HiPINEB 2016 Keynote

Some Architectural Solutions for Exascale Interconnects

José Duato
Universitat Politècnica de València
SPAIN
jduato@disca.upv.es
http://www.gap.upv.es

Outline

- Introduction
- Exascale Interconnection Networks
- Topologies: Scalability, Routing and Fault-Tolerance
- Power Efficiency
- Congestion Management
- Final Observations

Outline

- **Introduction**
- Exascale Interconnection Networks
- Topologies: Scalability, Routing and Fault-Tolerance
- Power Efficiency
- Congestion Management
- Final Observations

Introduction

Historical evolution

- Over the years, there have been very significant architectural innovations that translated into huge performance, cost, and/or scalability benefits
  - Link pipelining
  - Non-blocking three-stage Clos topology
  - Blocking and non-blocking multistage interconnection networks (MINs)
  - Fat-trees (BMINs) and their hierarchical layout
  - Packet switching, cut-through switching (virtual cut-through,wormhole)
  - OS bypassing (zero-copy protocols), RDMA

Introduction

Historical evolution

- Many other architectural innovations were contributed, enabling enhancements in some key areas
  - Reliability
    - Encoding schemes
    - CRCs
    - Retransmission protocols
    - Hardware redundancy and fault bypassing
    - Fault-tolerant routing
    - Automatic path migration
    - Network reconfiguration

Introduction

Historical evolution

- Many other architectural innovations were contributed, enabling enhancements in some key areas
  - Performance and scalability
    - Signal modulation
    - Topologies that match application traffic patterns (3D torus, tree)
    - Topologies that exploit locality, hierarchical topologies
    - Virtual channels
    - Adaptive routing, load balancing
    - Hardware supported broadcast, multicast, barrier sync, reduction
    - Protocol off-loading

Introduction

Historical evolution

- Many other architectural innovations were contributed, enabling enhancements in some key areas
  - Switch architecture
    - Several buffering schemes (input, output, crosspoint, combined input-output)
    - Partitioned / sliced crossbars
    - Distributed arbiters (hierarchical, two- and three-phase schemes)
    - Techniques to reduce HOL blocking (VOQs, VL mapping schemes)
    - Network management support

Introduction

Historical evolution

- Many other architectural innovations were contributed, enabling enhancements in some key areas
  - QoS, traffic isolation
    - Virtual channels / lanes
    - Arbitration and preemption algorithms
    - Service levels
    - Network partitioning (virtual networks)
    - VL mapping schemes

Introduction

Historical evolution

- There are even innovations included in commercial products that (almost) nobody uses
  - Reactive congestion management
    - Detection at switches and notification to the sources contributing to congestion
    - Throttling algorithms that regulate injection based on congestion notifications
    - Closed loop control system with a pure delay in the feedback chain that easily leads to instability and oscillatory behavior
    - Very difficult to tune, even for a small testbed with constant injection rates
    - Tuning difficulty increases with network size (longer delay) and link bandwidth
    - Poor performance, due to periodic link underutilization

Introduction

Historical evolution

- As it happened with most technologies, most of the innovations with a large impact were introduced quite early
- Most of the performance improvement over the last decade came from higher link bandwidth (technology improvement) and faster related components (faster PCIe, faster processor)
- A significant part of the academic effort is devoted to hardware and software optimizations that deliver marginal benefits:
  - Optimized communication libraries and fabric management
  - New topologies that represent different trade-offs between locality, average distance, diameter, path diversity, scalability, cost, layout, etc.

Introduction

Historical evolution

Everything seems to indicate that interconnection networks are a mature field, and most of the future improvements will come from technology advances

Introduction

Example of recent commercial interconnect

- Intel Omni-Path Architecture (OPA)
  - Leverages Ethernet and InfiniBand PHYs. Same link bandwidth (25GB/s)
  - Same fat-tree topology and layout, but larger address field (24 bits)
  - Benefits from larger switch port count. Edge switches provide 48 ports
  - Intel claims that main improvements over InfiniBand come from connectionless transmission and optimized communication libraries, but Mellanox already introduced similar enhancements in InfiniBand
  - Other benefits include shorter latency, link-level retransmission, priority interleaving, QoS support, and virtual network partitioning
  - Roughly the same congestion management mechanism as in InfiniBand

Is this the interconnect technology required for Exascale computing?

Outline

- Introduction
- **Exascale Interconnection Networks**
- Topologies: Scalability, Routing and Fault-Tolerance
- Power Efficiency
- Congestion Management
- Final Observations

Exascale Interconnection Networks

What does the Exascale challenge imply?

	2010	2018	Factor Change
System peak	2 Pf/s	1 Ef/s	500
Power	6 MW	20 MW	3
System Memory	0.3 PB	10 PB	33
Node Performance	0.125 Gf/s	10 Tf/s	80
Node Memory BW	25 GB/s	400 GB/s	16
Node Concurrency	12 cpus	1,000 cpus	83
Interconnect BW	1.5 GB/s	50 GB/s	33
System Size (nodes)	20 K nodes	1 M nodes	50
Total Concurrency	225 K	1 B	4,444
Storage	15 PB	300 PB	20
Input/Output bandwidth	0.2 TB/s	20 TB/s	100

The Opportunities and Challenges of Exascale Computing. Summary Report of the Advanced Scientific Computing Advisory Committee (ASCAC) Subcommittee. U.S. Department of Energy, Fall 2010

Exascale Interconnection Networks

Current situation

- **Breakdown of Moore's Law** and **Dennard Scaling**: Transistors may become smaller but power density is no longer constant but increases, so no way for "ever faster chips"
- Current multicore processors on the way to achieve **more computing power** and **less power consumption**
  - Current ARM products offer a good performance/watt ratio
  - Expected Intel, AMD or NVIDIA power-efficient solutions
- **Accelerators** can help to increase performance in heterogeneous systems while keeping power consumption
  - GPGPUs deliver much better performance/watt than CPUs
  - rCUDA is able to double throughput but increases network traffic

Exascale Interconnection Networks

June 2013 Green500 list

Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
1	3,208.83	CINECA	Eurora - Eurotech Aurora HPC 10-20, Xeon E5-2687W 8C 3.100GHz, Infiniband QDR, NVIDIA K20	30.70
2	3,179.88	Selex ES Chieti	Aurora Tigon - Eurotech Aurora HPC 10-20, Xeon E5-2687W 8C 3.100GHz, Infiniband QDR, NVIDIA K20	31.02
3	2,449.57	National In… Computational Sciences/Univer… …ssee	Beacon - Appro GreenBlade GB824M, Xeon E5-2670 8C 2.600GHz, Infiniband FDR, Intel Xeon Phi 5110P	45.11
4	2,351.10	King Abdulaziz Cit…	SANAM - Adtech, ASUS ESC4000/FDR G2, Xeon E5-… …D FirePro	179.15
5	2,299.15	IBM Thomas J. W…	…Custom	82.19
6	2,299.15	DOE/SC/Argonne…	…600GHz,	82.19
7	2,299.15	Ecole Polytechnique Federale de Lausanne	…BQC 16C 1.600GHz, Custom Interconnect	82.19
8	2,299.15	Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw	BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect	82.19
9	2,299.15	DOE/SC/Argonne National Laboratory	Vesta - BlueGene/Q, Power BQC 16C 1.60GHz, Custom	82.19
10	2,299.15	University of Rochester	BlueGene/Q, Power BQC 16C 1.60GHz, Custom	82.19

1 ExaFLOP = 311 MW

* Performance data obtained from publicly available sources including TOP500

Exascale Interconnection Networks

Current Green500 list

Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
1	7,031.58	Institute of Physical and Chemical Research (RIKEN)	Shoubu - ExaScaler-1.4 80Brick, Xeon E5-2618Lv3 8C 2.3GHz, Infiniband FDR, PEZY-SC	50.32
2	5,331.79	GSIC Center, Tokyo Institute of [illegible]ology	TSUBAME-KFC/DL - LX 1U-4GPU/104Re-1G Cluster, Intel Xeon E5-2620v2 6C 2.1GHz, Infiniband FDR, NVIDIA Tesla K80	51.13
3	5,271.81	GSI Helmho[illegible]	ASUS ESC4000 FDR/G2S, Intel Xeon E5-2690v2 10C 3GHz, Infiniband FDR, AMD FirePro S9150	57.15
4	4,778.46	Institute of Modern Physi[illegible] Chinese Acad[illegible]	[illegible]on Cluster W780I, Xeon E5-2640v3 8C 2.6GHz, Infiniband	65.00
5	4,112.11	Stanford Res[illegible] Center	[illegible]C 2.8GHz,	190.00
6	3,856.90	IT Company	[illegible].4GHz, 10G	58.00
7	3,775.45	Internet Service	Inspur TS10000 HPC Server, Intel Xeon E5-2620v2 6C 2.1GHz, 10G Ethernet, NVIDIA Tesla K40	110.00
8	3,775.45	Internet Service	Inspur TS10000 HPC Server, Intel Xeon E5-2620v2 6C 2.1GHz, 10G Ethernet, NVIDIA Tesla K40	110.00
9	3,775.45	Internet Service	Inspur TS10000 HPC Server, Intel Xeon E5-2620v2 6C 2.1GHz, 10G Ethernet, NVIDIA Tesla K40	110.00
10	3,775.45	Internet Service	Inspur TS10000 HPC Server, Intel Xeon E5-2620v2 6C 2.1GHz, 10G Ethernet, NVIDIA Tesla K40	110.00

* Performance data obtained from publicly available sources including TOP500

1 ExaFLOP = 142 MW

Exascale Interconnection Networks

How to achieve Exascale goals?

- It is still clearly necessary to drastically increase the performance/watt ratio to achieve **Exascale goals**, but **HOW?**
- Most likely approach:
  - **Exascale processors are likely to reduce their peak computing power to lower power consumption**
  - **Accelerators will continue to be used to increase node peak computing power while keeping power consumption down**

Exascale Interconnection Networks

Massive paralelism in Exascale systems

	2010	2018	Factor Change
System peak	2 Pf/s	1 Ef/s	500
Power	6 MW	20 MW	3
System Memory	0.3 PB	10 PB	33
Node Performance	0.125 Gf/s	10 Tf/s	80
Node Memory BW	25 GB/s	400 GB/s	16
Node Concurrency	12 cpus	1,000 cpus	83
Interconnect BW	1.5 GB/s	50 GB/s	33
System Size (nodes)	20 K nodes	1 M nodes	50
Total Concurrency	225 K	1 B	4,444
Storage	15 PB	300 PB	20
Input/Output bandwidth	0.2 TB/s	20 TB/s	100

The Opportunities and Challenges of Exascale Computing. Summary Report of the Advanced Scientific Computing Advisory Committee (ASCAC) Subcommittee. U.S. Department of Energy, Fall 2010

Exascale Interconnection Networks

How to achieve Exascale goals?

- It is still clearly necessary to increase drastically the performance/watt ratio to achieve **Exascale goals**, but **HOW?**
- Most likely approach:
  - **Exascale processors are likely to reduce their peak computing power to lower power consumption**
  - **Accelerators will continue to be used to increase node peak computing power while keeping power consumption down**
- Thus, **interconnection networks able to connect a huge number of nodes are likely to be required in future Exascale systems**
- However, designing such interconnection networks is not obvious

Exascale Interconnection Networks

Interconnection Networks in the Exascale challenge

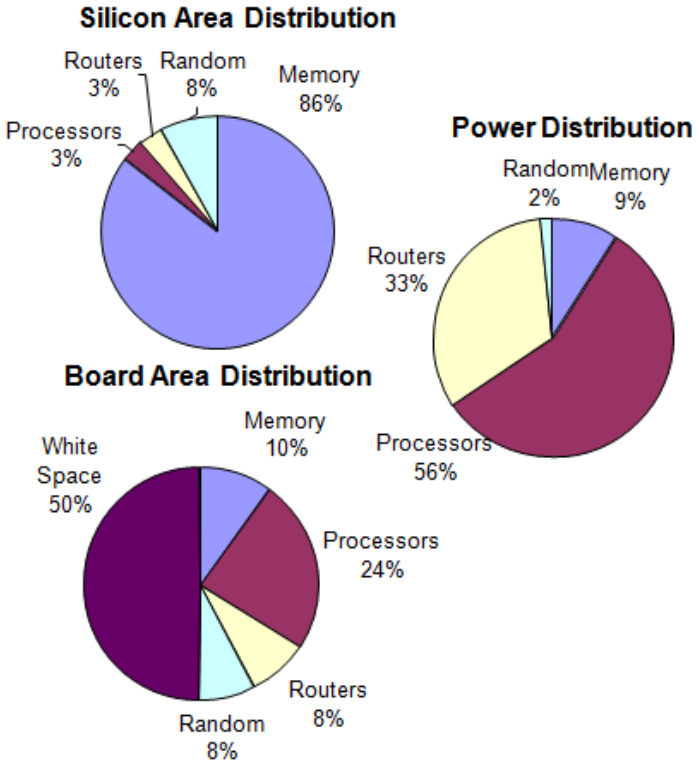
	2010	2018	Factor Change
System peak	2 Pf/s	1 Ef/s	500
Power	6 MW	20 MW	3
System Memory	0.3 PB	10 PB	33
Node Performance	0.125 Gf/s	10 Tf/s	80
Node Memory BW	25 GB/s	400 GB/s	16
Node Concurrency	12 cpus	1,000 cpus	83
Interconnect BW	1.5 GB/s	50 GB/s	33
System Size (nodes)	20 K nodes	1 M nodes	50
Total Concurrency	225 K	1 B	4,444
Storage	15 PB	300 PB	20
Input/Output bandwidth	0.2 TB/s	20 TB/s	100

The Opportunities and Challenges of Exascale Computing. Summary Report of the Advanced Scientific Computing Advisory Committee (ASCAC) Subcommittee. U.S. Department of Energy, Fall 2010

Exascale Interconnection Networks

Power Consumption in Interconnection Networks

- **Power consumption fraction** of the interconnection network is **one third** of total power
- Most of the network power consumption is **devoted to links**
- Link power consumption is very difficult to reduce because **signal attenuation** will be worse at higher clock frequencies.
- **Optical links** are required beyond a certain distance



The Opportunities and Challenges of Exascale Computing. Summary Report of the Advanced Scientific Computing Advisory Committee (ASCAC) Subcommittee. U.S. Department of Energy, Fall 2010

Exascale Interconnection Networks

Challenges in Exascale Interconnection Networks

- We are within a factor of two of the expected link bandwidth
- Scalability: The network has to scale to one million nodes
- Reliability: Probability of failure will increase with # of nodes
- Fault tolerance: Current techniques seem to work well against transmission errors and component failures. Will they scale?
- Cost: The network should not be overdimensioned. Same components should be valid for medium to huge systems
- Power consumption: Should be reduced by a factor of 7 to 10
- Congestion Management : Will become mandatory

Exascale Interconnection Networks

Challenges in Exascale Interconnection Networks

- Scalability: The network has to scale to one million nodes
- Reliability: Probability of failure will increase with # of nodes
- Fault tolerance: Current techniques seem to work well against transmission errors and component failures. Will they scale?
- Cost: The network should not be overdimensioned. Same components should be valid for medium to huge systems
- Power consumption: Should be reduced by a factor of 7 to 10
- Congestion Management : Will become mandatory

They must not be considered separately, since they are closely related to each other

Exascale Interconnection Networks

Challenges in Exascale Interconnection Networks

- Scalability: The network has to scale to one million nodes
- Reliability: Probability of failure will increase with # of nodes
- Fault tolerance: Current techniques seem to work well against transmission errors and component failures. Will they scale?
- Cost: The network should not be overdimensioned. Same components should be valid for medium to huge systems
- Power consumption: Should be reduced by a factor of 7 to 10
- Congestion Management : Will become mandatory

Scalability (up and down), fault tolerance, cost: Can be addressed mostly by defining a suitable topology

