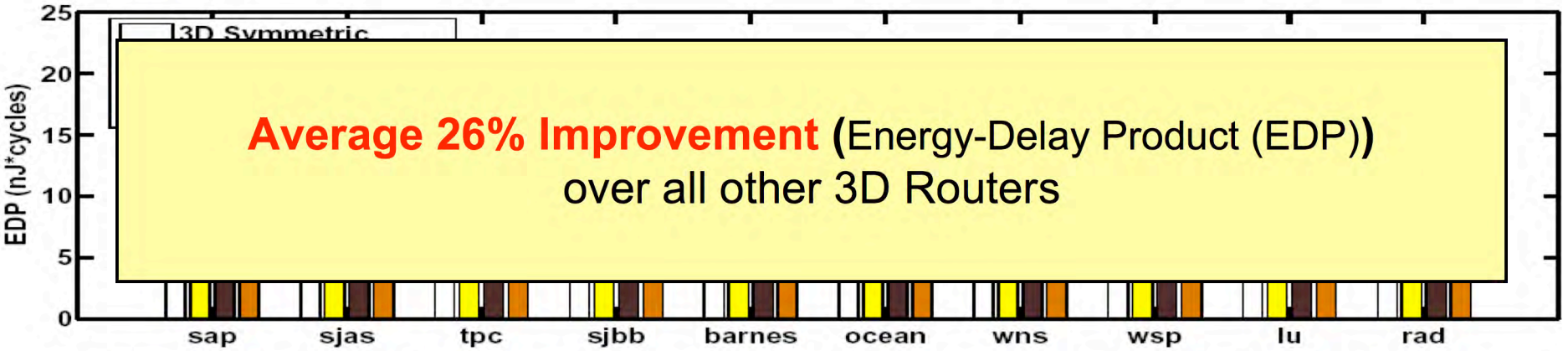



Energy-Delay Product (EDP) with various Commercial and Scientific Workloads

# Performance Evaluation with Workload Traces

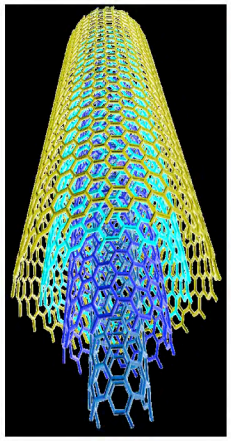



Average Latency with various Commercial and Scientific Workloads

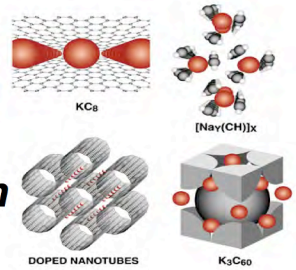


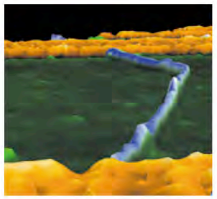

Energy-Delay Product (EDP) with various Commercial and Scientific Workloads

# Looking into Future: CNT- based Interconnects ?

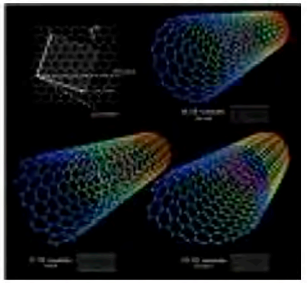


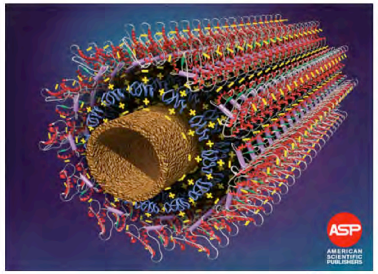
***IBM has announced that its researchers have built the first complete electronic integrated circuit around a single "carbon nanotube" molecule, a new material that shows promise for providing enhanced performance over today's standard silicon semiconductors.***



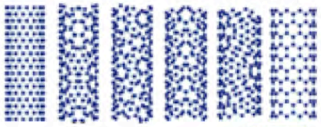



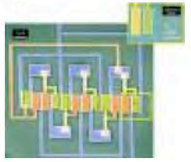
***"Carbon nanotube transistors have the potential to outperform state-of-the-art silicon devices," said Dr. T.C. Chen, vice president, Science & Technology, IBM Research.***





