DEPARTAMENTO DE SISTEMAS INFORMÁTICOS

UCLM UNIVERSIDAD DE CASTILLA-LA MANCHA

Las redes de interconexión
una de las claves del reto Exascale

Francisco José Quiles Flor
Universidad de Castilla-La Mancha
SPAIN
Francisco.Quiles@uclm.es

Style Powered by: ACDA

10 de noviembre 2017, Granada

Outline

- Introduction
- The context
- Congestion basics
- Should we care about congestion in current and future interconnection networks?
- Solutions: How can congestion be managed?
- Challenges

TOP500

¿Qué es el reto Exascale?

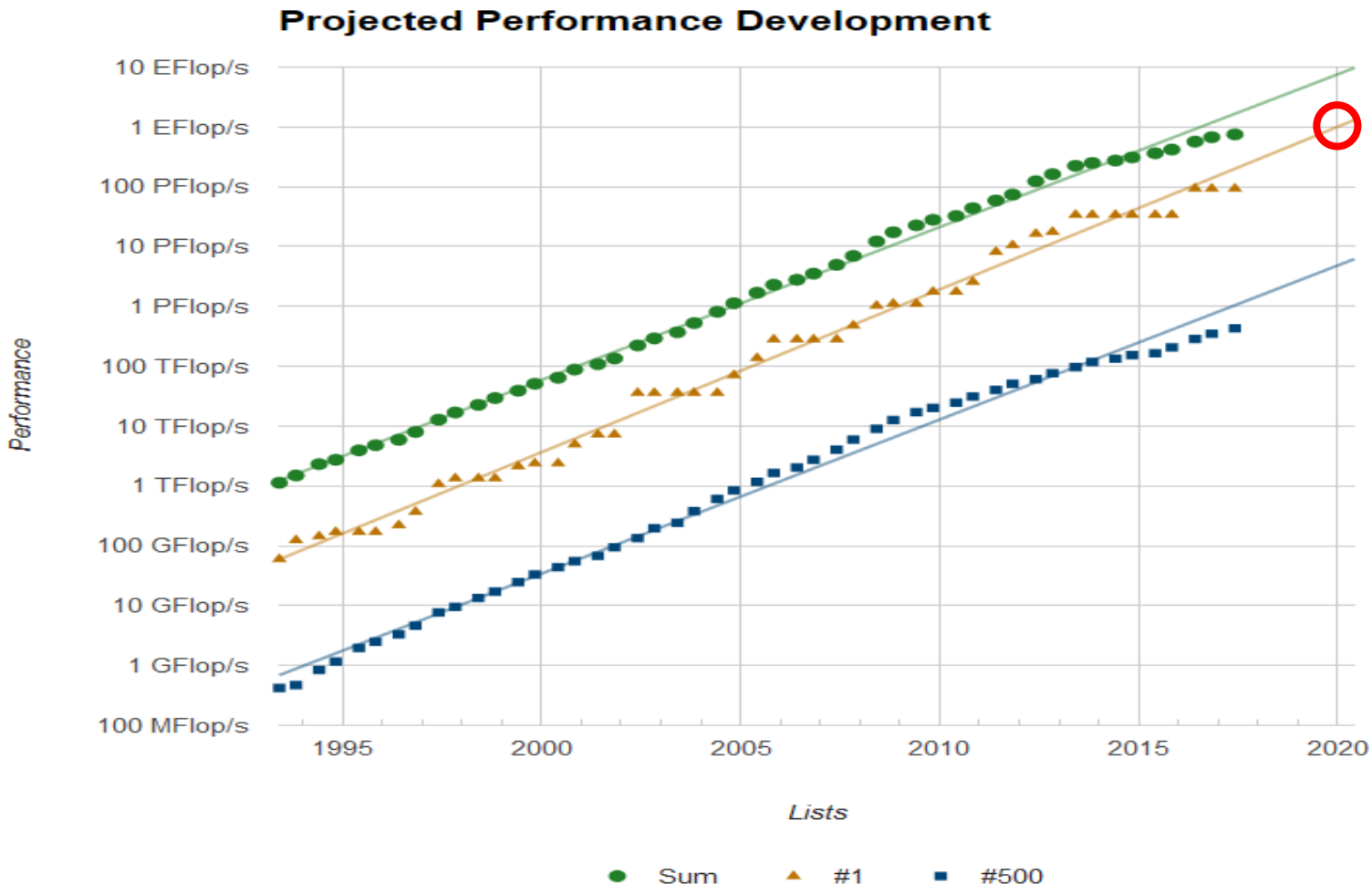
- El TOP500 son los 500 computadores científicos más rápidos del mundo.
  - Patrocinado por:
    - Universidad de Manheim
    - Universidad de Tennesse
    - NERSC/LBNL
  - El mejor rendimiento con el Linpack benchmark

www.top500.org

TOP5 – junio 2017

Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	National Supercomputing Center in Wuxi China	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRCPC	10,649,600	93,014.6	125,435.9	15,371
2	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
3	Swiss National Supercomputing Centre (CSCS) Switzerland	**Piz Daint** - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 Cray Inc.	361,760	19,590.0	25,326.3	2,272
4	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
5	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890