Outline

- Introduction
- Exascale Interconnection Networks
- **Topologies: Scalability, Routing and Fault-Tolerance**
- Power Efficiency
- Congestion Management
- Final Observations

Topologies

Scaling to 1M endnodes

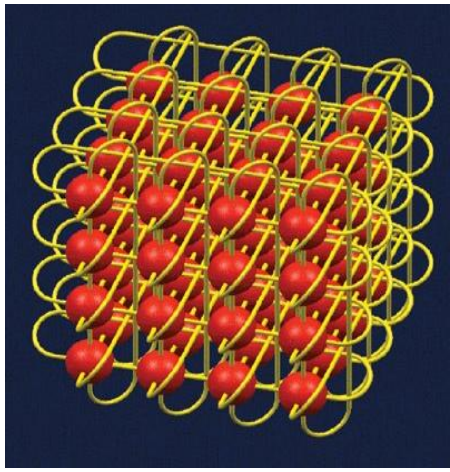
- Main objectives:
  - **High connectivity**
  - **Low latency and high throughput**
  - Reducing **cost and power consumption**
- Design trends:
  - Reducing **network diameter** (reaching more nodes in fewer hops)
  - Optimizing the **number of components** (no overdimension)
  - Cost-efficient **routing algorithms**
  - Providing **alternative paths** to select in case of failure

Topologies
Suitable Approaches

- **Specialized** topologies that match the applications communication patterns while being relatively inexpensive
  - **3D torus** (and a binary tree) like in the Cray 3TD and IBM BlueGene
- **Cost-effective general-purpose** topologies
  - **Fat tree** and its different optimizations (Orthogonal FT, Slimmed FT)
- **Hierarchical** topologies that adapt to the two cabling technologies (copper and optical fiber)
  - **Dragonfly** and others. Can combine topologies with spare links
- **Hybrid** topologies that exploit on-node & off-node switches
  - **KNS** and others. Also valid for on-NIC switches and off-node switches

Topologies

Specialized topologies



- Many of the HPC applications requiring Exascale performance simulate 3D models
- These simulations require neighbor-to-neighbor communication in all directions
  - A 3D torus matches this pattern
- An additional tree network may be required for data and code distribution, barrier sync, and collective communication
  - Separate network (IBM BlueGene) o embedded (Cray T3E)
- Not valid for general purpose applications
  - 1 Mnode network (128x128x64): 32-to-1 oversubscription, diameter=160

Topologies

General-purpose topologies

- Most manufacturers cannot afford specialized designs for huge supercomputers that are not suitable for smaller sizes
- When application behavior is not known in advance or a large varied mix will be run, link oversubscription should be limited
- The **fat tree** has become very popular because it provides full bisection bandwidth (no oversubscription) and it has a very convenient layout up to a certain size
  - First stage within the rack (edge switches) with copper links, second and third stages within director switches
  - Optical fiber links to connect edge switches with director switches
  - Orthogonal board arrangement within director switches

Topologies

Fat tree and derivatives

- Scaling to 1 Mnodes is not trivial. Assuming 64-port switches by the time 1 ExaFLOPS is reached, a fat tree would require:
  - Four stages, 114,688 switches, 1,048,576 NICs, and 4,194,304 cables
  - A new stage is required between edge and director switches
- Researchers in industry and academia are trying to optimize fat trees to reduce their cost and increase scalability
  - Bisection bandwidth across upper stages and/or path diversity can be traded for reduced cost and increased node count for a given diameter
  - This will produce oversubscription (in general or for specific patterns)
  - Application locality should be exploited to prevent network congestion

Topologies

Optimization bounds

- Useful for knowing the available margin for optimization
- Keeping full bisection bandwidth and path diversity = 1
  - Since the orthogonal fat tree was only defined for latency = 3, we can use unidirectional MINs (UMINs) as a reference
  - A UMIN provides a single path between each source-destination pair, achieves minimum diameter but all the nodes are at a distance equal to the diameter. It requires n stages, $n = \log_k N$
  - A unidirectional Benes network provides the same path diversity as a fat tree, and does so at the expense of increasing from n to 2n-1 stages
  - So, at best, reducing path diversity to 1 could reduce fabric latency from 2n-1 to n hops, and total switch cost by the same factor (less than 2X)

Topologies
Optimization bounds

- Useful for knowing the available margin for optimization
- Reducing bisection bandwidth at all stages to the minimum
  - Moving from a fat tree to a tree. Only one link from each switch goes up
  - Huge increase in the number of nodes for the same number of stages (for 4 stages and 64-port switches, from 1 to 16 million nodes)
  - Large reduction in total switch cost (by a factor of more than 6X)
  - But still four stages required to reach 1 Mnode
  - And huge oversubscription (63-to-1), only twice as bad as for a 3D torus

Topologies

Fat tree derivatives

- Orthogonal fat tree
  - Defined only for latency equal to three. Cannot connect 1 Mnodes when using 64-port switches
  - Only within a factor of 2X from the cost and latency of a full fat tree
  - For certain traffic patterns, it may suffer from huge oversubscription
    - When, for every switch, all the nodes attached to that switch simultaneously send traffic to the respective nodes attached to a different switch
    - Since path diversity between first stage switches has been reduced to one, oversubscription would be 32-to-1 for 64-port switches (same as for 3D torus)
    - Oversubscription dramatically changes from one communication pattern to another one, forcing programmers to be aware of the topology constraints

Topologies
Fat tree derivatives

- Slimmed fat trees
  - Wide margin to define cost & scalability vs. oversubscription trade-off
  - The lower to upper port ratio (oversubscription rate) can be independently set for each stage
    - First stage to match # of nodes in a rack, upper stages depending on application locality, director switch organization, and/or cable cost and layout
  - Oversubscription independent of traffic pattern (to other switches)
    - 31,936 switches (vs. 114,688 in the full fat tree) for a 3-to-1 oversubscription
    - 52,480 switches for 1,024,000 nodes with a 5-to-3 oversubscription
    - Cost (and power) reduction is better than oversubscription increase
    - Maximum latency remains the same

Routing

Efficient Deterministic Routing Algorithms for Indirect Networks

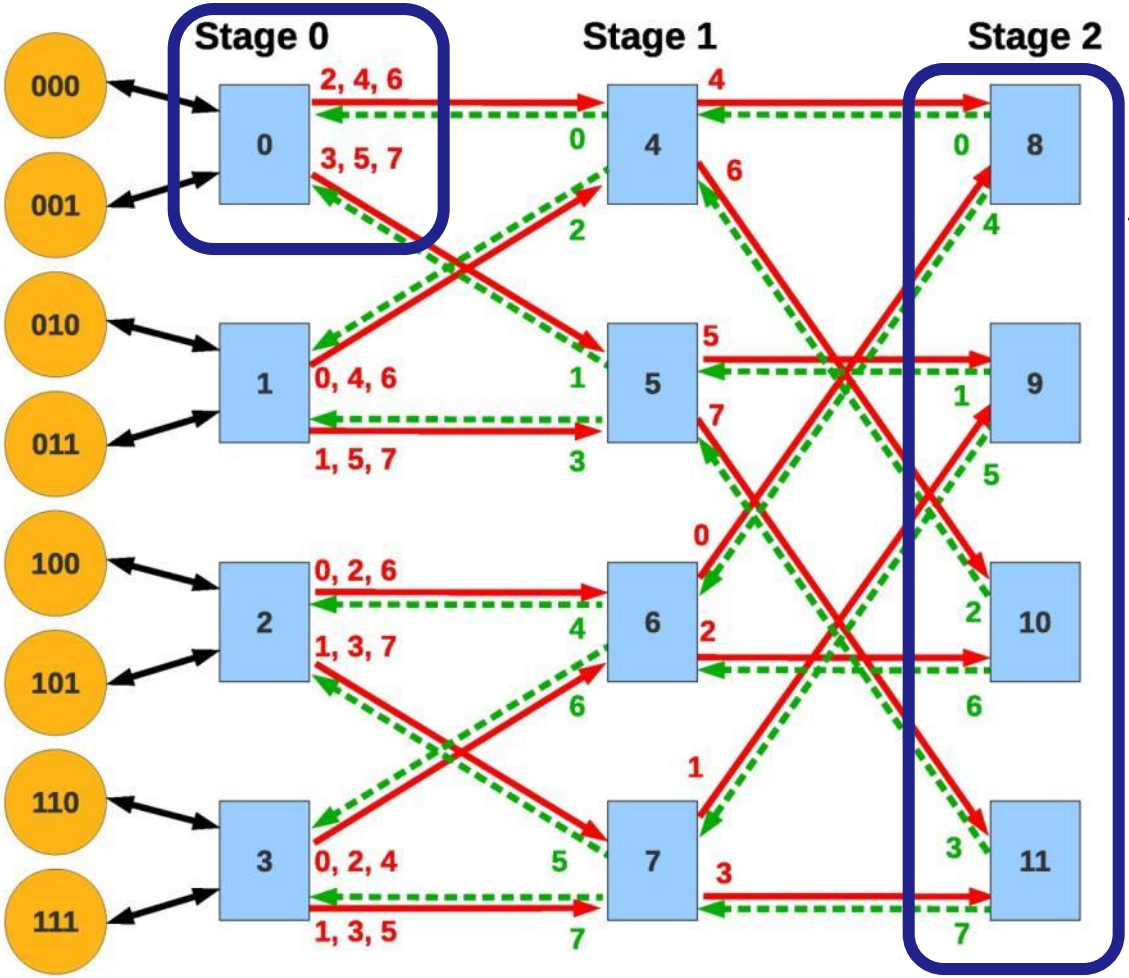
- **Tailored to specific network topologies**
- **Balance the destinations** among the different paths
- Achieve **similar performance** to **adaptive routing** while requiring **fewer resources** to be implemented
- They **don't** produce packet **out-of-order delivery** problems
- Can **be recomputed if some faults appear** in the network
- In case of congestion, the **size of the congestion tree** is minimal

[1] Crispín Gómez Requena, Francisco Gilabert Villamón, María Engracia Gómez, Pedro López, José Duato: ***Deterministic versus Adaptive Routing in Fat-Trees****. IPDPS 2007: 1-8*

[2] Eitan Zahavi, Greg Johnson, Darren J. Kerbyson, Michael Lang: ***Optimized InfiniBandTM fat-tree routing for shift all-to-all communication patterns.*** *Concurrency and Computation: Practice and Experience 22(2): 217-231 (2010)*

Routing

Example of Efficient Routing: DESTRO in a k-ary n-tree

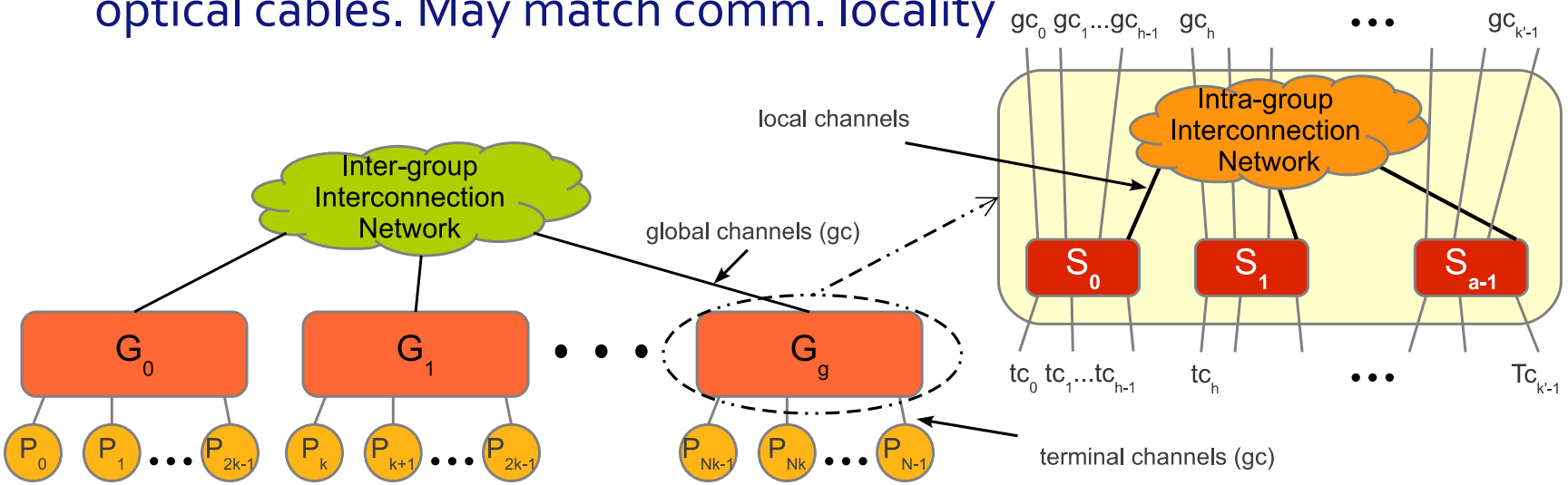


Balances the use of links by different paths

Topologies

Hierarchical Networks

- Most prominent example are **Dragonflies**
- **Hierarchical network** (3-levels): switch, group, and system
- Global links are significantly long (optical fiber is mandatory)
- Network diameter reduction but routing becomes quite complex
- The main benefit is that they match the split between copper and optical cables. May match comm. locality

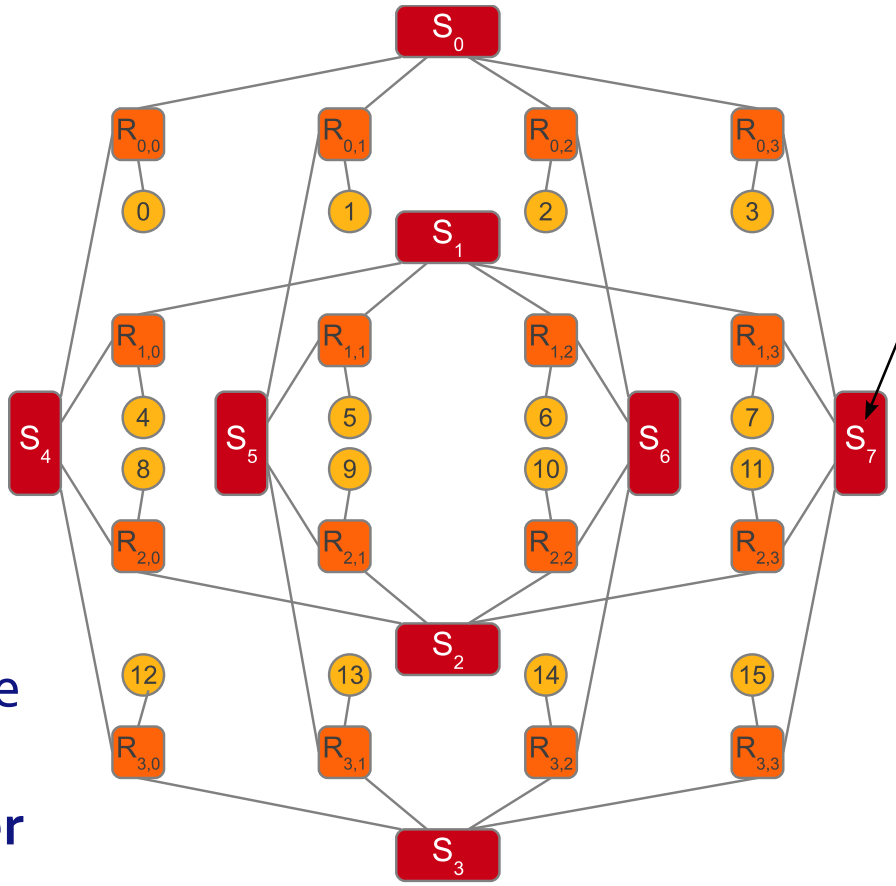


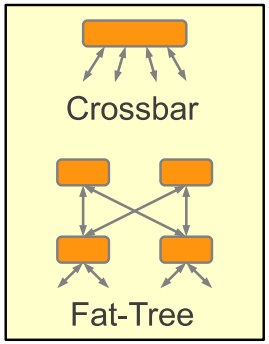
John Kim, William J. Dally, Steve Scott, Dennis Abts: ***Technology-Driven, Highly-Scalable Dragonfly Topology****. ISCA 2008: 77-88*