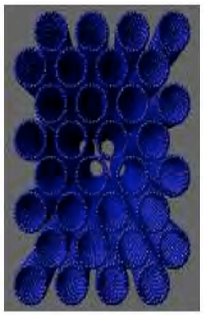

***"Intel is eyeing carbon nanotubes as a possible replacement for copper wires inside semiconductors, a switch that one day could eliminate some big problems for chipmakers." CNET News***





# CNT - An Interconnect Medium?

- High current carrying capacity: Reported Current densities of $10^5$ A/cm$^2$, 1000 times more than copper
- Higher reliability against electromagnetic failures
- Higher signal integrity, less cross-talk issues
- Higher thermal conductivity can help mitigate thermal issues
- High contact resistance – can be solved by using parallel bundles of SWCNTs

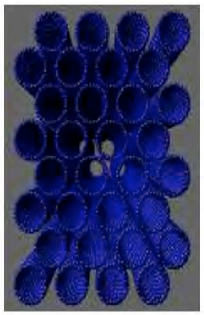


# CNT - An Interconnect Medium?

- High current carrying capacity: Reported Current
  er
  thermal issues

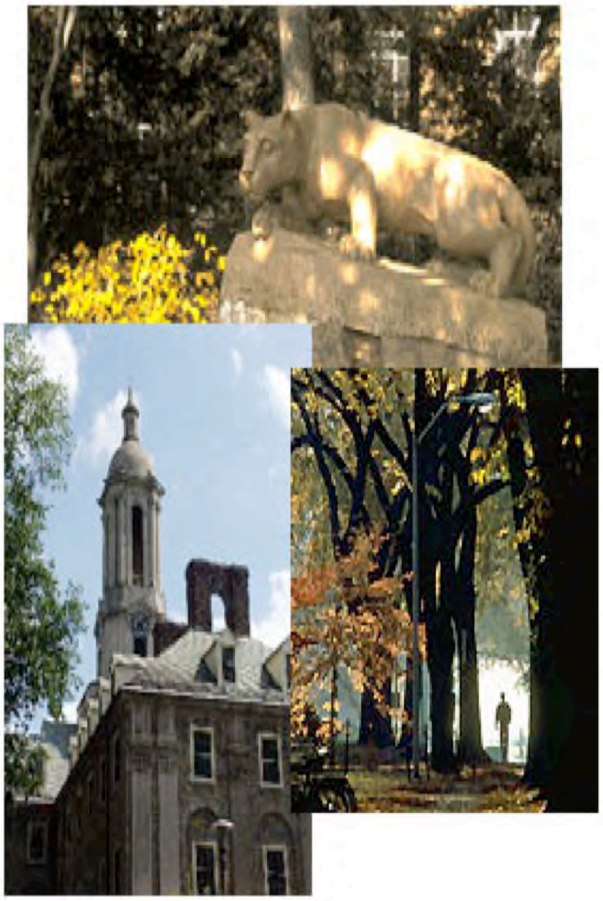
How will these affect NoC Design Decisions?
Will Hybrid Cu-CNT interconnects be attractive?
Can CNT-3D technologies combine to offer new benefits?

- High contact resistance – can be solved by using parallel bundles of SWCNTs



# Summarizing Our NoC Research

# Research Group



Faculty:
Chita R. Das
Vijaykrishnan Narayanan
Yuan Xie

Graduate Students:
Jongman Kim
Dongkook Park
Chrysostomos Nicopoulos
Reetuparna Das

Collaborators:
Intel, IBM, Xilinx, AMD
Univ. of Bologna
Rice University

# Conclusions

- On-chip interconnects will play a significant role in designing next generation multicore architectures
- A holistic approach considering performance, energy, reliability and thermal issues is essential in designing NoC architectures
- Development of accurate performance, energy, reliability, thermal models/tools for NoC is necessary
- The proposed RoCo and 3D routers seems quite promising
- 3D and CNT-based designs are under progress

# Thank You!