Evolución en el TOP500



Evolución en Supercomputación

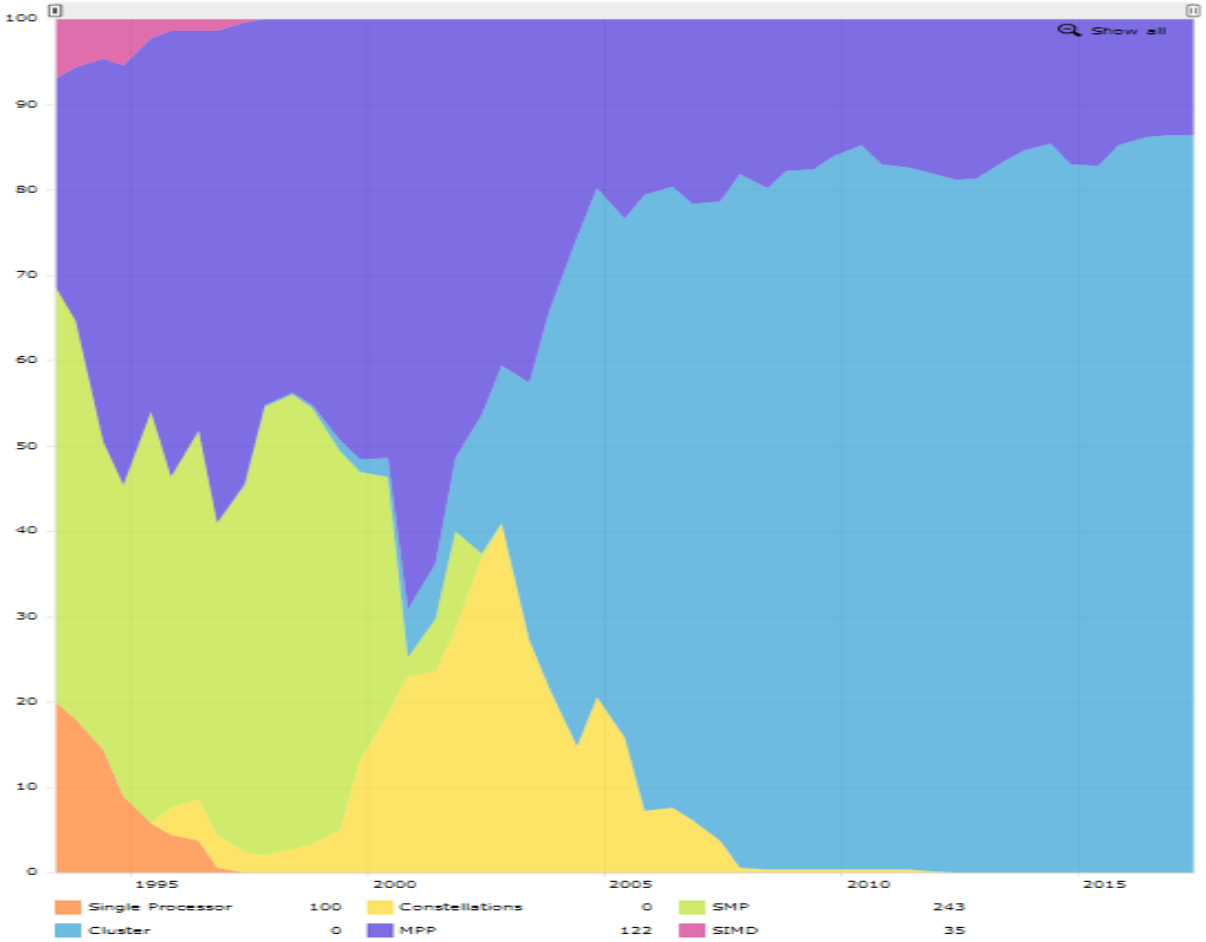
1950	Univac-1	1 Kflops (10^3 flop/seg)
1965	IBM 7090	100 Kflops (10^5 flop/seg)
1970	CDC 7600	10 Mflops (10^7 flop/seg)
1976	Cray-1	100 Mflops (10^8 flop/seg)
1982	Cray X-MP	1 Gflops (10^9 flop/seg)
1990	TMC CM-2	10 Gflops (10^{10} flop/seg)
1995	Cray T3E	100 Gflops (10^{11} flop/seg)
2000	IBM SP	1 Tflops (10^{12} flop/seg)
2002	Earth Simulator	40 Tflops (4×10^{12} flop/seg)
2004	BlueGene/L	70,72 Tflops ($70,72 \times 10^{12}$ flop/seg)
2009	Jaguar-Cray	1,759 Pflops ($1,759 \times 10^{15}$ flop/seg)
2010	Tianhe 1A	2,566 Pflops ($2,566 \times 10^{15}$ flop/seg)
2017	SunWay	93 Pflops (93×10^{15} flop/seg)