Topologies

Hybrid Networks (KNS)

- Designed for **huge network sizes**
- Based on merging **direct** and **indirect topologies**
- Reduce the **diameter**, **number of switches and links**
- **Some path diversity**, which allows a good level of fault-tolerance
- **Low latency, high-throughput** and **lower cost** than indirect networks



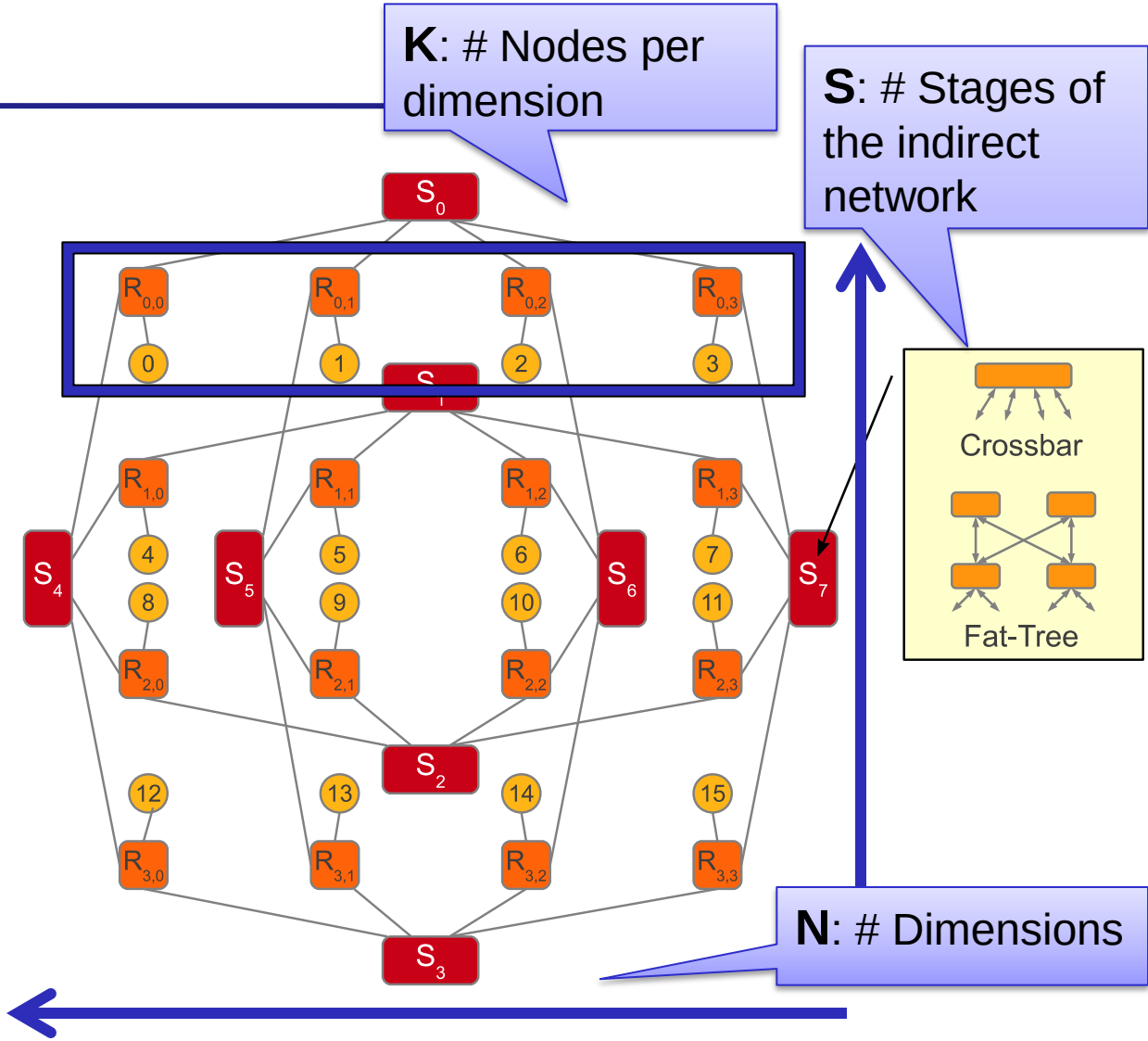


Roberto Peñaranda, Crispín Gómez Requena, María Engracia Gómez, Pedro López, José Duato: ***A New Family of Hybrid Topologies for Large-Scale Interconnection Networks****. NCA 2012: 220-227*

Topologies

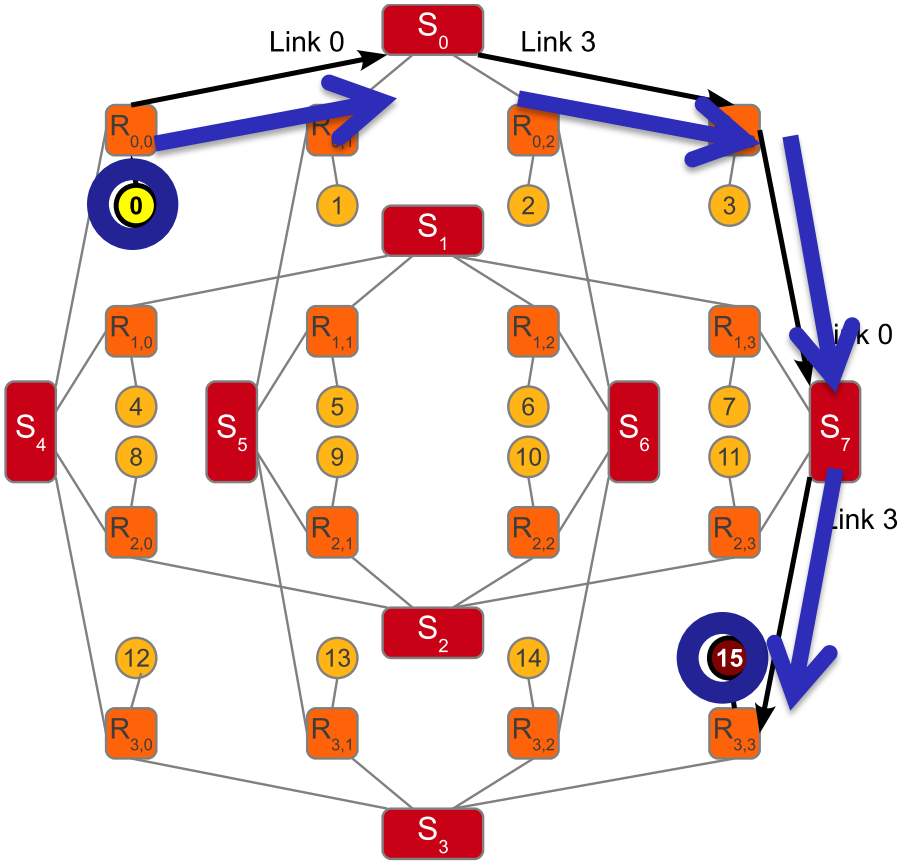
KNS hybrid topology

- Nodes are organized **orthogonally**, in several dimensions, like in direct networks:
  - **Routers**
- Dimensions are connected by means of **indirect networks**:
  - **Crossbar, Fat-tree, …**
- Defined by using **three parameters**: K, N, and S



Routing

Example of Hybrid-DOR in a KNS hybrid topology



Roberto Peñaranda, Crispín Gómez Requena, María Engracia Gómez, Pedro López, José Duato: ***A New Family of Hybrid Topologies for Large-Scale Interconnection Networks****. NCA 2012: 220-227*

Topologies

KNS hybrid topology

- For huge sizes, KNS is **superior** to existing topologies:
  - It provides **switching capabilities at both switches and network interfaces**, and not only at switches (like indirect networks) or at network interfaces (like direct networks).
  - It **provides several disjoint alternative paths**, all of them having the same length, unlike other topologies with high connectivity (e.g. the flattened butterfly provides many alternative paths longer than the minimal one).
  - It **directly benefits from the best routing techniques** for orthogonal direct networks and for fat trees, **requiring neither hierarchical nor non-minimal routing algorithms** for achieving enough path diversity.

Topologies
KNS hybrid topology

- KNS is not easy to incorporate in commercial designs
  - It would only require small changes in the NIC ASIC design if the NIC already implements two or more ports (for redundancy or higher BW). Most commercial NICs (Omni-Path, InfiniBand) implement just one port
  - For two-port NICs, the benefits are small for a 1 Mnode 2D KNS:
    - It would require 98,304 switches (64-port) for full bisection bandwidth (14% savings with respect to a fat tree in which only one port per NIC is used)
    - However, it would provide two fully disjoint paths between any pair of nodes (redundancy equivalent to two fat trees in parallel, one per NIC port)
  - Benefits become much larger when only one switch (and not a small fat tree) is required in each dimension

Topologies
KNS hybrid topology

- KNS is ahead of current and near future needs
  - Benefits become much larger when only one switch (and not a small fat tree) is required in each dimension
    - For 1 Mnode and 64-port switches, it would require NICs with four ports
    - But switch count would drop to 65,536 (43% savings w.r.t. a fat tree)
  - It is best suited for future multicore chips with an on-chip router
    - Intel already announced plans for including an Omni-Path port in its chips
    - Future processor chips will likely incorporate an on-chip router with a few ports
    - A 3D KNS with 128-port external switches would interconnect 2 Mnodes, using only 49,152 switches (57% savings w.r.t. a fat tree), with lower latency, and providing three disjoint paths (w.r.t. one for the fat tree)

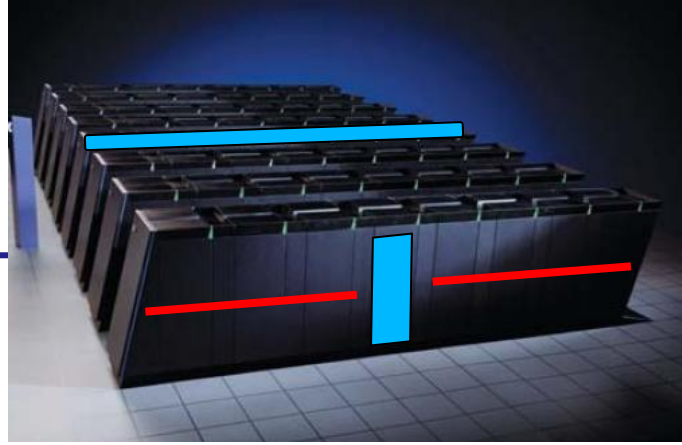
Topologies and Scalability

Fault Tolerance

- Most transmission errors are due to transient faults, and current recovery mechanisms seem to work well enough
- Probability of failure will increase with # of components
- In case of NIC failure, the corresponding node gets isolated
- Fat trees provide alternative paths but a **failure in a first stage switch** will disconnect all the attached nodes
- **Two-port NICs** are required for true network fault tolerance, but this requires doubling the number of ports in the fat tree
- Hybrid topologies offer several **alternative paths** natively

Topologies and Scalability

Some additional issues



- Optimized layout
  - Middle-of-rack switches with copper cables, switch rack(s) in the middle of each row, row(s) of director switches in the middle of the room
- Flow control
  - A significant fraction of the cables must be very long (possibly exceeding 50 m in Exascale computers)
  - At 25 Gbps, signal propagates less than 1 cm per injected bit. For 50 m 4X links, buffer size must be at least **5 Kbytes** for running at full speed
  - When implementing multiple virtual lanes, buffer space should be dynamically assigned (i.e., a large shared flow-controlled **landing pad** plus small **dedicated buffers** for each virtual lane)

Outline

- Introduction
- Exascale Interconnection Networks
- Topologies: Scalability, Routing and Fault-Tolerance
- **Power Efficiency**
- Congestion Management
- Final Observations

Power Efficiency

Motivation

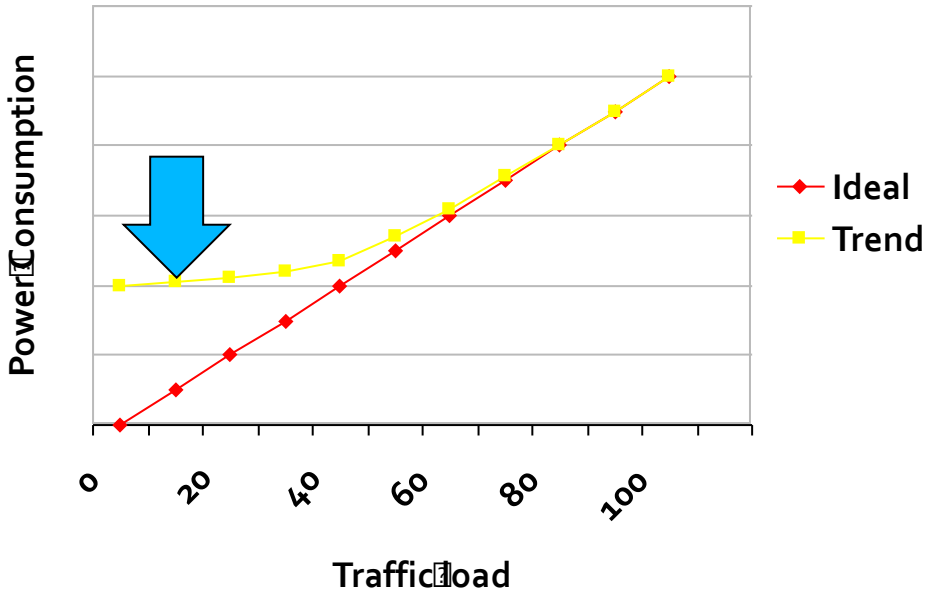
- High **cost of the power consumption bill** for large HPC systems: power and cooling
- The **interconnection network power consumption fraction** is about 20% of the total idle power, increasing an additional 20% when simple benchmarks are used [1]
- Significant **reductions in power consumption for CPUs**, thus increasing the percentage consumed by interconnects
- Power efficiency in HPC interconnects is thus a **challenge**:
  - **Idle networks** have a high power consumption
  - **Hw/Sw infrastructure** must provide power saving mechanisms

[1] Torsten Hoefler: ***Software and Hardware Techniques for Power-Efficient HPC Networking****. Computing in Science and Engineering 12(6): 30-37 (2010)*

Power Efficiency

Energy consumption

- Most of the interconnects power is consumed by **links**
- **Number and length** of the links is important
- **Contention** increases the power consumption
- Proposed solutions:
  - Hardware
  - Software



Power Efficiency

Software solutions

- **Proactive solutions:**
  - Schedule the traffic so that hot-spots are minimized
  - Maintain the network with low utilization
- **Problems of software solutions:**
  - Technology advances lead to **increasing link speed**
  - Exascale topology sizes make **traffic scheduling very complex**
  - Even at low network utilization, the **idle power consumed by the links is significant**

Power Efficiency

Hardware solutions

- **Dynamic Voltage Scaling (DVS)**
  - Adds **complexity**
  - Introduces **delay overhead**
- **Turn off the links** completely:
  - Requires a **fault-tolerant routing algorithm**
  - **Path diversity** is also required
  - Adds **complexity**
  - Slow reaction to **traffic bursts**

Power Efficiency

Hardware solutions

- If ports are connected to **aggregated serial links** (i.e. 4x, 8x…): **Dynamically turning on and off individual links** of the same port (w/o disabling it completely):
  - Connectivity is not affected
  - The routing algorithm is preserved
- **Common problems of hardware solutions**:
  - **Slow reaction** when traffic bursts appear
  - Traffic bursts may **produce congestion trees**