Superando Star Trek

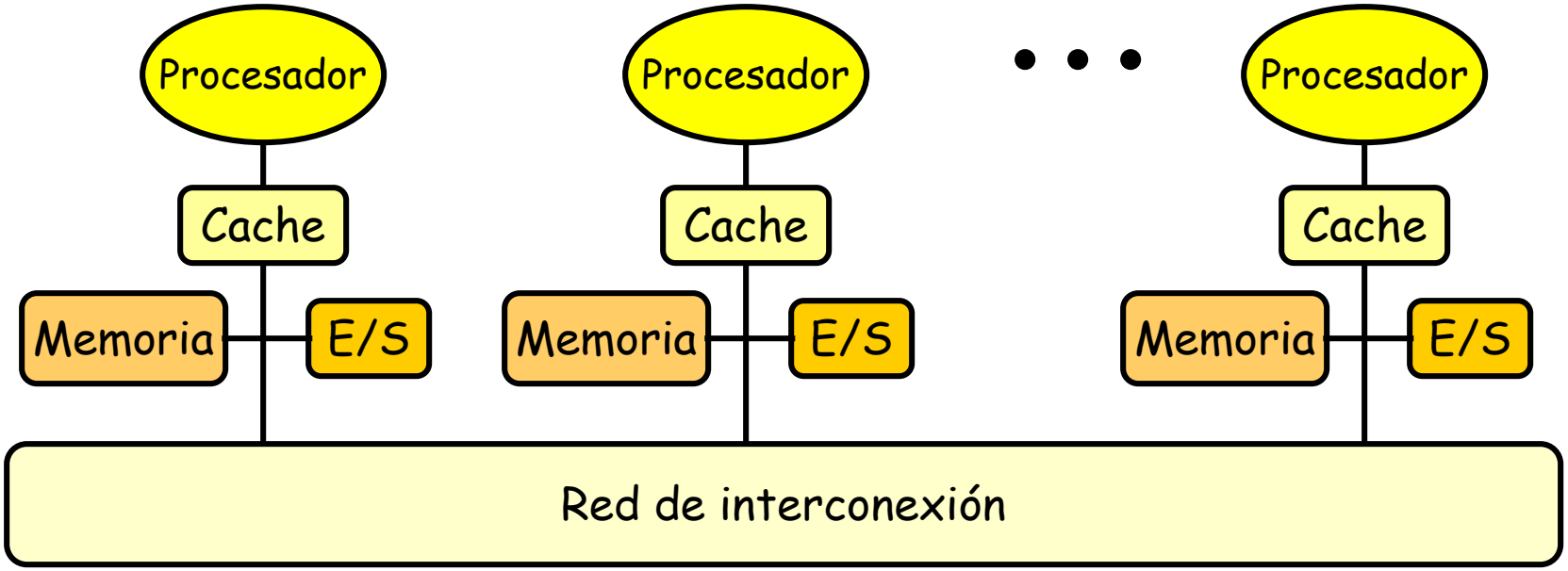
- En un episodio de Star Trek: TNG se cita que la potencia de cálculo del ordenador de la nave (Data) es 60 trillones de operaciones por segundo, es decir 60 Te
- El episodio es de 1989 y
más rápido que el orde
- Hoy en día, en 2017, 27
ordenador más rápido
más rápido que DATA,
DATA, 2338.

Ni la ciencia fi

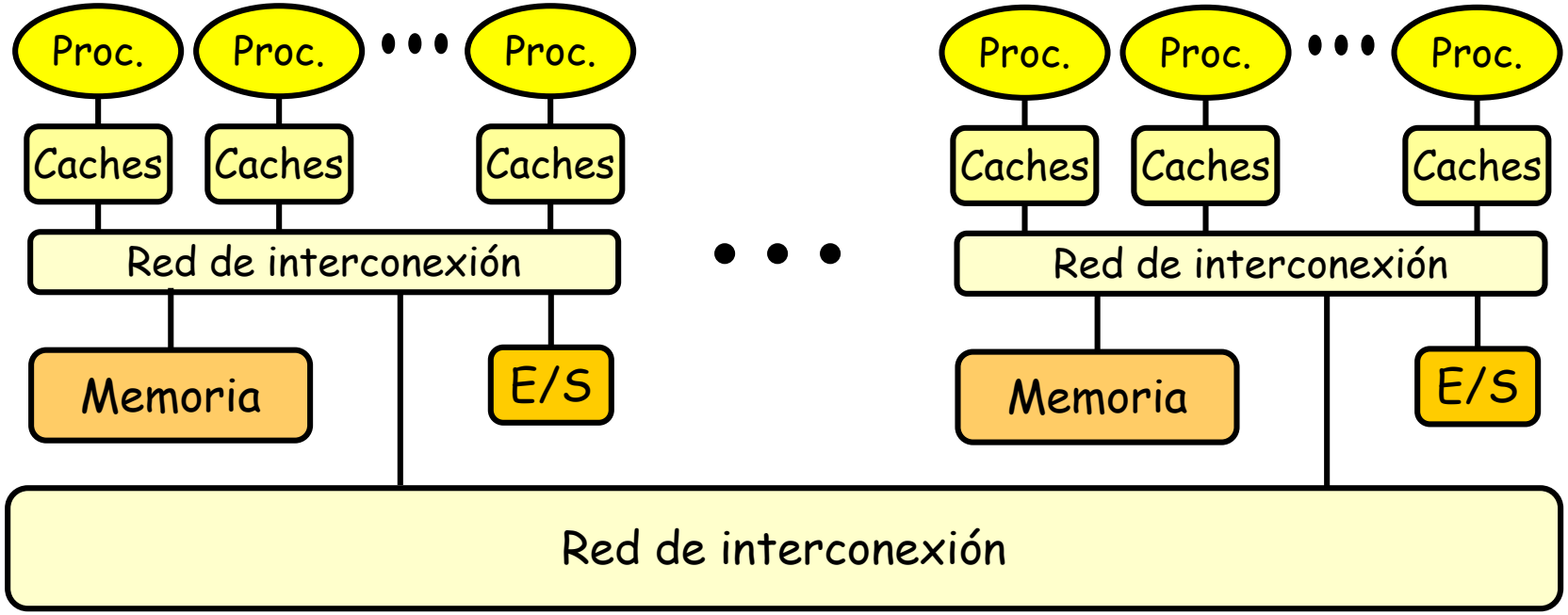
Evolución del uso de las arquitecturas



Arquitectura de memoria distribuida

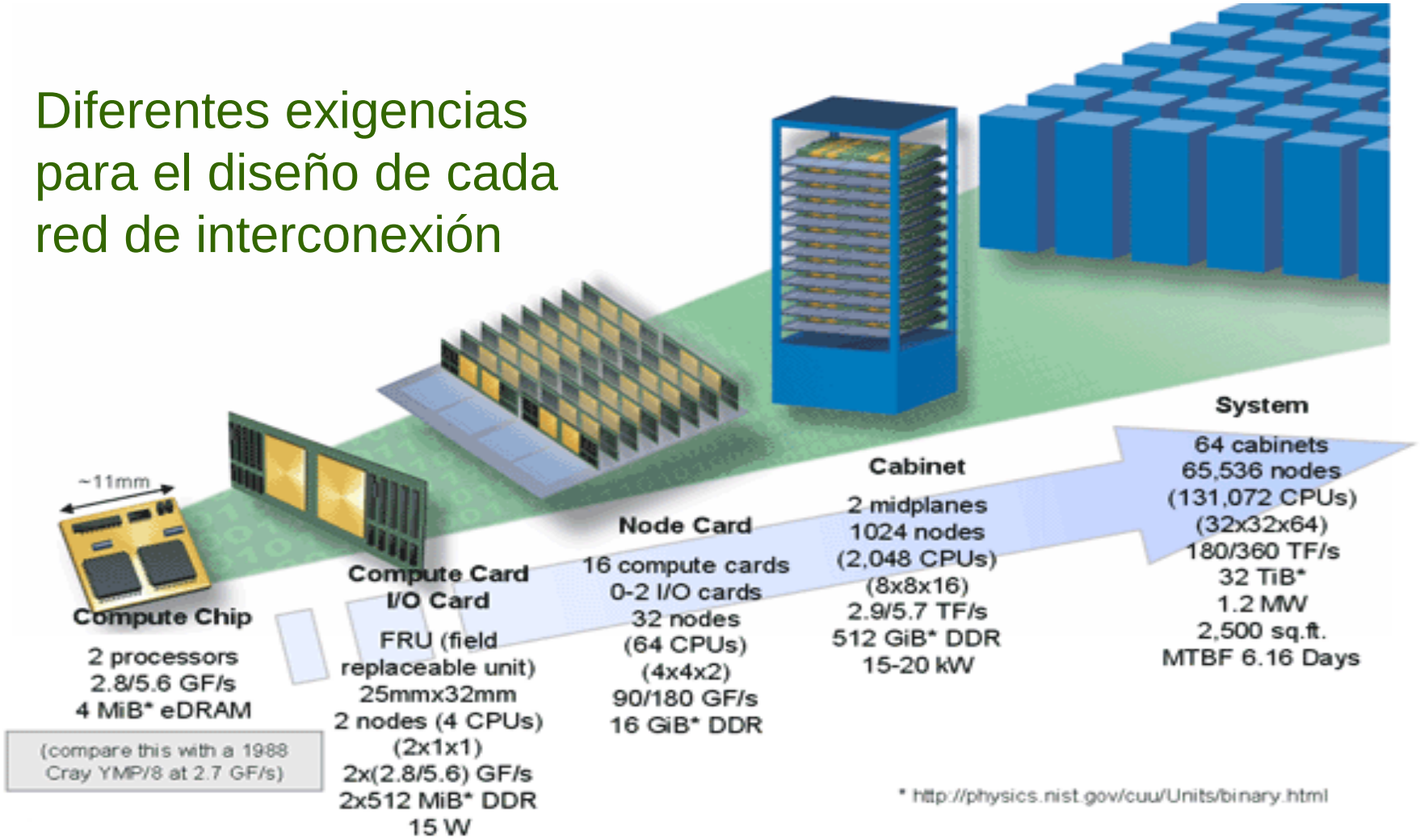


Memoria compartida distribuida

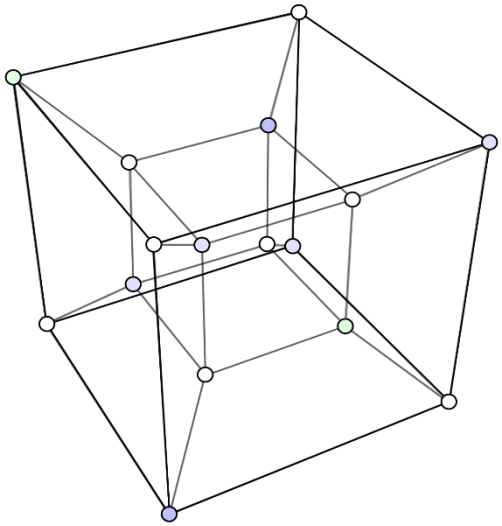


Blue Gene (2004)