Marina Alonso, Salvador Coll, Juan-Miguel Martinez, Vicente Santoja, Pedro López and José Duato. ***Power Saving in regular interconnection networks****. Journal on Parallel Computing. December 2010*

Power Efficiency

Integral solutions (yet to be developed)

- Components for global integral power management:
  - **Traffic schedulers** that try to leave some paths idle while loading others
  - **Traffic load estimators** based on local measurements (activity history?)
  - Centralized or distributed **power management algorithm**
  - Mechanisms to **dynamically reduce link power** consumption
- The most expensive components (power reduction mechanisms) already exist in most commercial interconnects
  - Link bandwidth negotiation mechanisms and protocols (used for DFS)
  - Dynamic link narrowing (e.g., Omni-Path's Dynamic Lane Scaling and similar mechanisms in InfiniBand and HyperTransport)

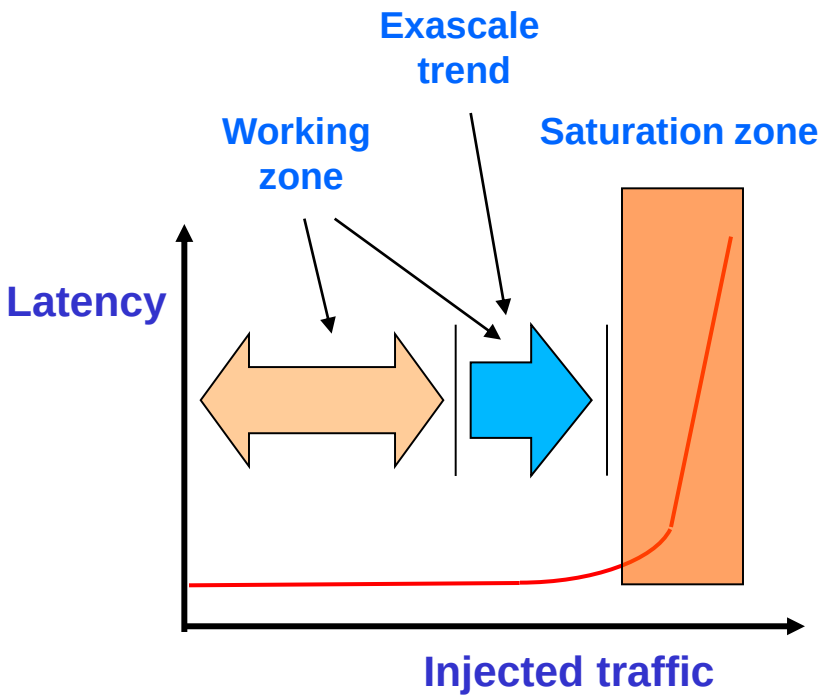
Outline

- Introduction
- Exascale Interconnection Networks
- Topologies: Scalability, Routing and Fault-Tolerance
- Power Efficiency
- **Congestion Management**
- Final Observations

Congestion Management

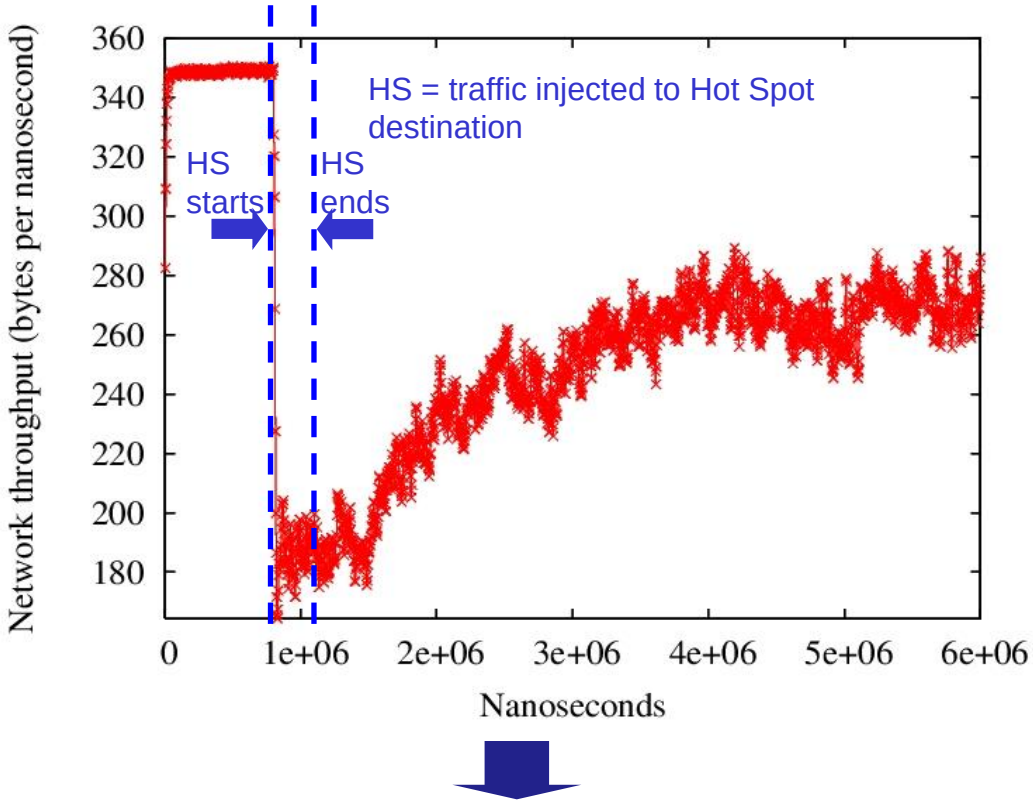
Why is congestion management necessary?

- Exascale networks: around **one million endnodes**
- **Cost and power constraints** lead to use the minimum number of components, thus working close to the **saturation zone and increasing congestion probability**
- **Power management** policies react slowly to traffic bursts



Congestion Management

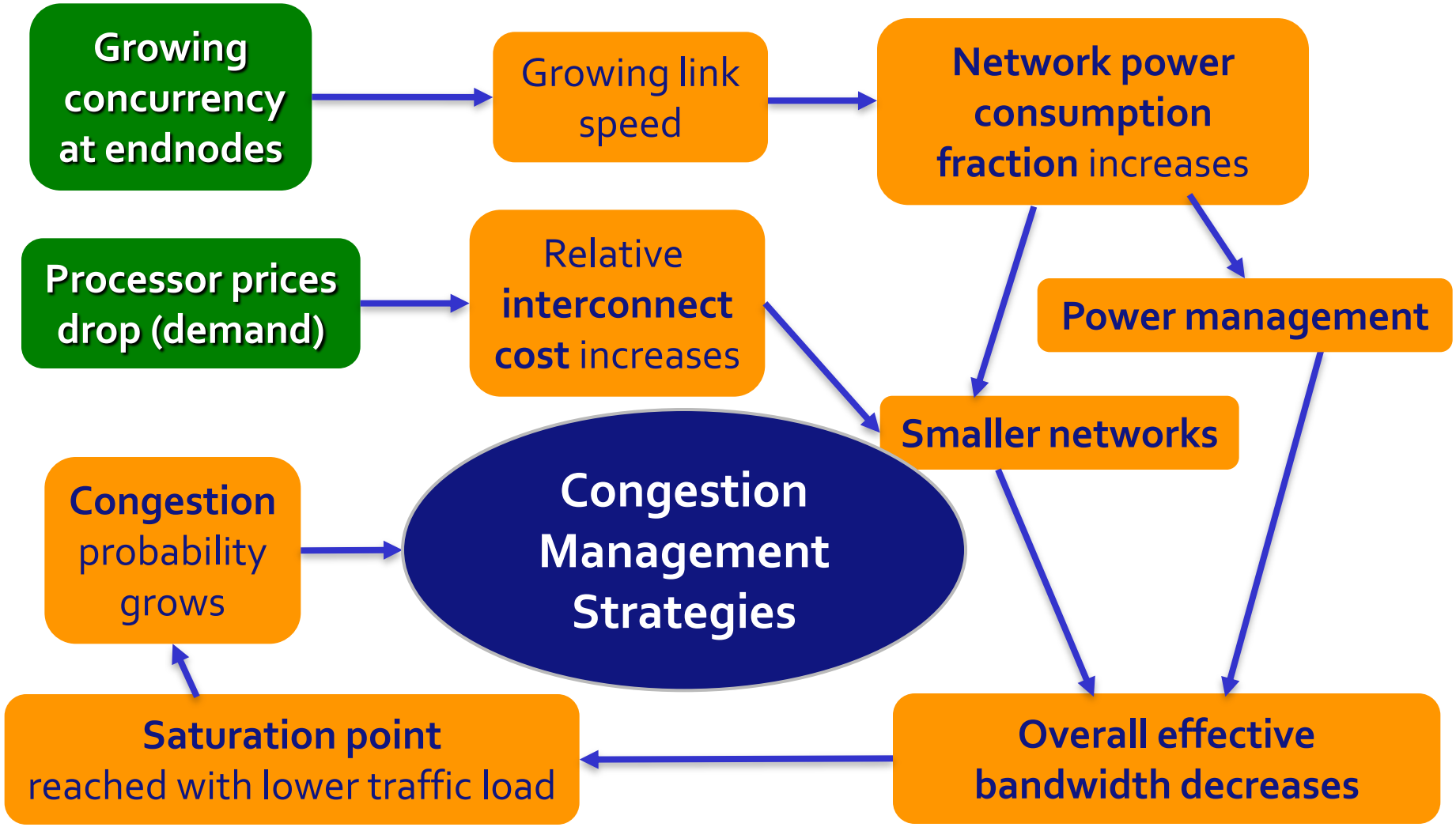
Why is congestion management necessary?



At saturation, **network performance** drops dramatically due to the cumulative effects of congestion situations

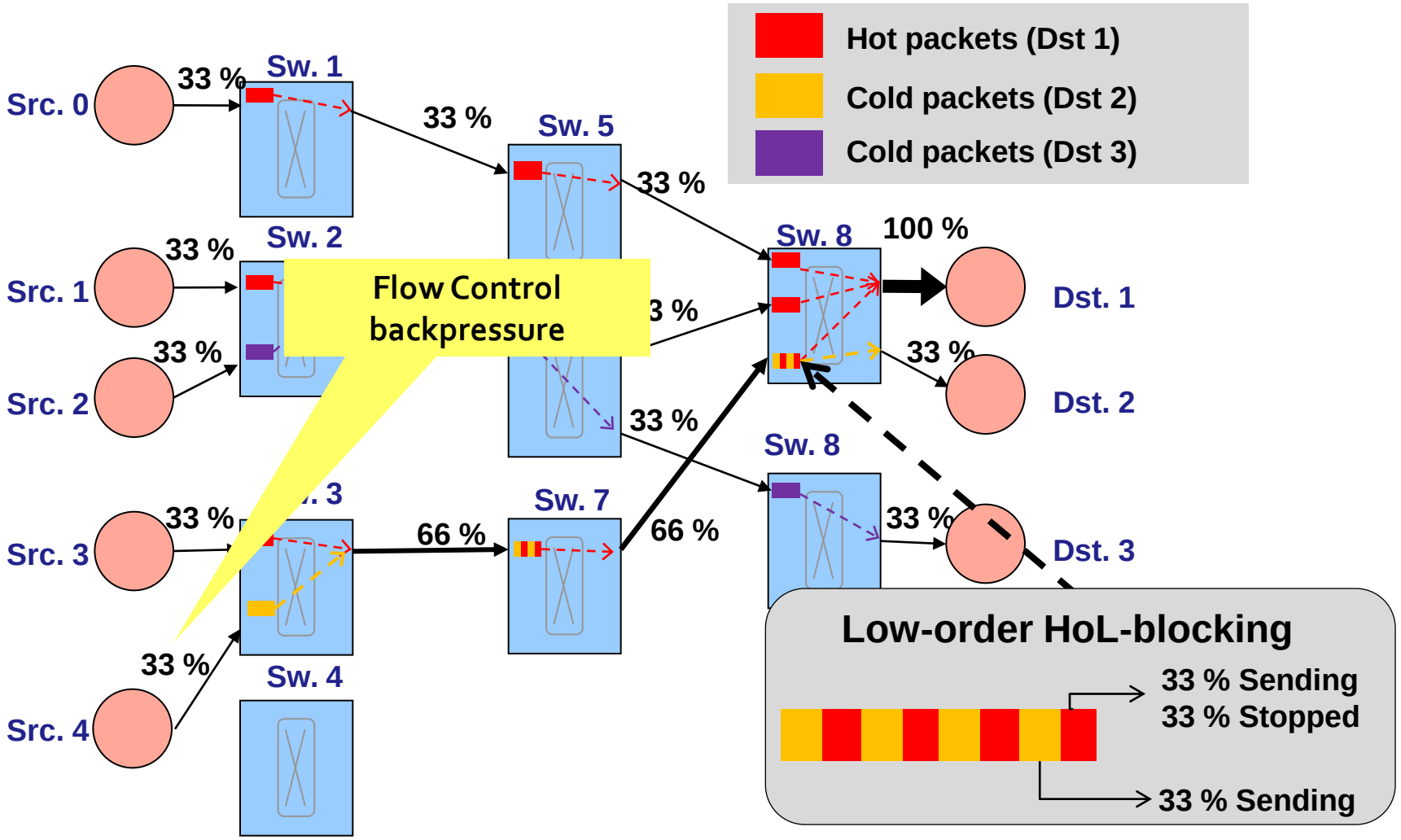
Congestion Management

Why is congestion management necessary?



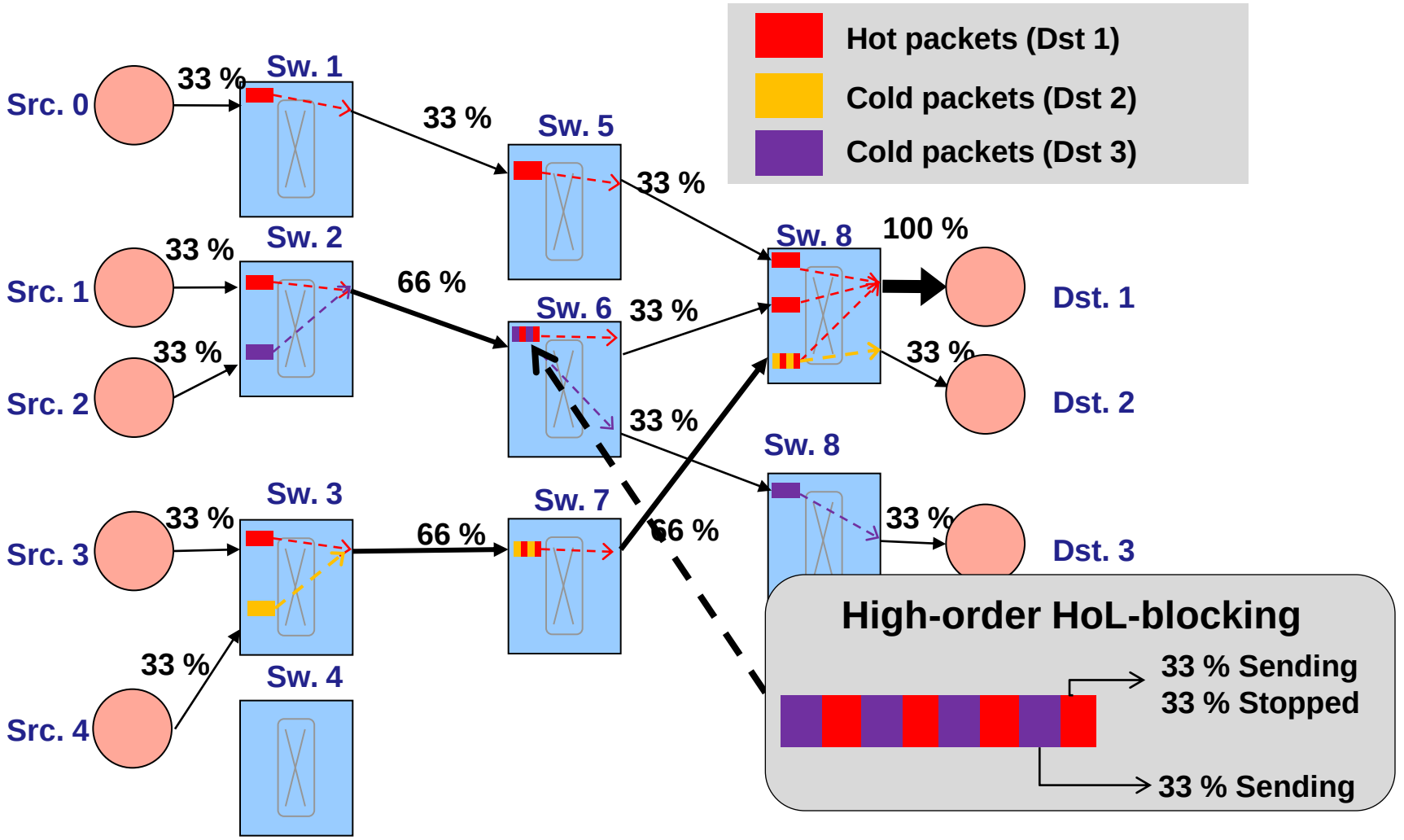
Congestion-Derived Problems

Low-Order Head-of-Line (HoL) Blocking



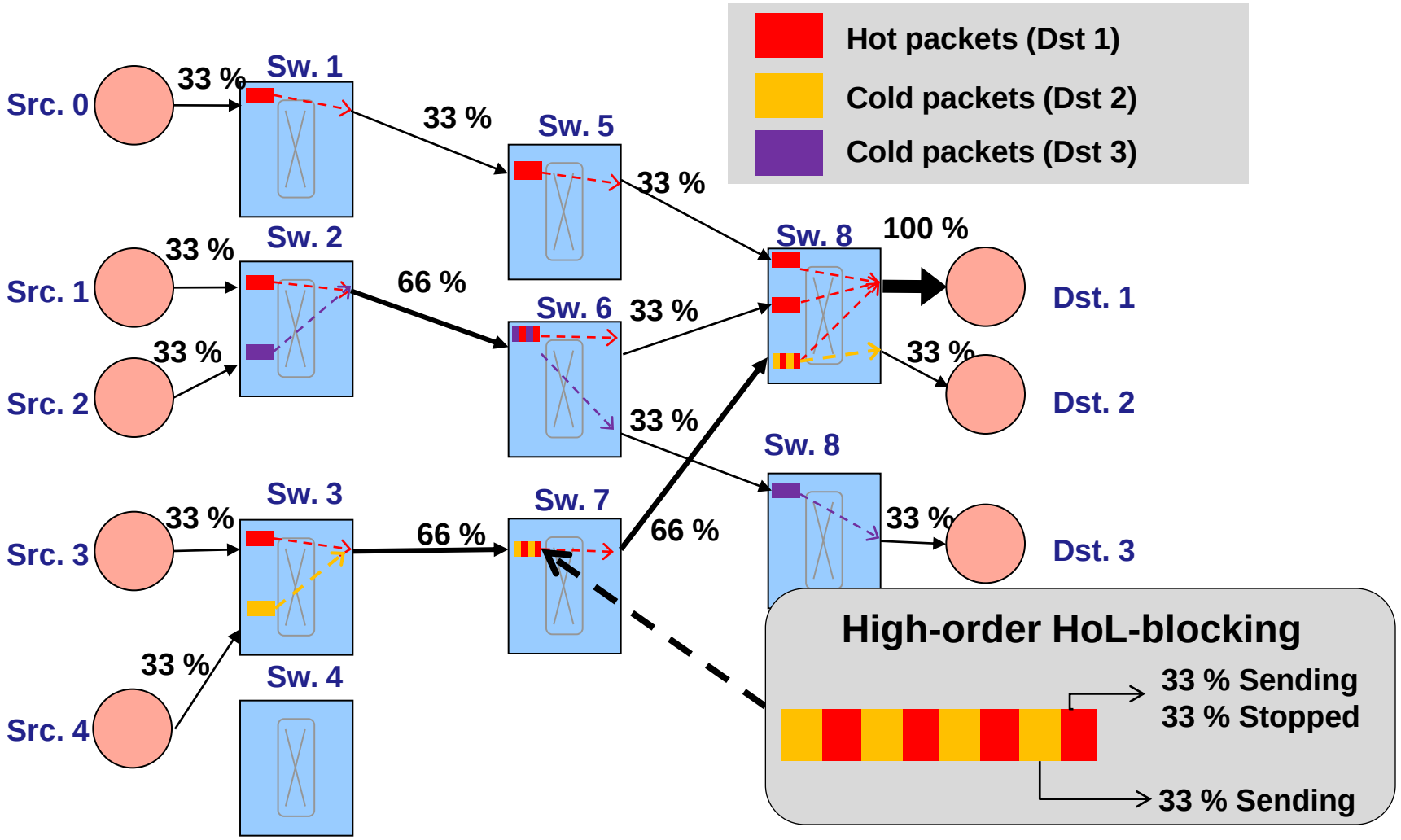
Congestion-Derived Problems

High-Order Head-of-Line (HoL) Blocking



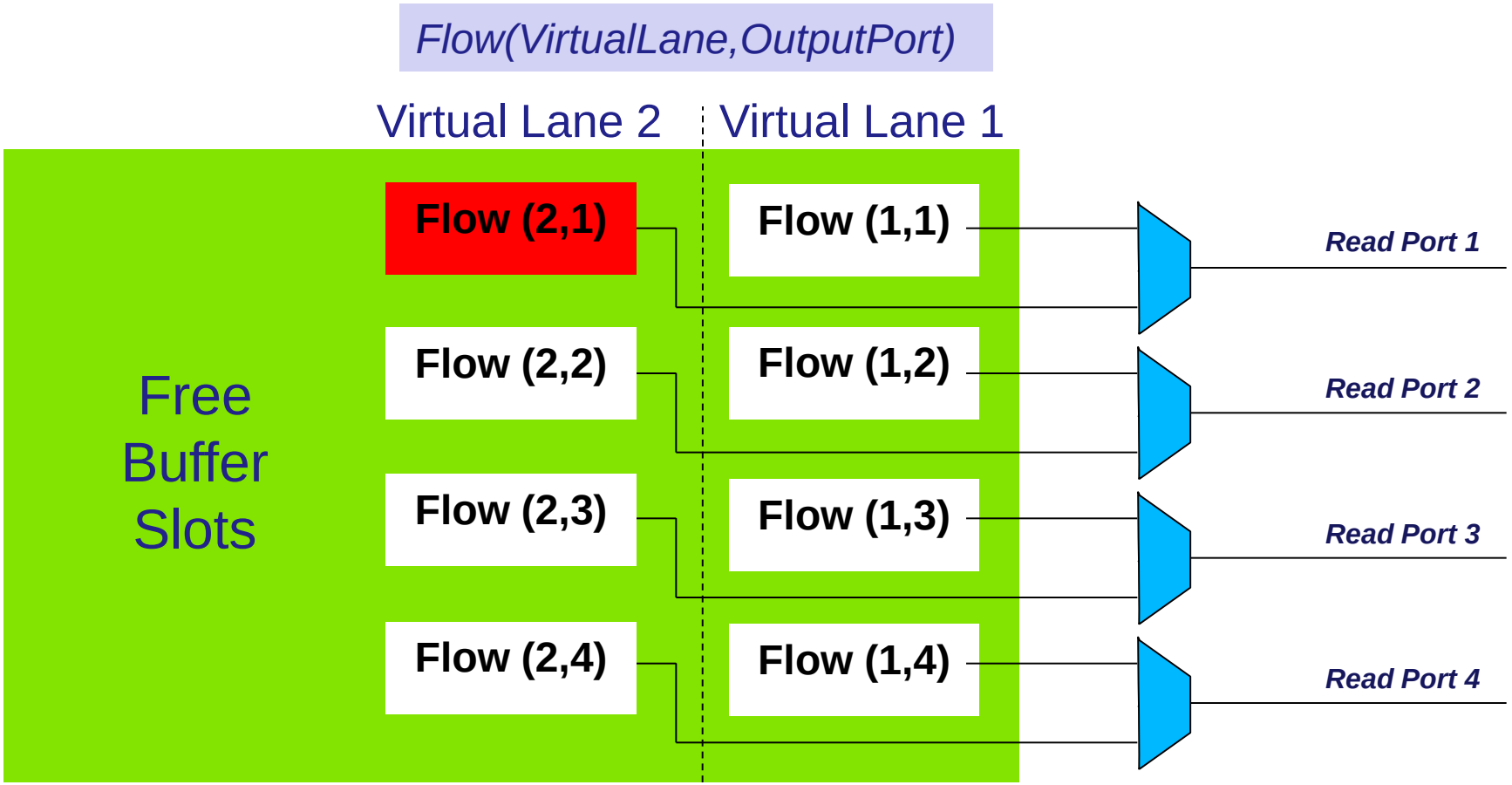
Congestion-Derived Problems

High-Order Head-of-Line (HoL) Blocking



Congestion-Derived Problems

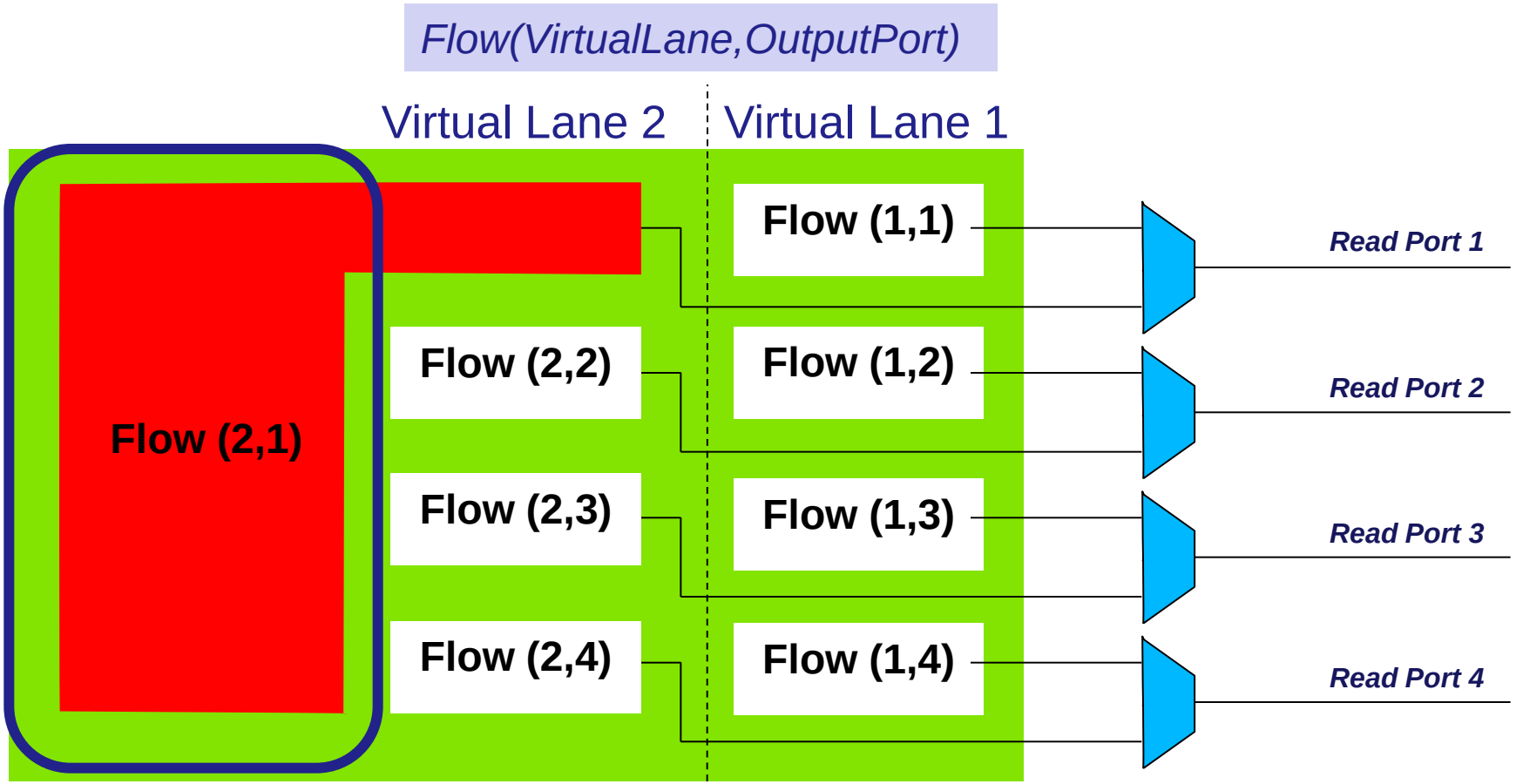
Buffer Hogging / Intra-VL hogging



Kenji Yoshigoe: Threshold-based Exhaustive Round-Robin for the CICQ Switch with Virtual Crosspoint Queues. ICC 2007: 6325-6329

Congestion-Derived Problems

Buffer Hogging / Intra-VL hogging



Kenji Yoshigoe: Threshold-based Exhaustive Round-Robin for the CICQ Switch with Virtual Crosspoint Queues. ICC 2007: 6325-6329

Congestion Management

How can congestion be managed?

- Different approaches to congestion management:
  - Packet dropping
  - Proactive techniques
  - Reactive techniques
  - HoL-blocking reduction techniques
  - HoL-blocking elimination techniques
  - Hybrid techniques

Congestion Management

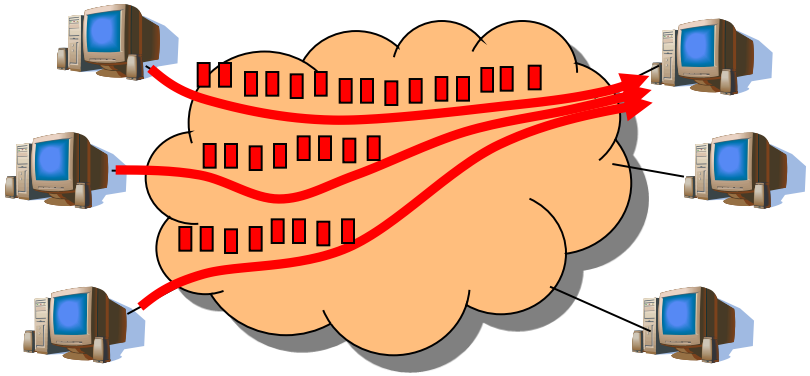
How can congestion be managed?