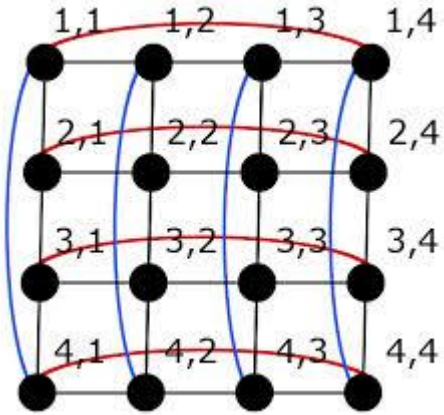
Diferentes exigencias para el diseño de cada red de interconexión



Topologías

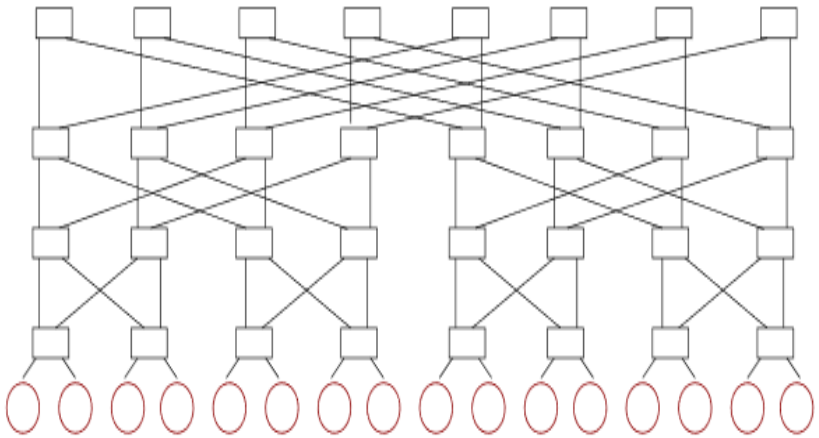


Hipercubo, Intel iPSC (1984)

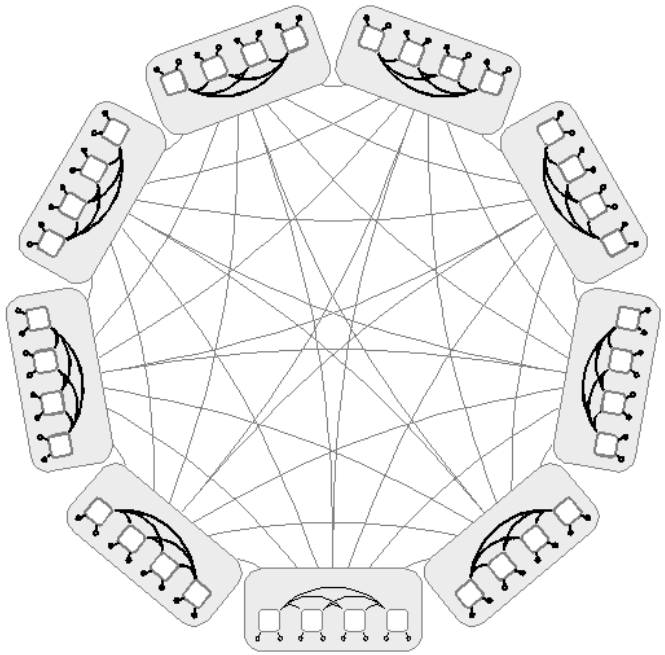


Toro, Cray Jaguar (2009)