- Different approaches to congestion management:
  - Packet dropping. High latency variability. Not suitable for HPC
  - Proactive techniques. High setup time. Too slow for HPC
  - Reactive techniques. Available in IBA and OPA. Difficult to tune
  - HoL-blocking reduction techniques. VL mapping. Easy, effective
  - HoL-blocking elimination techniques. Dynamic queue allocation
  - Hybrid techniques. Combines benefits from reactive and HOL-blocking elimination techniques

Congestion Management

Reactive congestion management

- A.K.A. congestion recovery
- Injection limitation techniques (injection throttling) using closed-loop feedback
- Does not scale with network size nor link bandwidth
  - Notification delay (proportional to distance / number of hops)
  - Link and buffer capacity (proportional to clock frequency)
  - May produce traffic oscillations (closed loop system with pure delay)



Congestion Management

Reactive congestion management

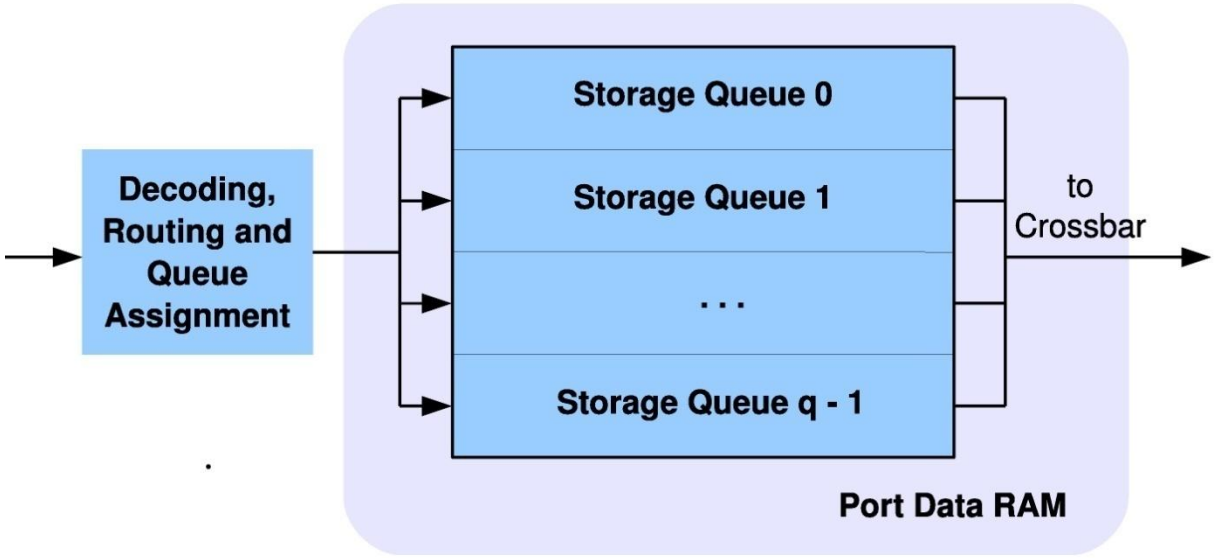
- Example: Infiniband FECN/BECN mechanism:
  - Two bits in the packet header are reserved for congestion notification
  - If a switch port is considered to be congested, the Forward Explicit Congestion Notification (FECN) bit in the header of packets crossing that port is set
  - Upon reception of such a "FECN-marked" packet, a destination will return a packet (Congestion Notification Packet, CNP) back to the source, whose header will have the Backward Explicit Congestion Notification (BECN) bit set
  - Any source receiving a "BECN-marked" packet will then reduce its packet injection rate for this traffic flow

E.G. Gran, M. Eimot, S.A. Reinemo, T. Skeie, O. Lysne, L. Huse, G. Shainer, "First experiences with congestion control in InfiniBand hardware", in Proceedings of IPDPS 2010, pp. 1–12.

Congestion Management

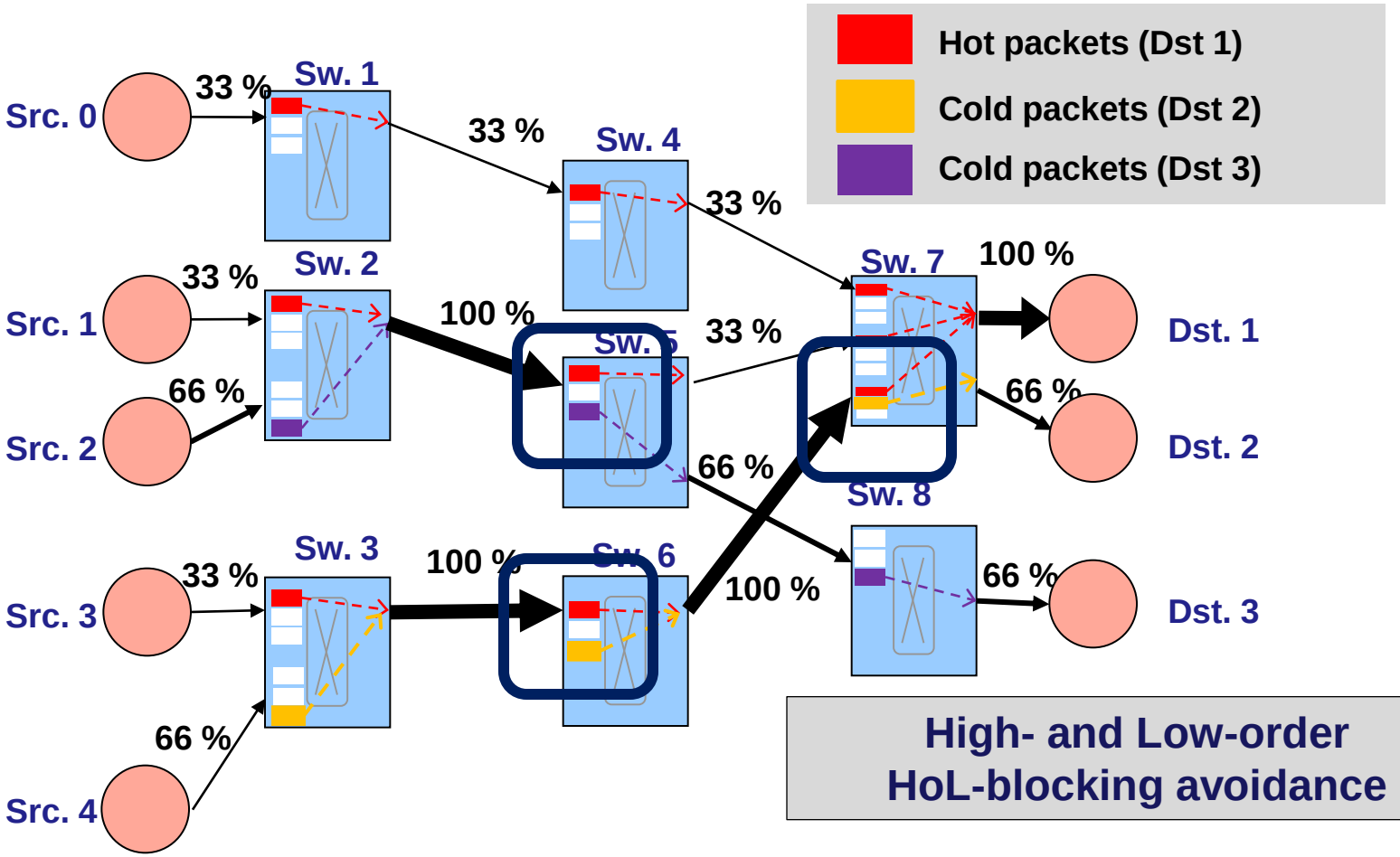
HoL-blocking reduction techniques

- These techniques rely on mapping groups of packet flows to different buffer queues (or VLs). Thus, each group becomes isolated and can't block the progress of flows in other groups
- Queuing schemes differ mainly in the criteria to map packets to queues and in the number of required queues per port



Congestion Management

Static mapping of hot flows to queues (or VLs)



Jesús Escudero-Sahuquillo, Pedro Javier García, Francisco J. Quiles, José Duato: ***An Efficient Strategy for Reducing Head-of-Line Blocking in Fat-Trees****. Euro-Par (2) 2010: 413-427*

Congestion Management

Generic Queue Mapping Schemes

Scheme	Low-order prevention	High-order prevention	Scalable (network size)	Scalable (#switch ports)
VOQnet	Yes	Yes	No	Yes
VOQsw	Yes	Partial	Yes	No
DAMQs	Yes	Partial	Yes	No
DBBM	Partial	Partial	Yes	Yes

In general, queue usage at some stages is not as efficient as it could be because they are "topology agnostic" schemes

Congestion Management

Topology- & Routing –Aware Mapping Schemes

Scheme	Topology	Low-order prevention	High-order prevention	Scalable (network size)	Scalable (#switch ports)
OBQA	Fat-Tree	Partial	Partial	Yes	Yes
vFtree	Fat-Tree	Yes	Partial	Yes	Yes
Flow2SL	Fat-Tree	Yes	Partial	Yes	Yes
BBQ	KNS	Partial	Partial	Yes	Yes

In general, they achieve similar or better performance than topology-agnostic schemes while requiring fewer queues per port, thus improving cost and performance

Congestion Management

vFtree – A commercial solution for FTs

HPCC test	Ftree (1 VL)	vFtree (3 VLs)	Improvement in %
Max. ping pong latency (ms)	0.002116	0.002116	0.0
Avg. ping pong latency (ms)	0.022898	0.013477	41.14
Min. ping pong latency (ms)	0.050500	0.043005	14.84
Naturally ordered ring latency (ms)	0.021791	0.014591	33.04
Randomly ordered ring latency (ms)	0.024262	0.015826	**34.77**
Max. ping pong bandwidth (MB/s)	1593.127	1594.338	0.07
Avg. ping pong bandwidth (MB/s)	573.993	830.909	44.75
Min. ping pong bandwidth (MB/s)	94.868	345.993	264.71
Naturally ordered ring bandwidth (MB/s)	388.969246	454.236253	16.78
Randomly ordered ring bandwidth (MB/s)	331.847978	438.604531	**32.17**

Wei Lin Guay, Bartosz Bogdanski, Sven-Arne Reinemo, Olav Lysne, Tor Skeie: ***vFtree - A Fat-Tree Routing Algorithm Using Virtual Lanes to Alleviate Congestion****. IPDPS 2011: 197-208*

Congestion Management

Example of Queue Mapping Scheme: Block Mapping

- At each port, map packets to queues according to the following expression:

$$SelectedQueue = \frac{Packet\_Destination \cdot Number\_Queues}{Number\_EndNodes}$$

- Easy implementation in **InfiniBand** technology:
  - Assign each packet an SL equal to the queue given by the above expression
  - Fill the SL-to-VL tables so that VL=SL

Pedro Yebenes, Jesús Escudero-Sahuquillo, Crispin Gomez-Requena, Pedro Javier García, Francisco J. Quiles and Jose Duato. ***BBQ: A Straightforward Queuing Scheme to Reduce HoL-Blocking in High-Performance Hybrid Networks****. Proceedings of Euro-Par 2013 .*

Congestion Management

Tailoring Queuing Schemes to Exascale Topologies

- The **queue assignment** criterion (i.e. the mapping policy) **should exploit the properties** of both network topology and routing scheme
- Metrics to analitically evaluate a **specific mapping of traffic flows (SLID,DLID) to SLs** (i.e. to VLs):
  - **VL Load**: Number of flows mapped to a VL at a specific port (strongly depends on the routing algorithm)
  - **Balancing Degree**: Difference between the maximum and minimum values for VL load (ideally zero)
  - **Overlapping Degree**: Meassures the number of flows simultaneously mapped to several VLs at the same port (must be low to reduce intra-VL hogging probability, ideally zero)

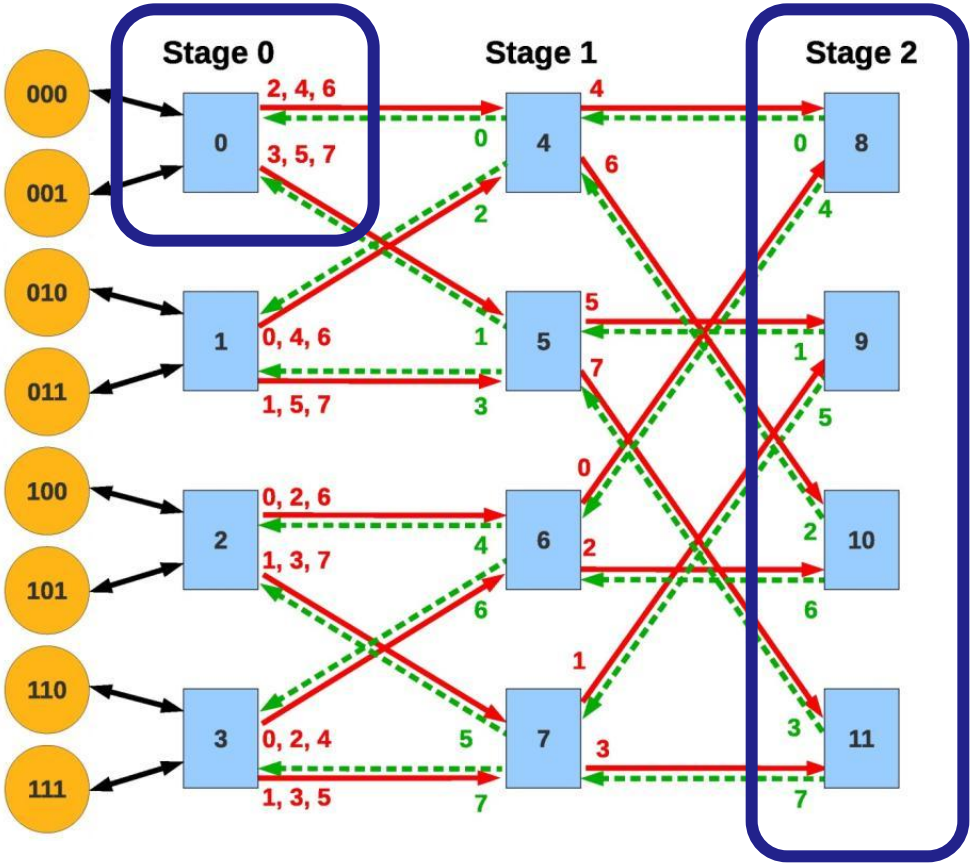
Congestion Management

Flow2SL - Adapting vFtree to Exascale FTs

- **Flow2SL** assumes **fat-tree** topologies and **D-mod-K routing**
- **Flow2SL** introduces **specific policies** for:
  - The mapping of **flows (SLID,DLID) to Service Levels**
  - The mapping of **Service Levels to Virtual Lanes**
- These policies leverage the traffic balance of the routing algorithm to **optimize the use of Virtual Lanes to separate different flows**
- **Impact of problems derived from congestion is reduced**:
  - High-order HoL-Blocking
  - Intra-VL buffer-hogging

Congestion Management

Flow2SL - Adapting vFtree to Exascale FTs



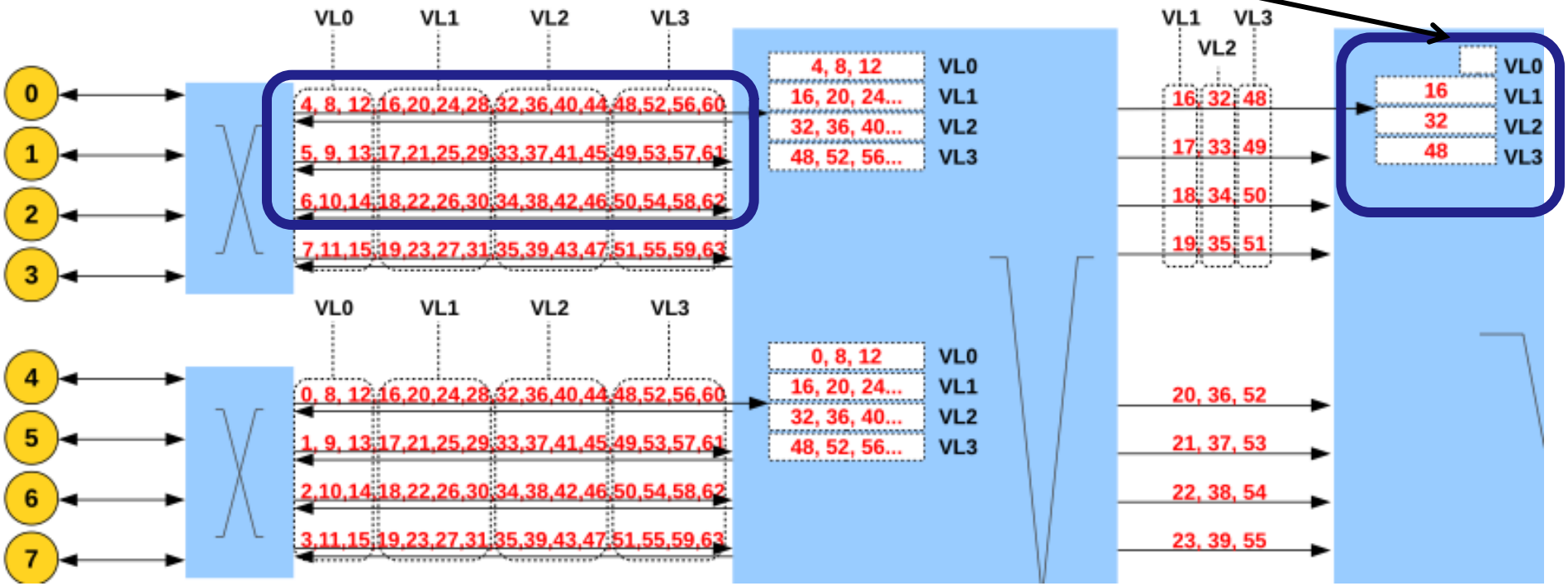
Balances the use of links by different paths