Topologías

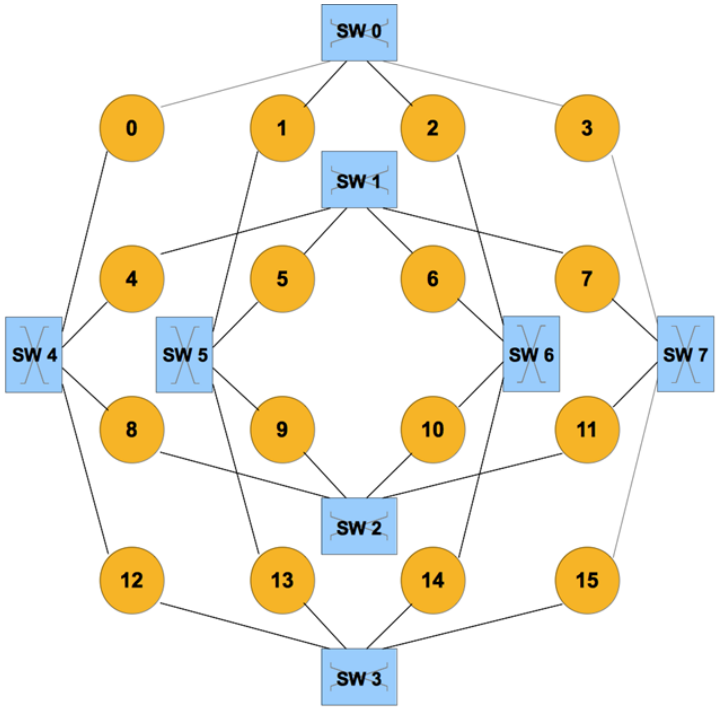


Fat Tree, Tianhe-2 (2017)

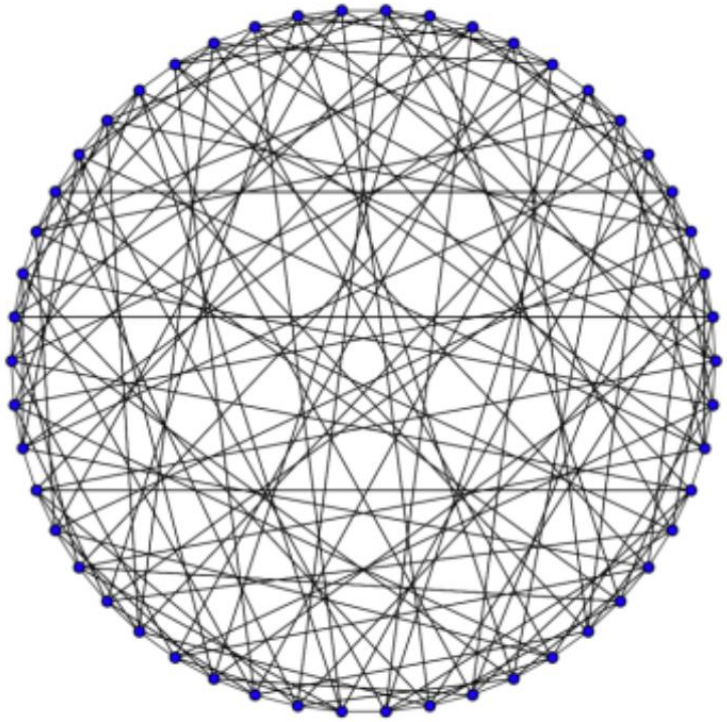


Dragonfly, Piz Daint (2017)

Topologías



KNS, Duato (2014)



Slim Fly, Hoefler (2012)

Otros temas

- Técnicas de conmutación
- Control de flujo
- Encaminamiento
  - Adaptatividad???
- Congestión

Outline

- Introduction
- The context
- Congestion basics
- Should we care about congestion in current and future interconnection networks?
- Solutions: How can congestion be managed?
- Challenges

The context

High-Performance Interconnection Networks

- High-performance interconnection networks are **key elements** in **High-Performance Computing (HPC) systems** and **datacenters**
- Applications/users demand increasing computing power/storage capacity
- Currently: Tens of thousands of processing and/or storage nodes
- Hundreds of thousands or millions of nodes expected to meet future demands (e.g Exascale challenge)

The context

What does the Exascale challenge consist in?

	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

The context

Current TOP500 list

Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	National Supercomputing Center in Wuxi China	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRCPC	10,649,600	93,014.6	125,435.9	15,371
2	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NU	3,120,000	33,862	[illegible]	17,808
3	Swiss National Supercomputing Centre (CSCS) Switzerland	Piz E5 int Cr				
4	DOE/SC/Oak Ridge National Laboratory United States	Tit 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.				
5	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890

Sunway TaihuLight – 1st TOP500
93 PFLOPS (peak) / 15,3 MW
1 ExaFLOP = 164 MW

The context

Current Green500 list

Rank	TOP500 Rank	System	Cores	Rmax (TFlop/s)	Power (kW)	Power Efficiency (GFlops/watts)
1	61	**TSUBAME3.0** - SGI ICE XA, IP139-SXM2, Xeon E5-2680v4 14C 2.4GHz, Intel Omni-Path, NVIDIA Tesla P100 SXM2 , HPE GSIC Center, Tokyo Institute of Technology Japan	36,288	1,998.0	142	14.110
2	465	**kukai** - ZettaScaler-1.6 GPGPU system, Xeon E5-2650Lv4 14C 1.7GHz, Infiniband FDR, NVIDIA Tesla P100 , ExaScalar Yahoo Japan Corporation Japan	10,080	460.7	[illegible]	[illegible].046
3	148	**AIST AI Cloud** - NEC 4U-8GPU Server, Xeon E5-2630Lv4 10C 1.8GHz, Infiniband EDR, NVIDIA Tesla P100 SXM2 , NEC National Institute of Advanced Industrial Science and Technology Japan	23,4[illegible]	[illegible]	[illegible]	[illegible]
4	305	**RAIDEN GPU subsystem** - NVIDIA DGX-1, Xeon E5-2698v4 20C 2.2GHz, Infiniband EDR, NVIDIA Tesla P100 , Fujitsu Center for Advanced Intelligence Project, RIKEN Japan	11,712	635.1	60	10.603
5	100	**Wilkes-2** - Dell C4130, Xeon E5-2650v4 12C 2.2GHz, Infiniband EDR, NVIDIA Tesla P100 , Dell University of Cambridge United Kingdom	21,240	1,193.0	114	10.428

1 ExaFLOP = 70 MW

The context

How to achieve the 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 (many more processors required)**
  - **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

The context

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

The context

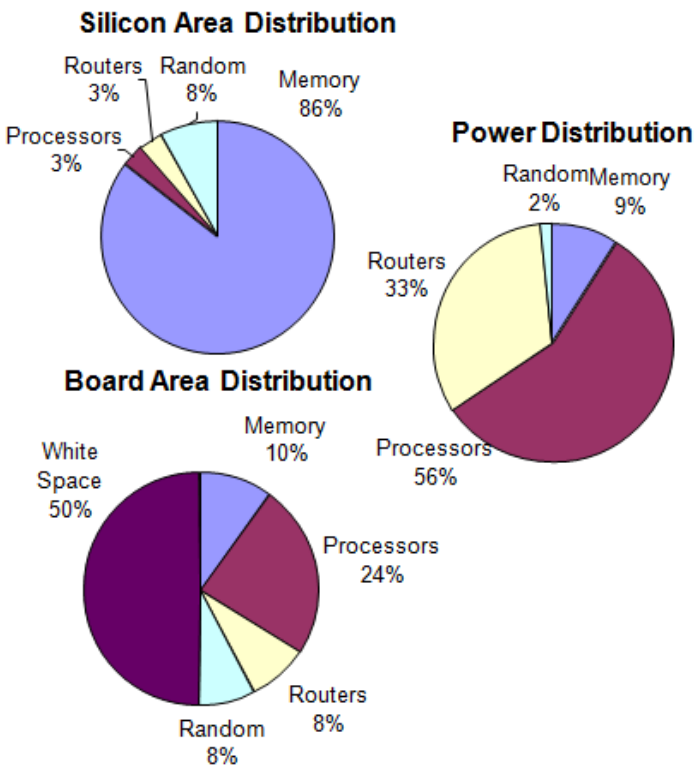
Interconnection Networks Performance

- Achieving **high network performance** is **mandatory** in current and future systems:
  - Very high **Throughput** (bytes delivered per time unit)
  - Very low **Latency** (time from packet generation to reception)
- Otherwise, the interconnection network may become the **system bottleneck**
- The network should not be overdimensioned due to **cost constraints** (Interconnects are no longer cheap)
- Hence, the network must be properly **designed** and **configured** to achieve **efficiently** the required performance

The context

Interconnection Networks Power Consumption

- **Power consumption fraction** of the interconnection network near 35% of total
- Most of the network power consumption is **devoted to the links**
- **Depending on the application**, the power consumption can be significantly affected