Congestion Management

Flow2SL - Adapting vFtree to Exascale FTs

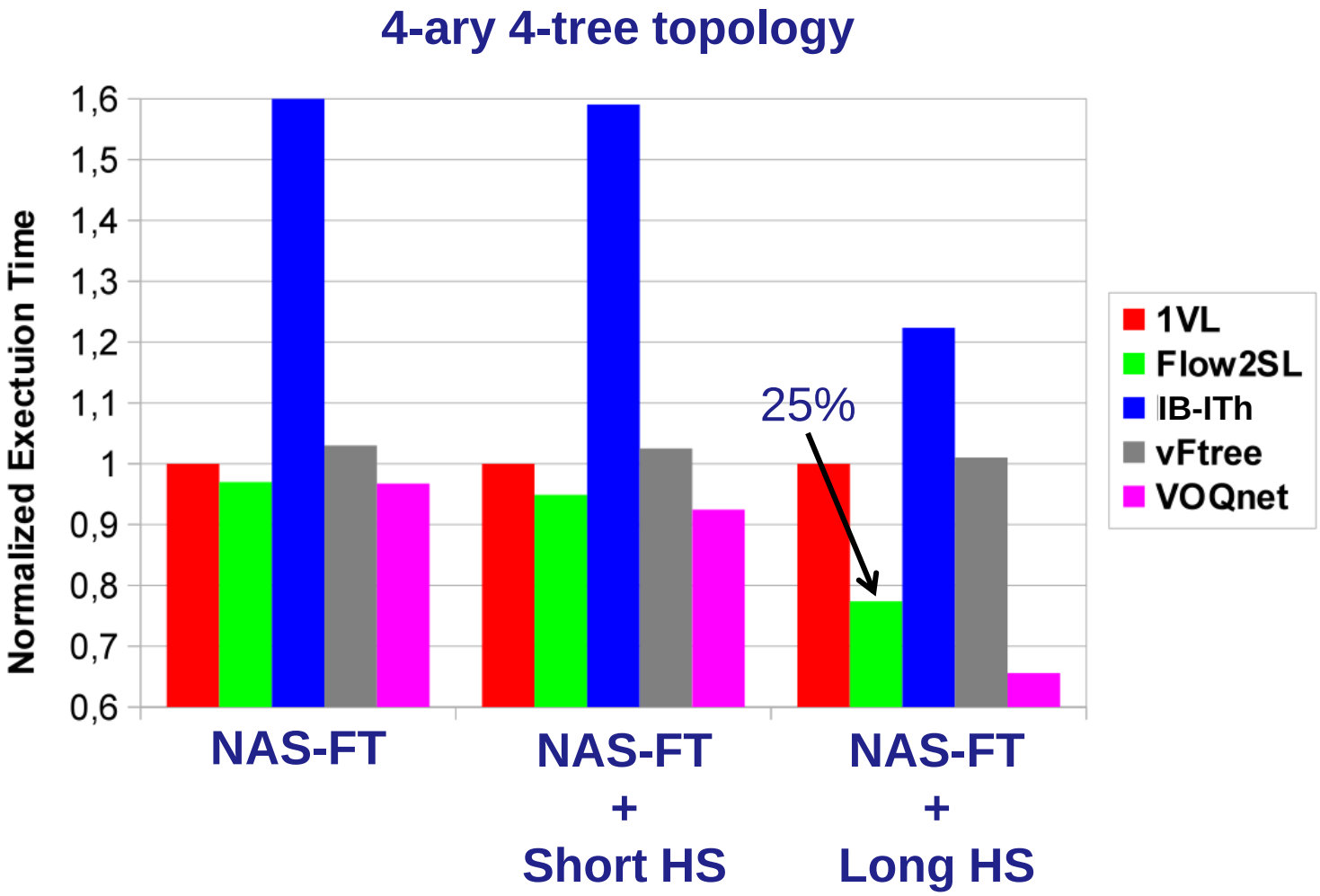
VL mapping to traffic flows according to DESTRO traffic balance

Reduction of the Intra-VL hogging in 3rd and next stages



Congestion Management

Flow2SL Simulation Results (NAS traffic traces)



Congestion Management

HoL-blocking elimination techniques

- Queue mapping schemes limit HoL-blocking and buffer-hogging as much as possible with the available queues, **but do not eliminate it completely.**
- A complete effectiveness in solving these problems would require dynamically allocating **extra queues to set aside packets that produce HOL-blocking**, paying an "extra-price" in terms of complexity and additional resources
- Several **Dynamic-Mapping Queuing Schemes** have been proposed:
  - **RECN** (deterministic source routing)
  - **FBICM** (deterministic distributed routing)
  - **DRBCM** (fat-trees with deterministic distributed routing, DESTRO-like routing)

Congestion Management

HoL-blocking elimination techniques

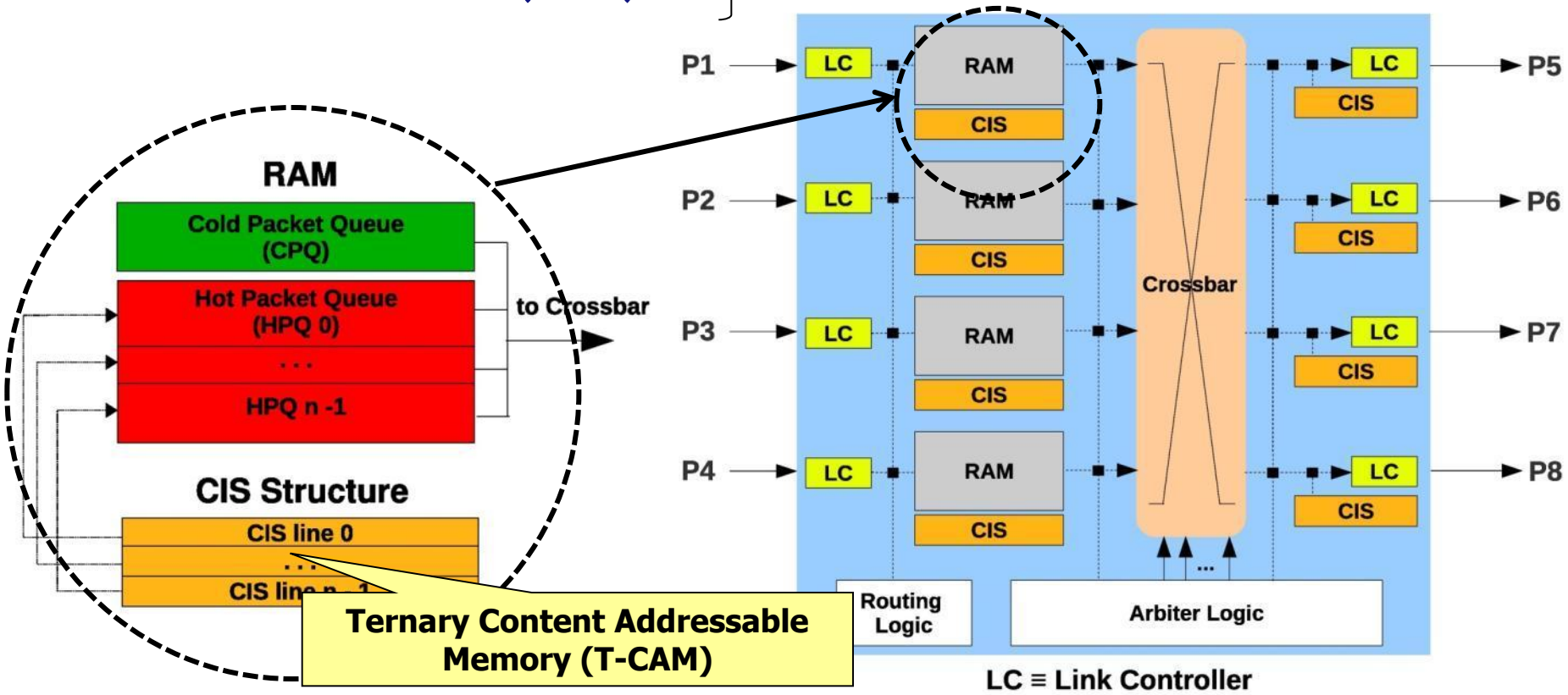
- **Congested points are detected** at any switch port of the network by measuring **queue occupancy**
- The **location** of any detected congested point is stored in a **control memory (a CAM or T-CAM line)** at any port forwarding packets towards the congested point
- A **special queue** associated with the CAM line is also **allocated to store only the packets that will cross that congested point**
- **Congestion information is progressively notified** to every port at upstream switches crossed by congested flows, where new CAM (or T-CAM) lines and special queues are allocated
- A packet arriving at a port is stored in the **standard queue** only if its **routing information does not match any CAM line**

Congestion Management

Example of Dynamic-Mapping Scheme: DRBCM

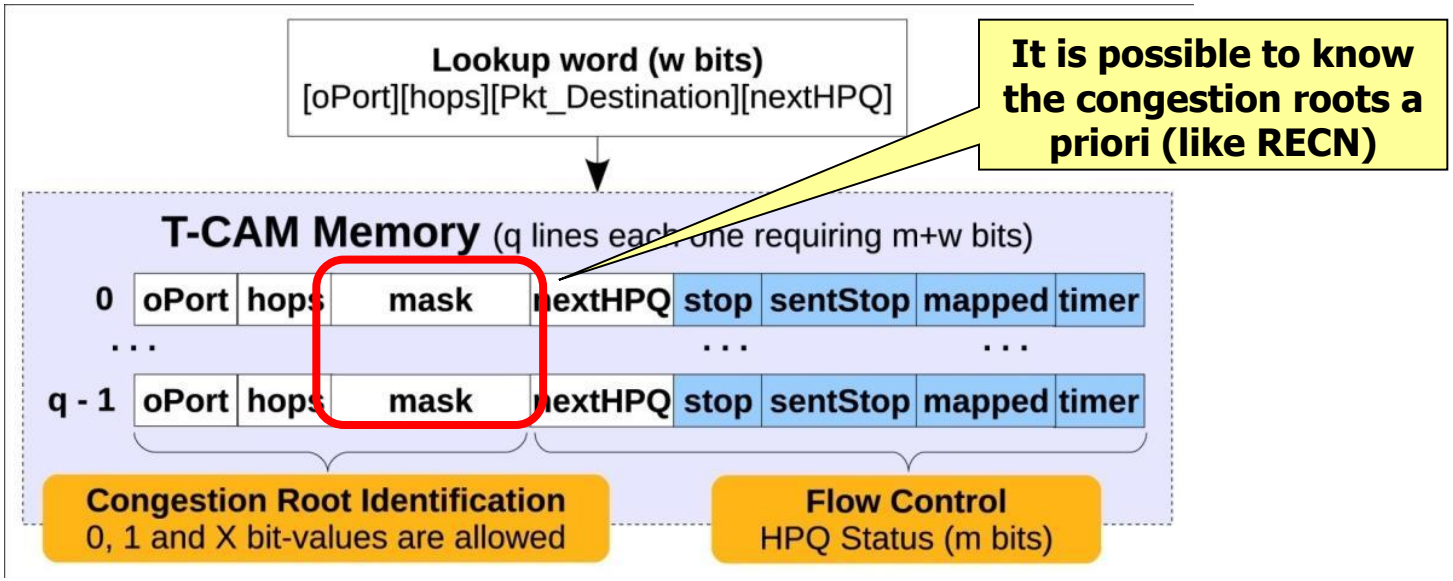
- **Cold Packets Queue (CPQ)**
- **Hot Packets Queues (HPQ)**

To **Isolate** hot flows from cold ones



Congestion Management

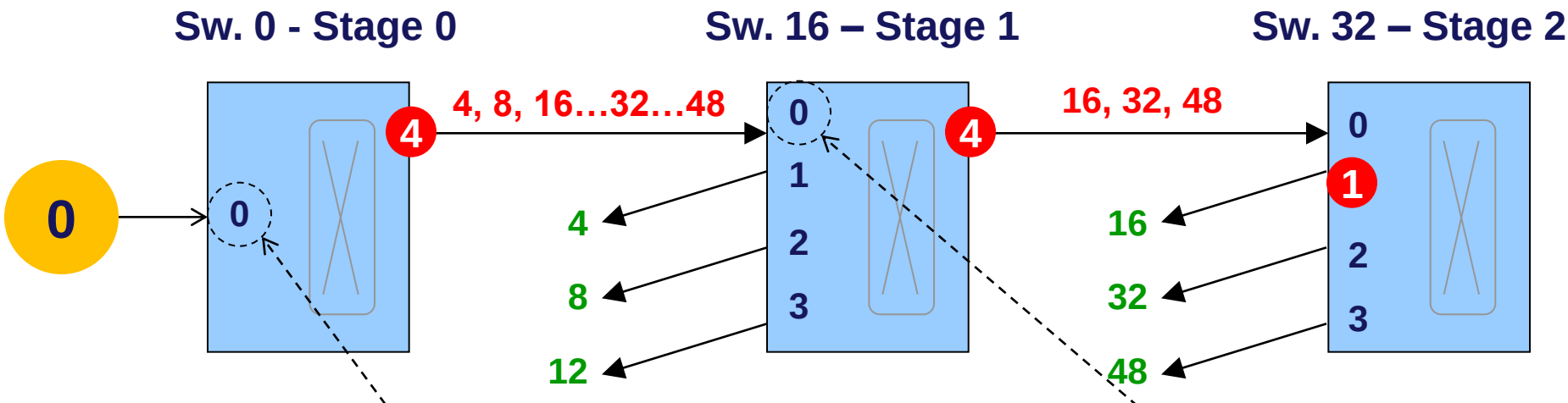
Example of Dynamic-Mapping Scheme: DRBCM



- The **mask field** (using values 0, 1 and X) **identifies all the destinations crossing a congestion root**
- The mask is **updated** as congestion information is **propagated**
- The rest of the fields are required to manage the T-CAM line operations (**flow-control, deallocation timer,** etc.)

Congestion Management

Example of Dynamic-Mapping Scheme: DRBCM



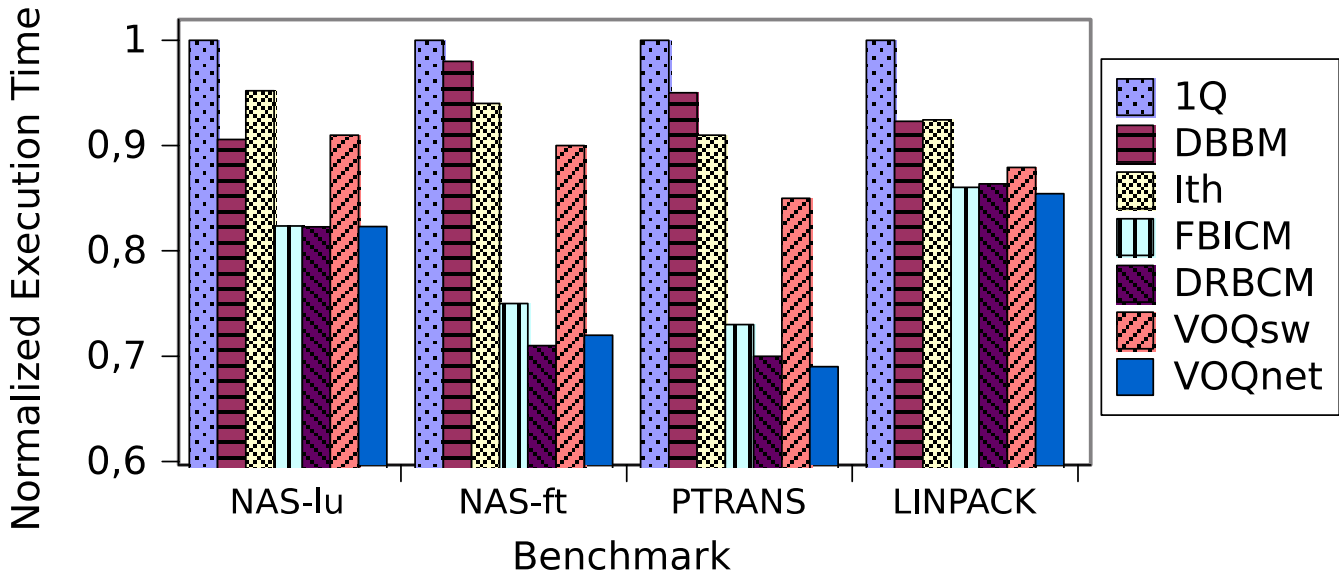
#	Root	Mask	Hops	oPort
1	Rz	010000	3	1
2	Ry	xx0000	2	4
3	Rx	xxxx00	1	4

#	Root	Mask	Hops	oPort
1	Rz	010000	2	1
2	Ry	xx00xx	1	4

Congestion Management

Example of Dynamic-Mapping Scheme: DRBCM

- **Execution Time of Real-Traffic Traces**



4-ary 4-tree
256 nodes

Jesus Escudero-Sahuquillo, Pedro J. Garcia, Francisco J. Quiles, Jose Flich, Jose Duato, ***An Effective and Feasible Congestion Management Technique for High-Performance MINs with Tag-Based Distributed Routing,*** *IEEE Transactions on Parallel and Distributed Systems, October.2013.*

Congestion Management

Drawbacks of Dynamic Mapping Schemes

- In scenarios with several different congested points, it is possible **to run out of special queues** at some ports
- The need for **CAMs** at switch ports increases **switch complexity, implementation cost and required silicon area per port**
- **Unfairness** in the **scheduling of hot flows** may appear

Congestion Management

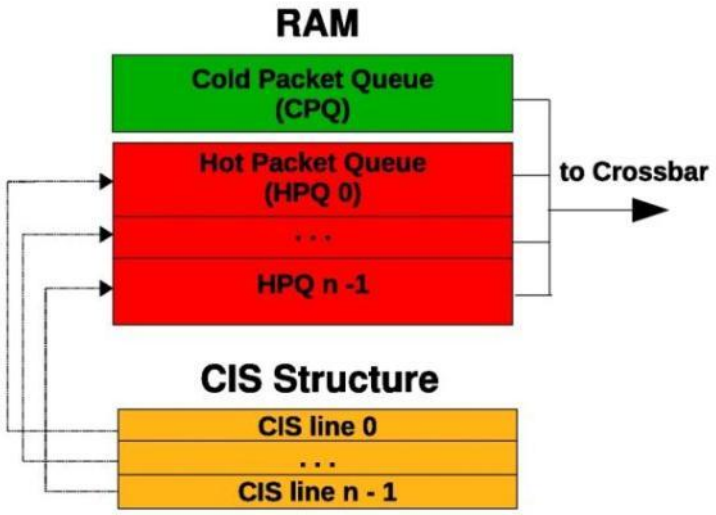
Hybrid Congestion Management Strategies

- **Combining Injection Throttling and Dynamic Mapping**:
  - Using **Dynamic Mapping to quickly and locally eliminate HoL-blocking,** propagating congestion information and allocating queues as necessary
  - Using **Injection Throttling to slowly eliminate congestion**, deallocating special queues whenever possible
  - Use of **Dynamic Mapping provides immediate response** and allows **reactive congestion management to be tuned for slow reaction**, thus avoiding oscillations
  - **Injection Throttling** drastically **reduces Dynamic Mapping buffer requirements** (just one or two queues per port)

Congestion Management

Example of Hybrid Congestion Management: CCFIT

- **Input ports organized like in RECN** (CAMs at input/output ports)
- **HPQs** assigned when the **CPQ exceeds a threshold**
- **Output ports in congestion state** , when HPQ reaches a **High Threshold**
- **Packets are marked (FECN) at output ports in congestion state**
- Output port congestion state is deactivated when a HPQ reaches the **Low Threshold**, and **no HPQs exceed that threshold**

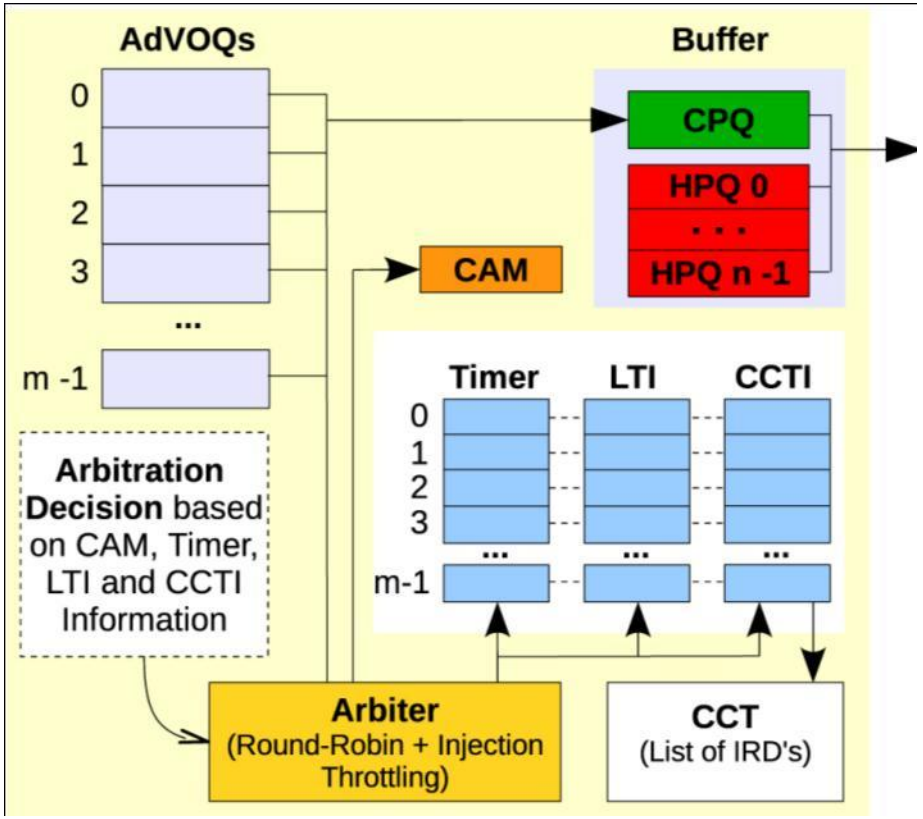


Jesús Escudero-Sahuquillo, Ernst Gunnar Gran, Pedro Javier García, Jose Flich, Tor Skeie, Olav Lysne, Francisco J. Quiles, José Duato: ***Combining Congested-Flow Isolation and Injection Throttling in HPC Interconnection Networks****. ICPP 2011: 662-672*

Congestion Management

Example of Hybrid Congestion Management: CCFIT

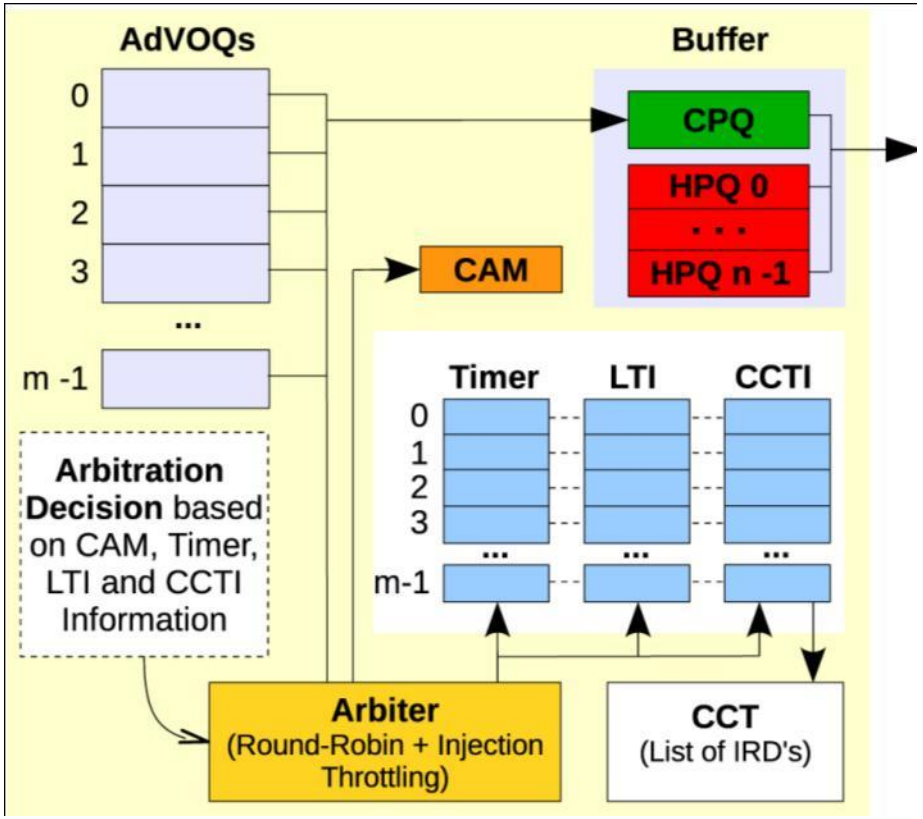
- HCAs must support both RECN-like queues + CAMs and typical InfiniBand Injection-Throttling structures (CCT, Timers, etc.)
- HCA arbiters must take into account information from different structures



Congestion Management

Example of Hybrid Congestion Management: CCFIT

- An IA receives a BECN and applies an **Injection Rate Delay** (IRD) value
- IRD = CCT[CCTI[destination]]
- CCTI[destination] may be increased if more BECNs are received at the IA
- A Timer[destination] is set on for a new calculated CCTI[destination]
- When Timer[destination] expires CCTI[destination] is decreased in one unit, then **the injection is slowly restored** (if no more BECNs arrive).
- **HPQs prevents HoL-Blocking** at IAs

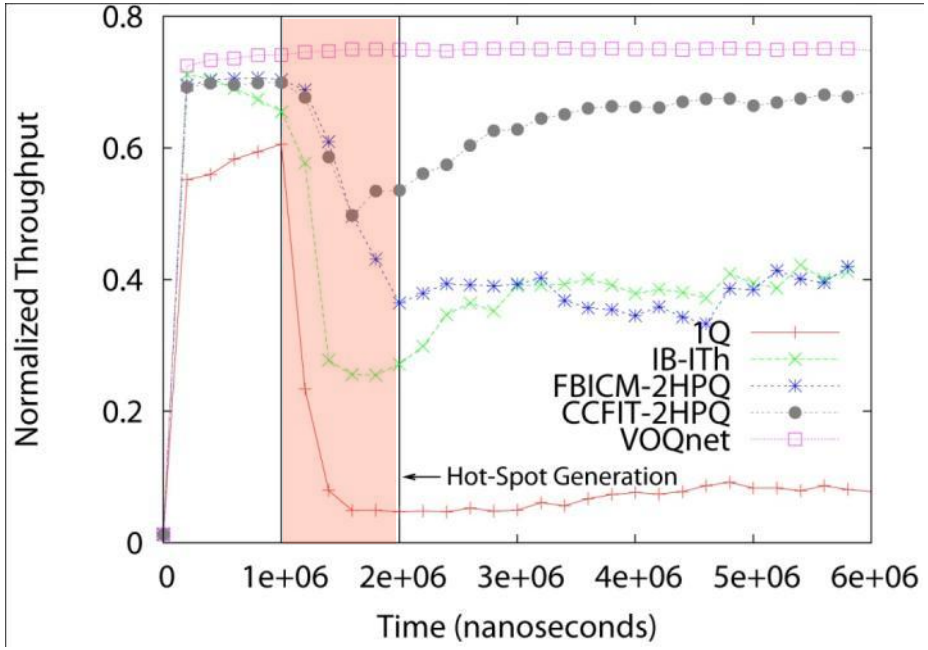


Jesús Escudero-Sahuquillo, Ernst Gunnar Gran, Pedro Javier García, Jose Flich, Tor Skeie, Olav Lysne, Francisco J. Quiles, José Duato: ***Combining Congested-Flow Isolation and Injection Throttling in HPC Interconnection Networks****. ICPP 2011: 662-672*

Congestion Management

Example of Hybrid Congestion Management: CCFIT

- **Normalized Throughput vs. Time, 4 Hot-Spots**



4-ary 4-tree
256 nodes

Jesús Escudero-Sahuquillo, Ernst Gunnar Gran, Pedro Javier García, Jose Flich, Tor Skeie, Olav Lysne, Francisco J. Quiles, José Duato: ***Combining Congested-Flow Isolation and Injection Throttling in HPC Interconnection Networks****. Proceedings of ICPP 2011:*

Congestion Management

Our philosophy

The **real problem** is not the congestion itself, but its **negative effects** (HoL-blocking and Buffer Hogging)

By **preventing HoL-blocking and Buffer Hogging**, congestion becomes harmless

And it does not matter the time spent to throttle or remove it

Congestion Management

Proposed approach

- Queue mapping schemes are easy and cheap to implement
  - They act as fences to **constrain the impact of HOL-blocking** produced by congested flows to a small fraction of total traffic
- Dynamic queue allocation schemes are complex and may increase switch latency, but they very effective and react fast
  - They effectively **eliminate all the HOL-blocking**, and thus, all the negative effects of congestion trees while requiring few extra resources
  - They **react very quickly** and **prevent performance degradation**
- Injection throttling is suitable as a complementary technique
  - Tuned to react slowly to avoid oscillations. **Releases dynamic queues**

Outline

- Introduction
- Exascale Interconnection Networks
- Topologies: Scalability, Routing and Fault-Tolerance
- Power Consumption
- Congestion Management
- **Final Observations**

Final Observations

- Most companies currently competing to design systems that could reach Exascale performance want to be able to reuse system components to commercialize smaller systems
  - Reluctant to include mechanisms not useful for smaller systems
  - Difficult trade-off regarding switch buffer sizes
    - Large buffers required to run long links at full speed. Not needed for short links
    - Large buffers consume silicon area and may lead to a smaller number of ports
- When comparing topologies, layout and link length matter
  - Long links must use optical fiber to reduce attenuation, increasing cost
  - A 40 meter cable has a propagation delay twice that of an OPA switch

Final Observations

- Production HPC systems will include storage subsystems
  - Either storage boxes or disks will be directly attached to the network
  - Storage should be distributed in such a way that bisection bandwidth could be minimized without leading the network to saturation
  - Switches should implement QoS mechanisms to separate traffic classes
  - Either storage is replicated in some way, or fully disjoint paths should be provided to access storage subsystems/devices (fat trees cannot do this)
- Nobody studied the combined behavior of power management and congestion management
  - Not even by simulation. Nor their separate behavior in huge systems

Acknowledgements

Special thanks to **Pedro J. García**, **Jesús Escudero**, and **Francisco J. Quiles**, from University of Castilla – La Mancha

Questions???