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

The context

Design 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

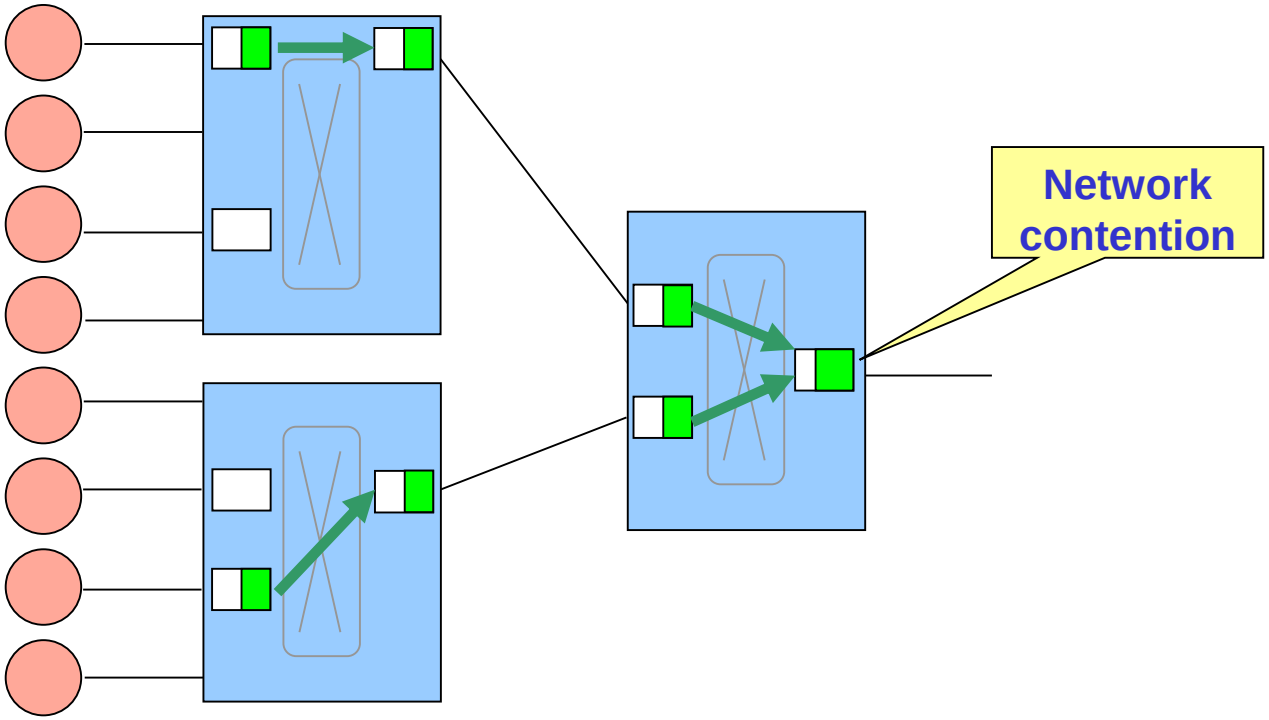
Outline

- Introduction
- The context
- Congestion basics
- Should we care about congestion in current and future interconnection networks?
- Solutions: How can congestion be managed?
- Challenges

Congestion basics

Contention

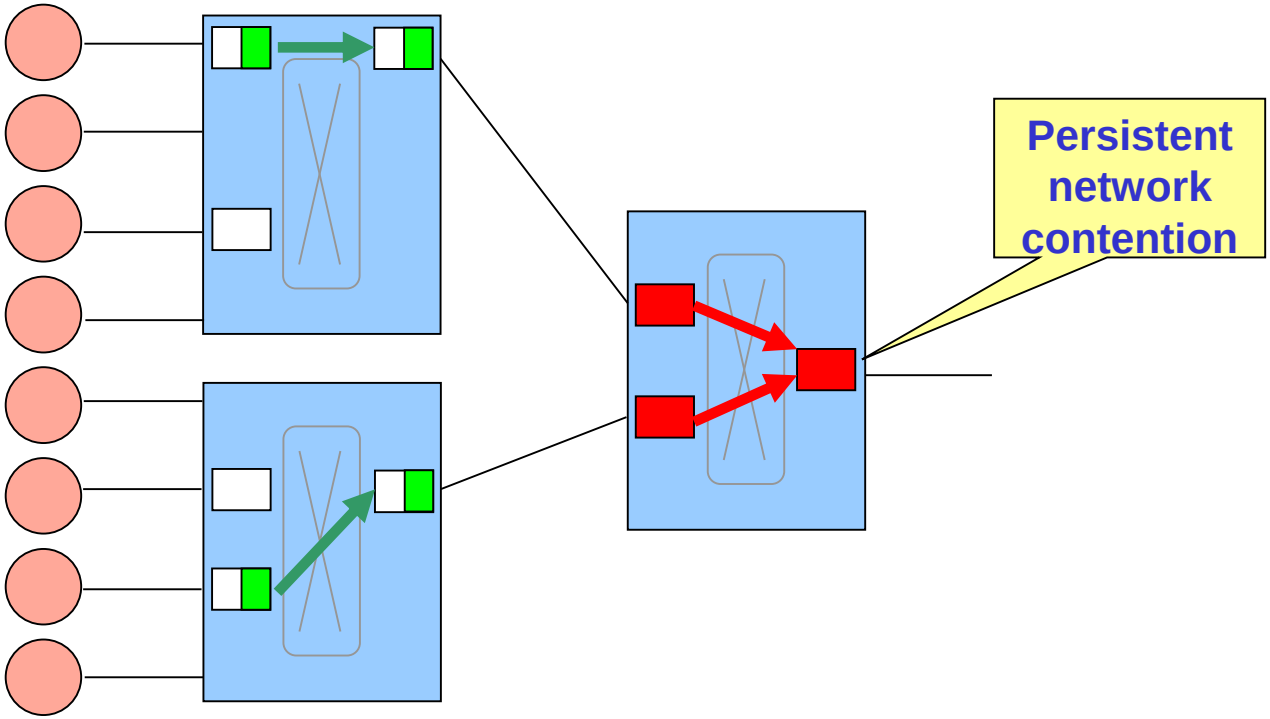
- Several packets from different flows request the same output port in a switch
- One packet makes progress, the others wait



Congestion basics

Congestion

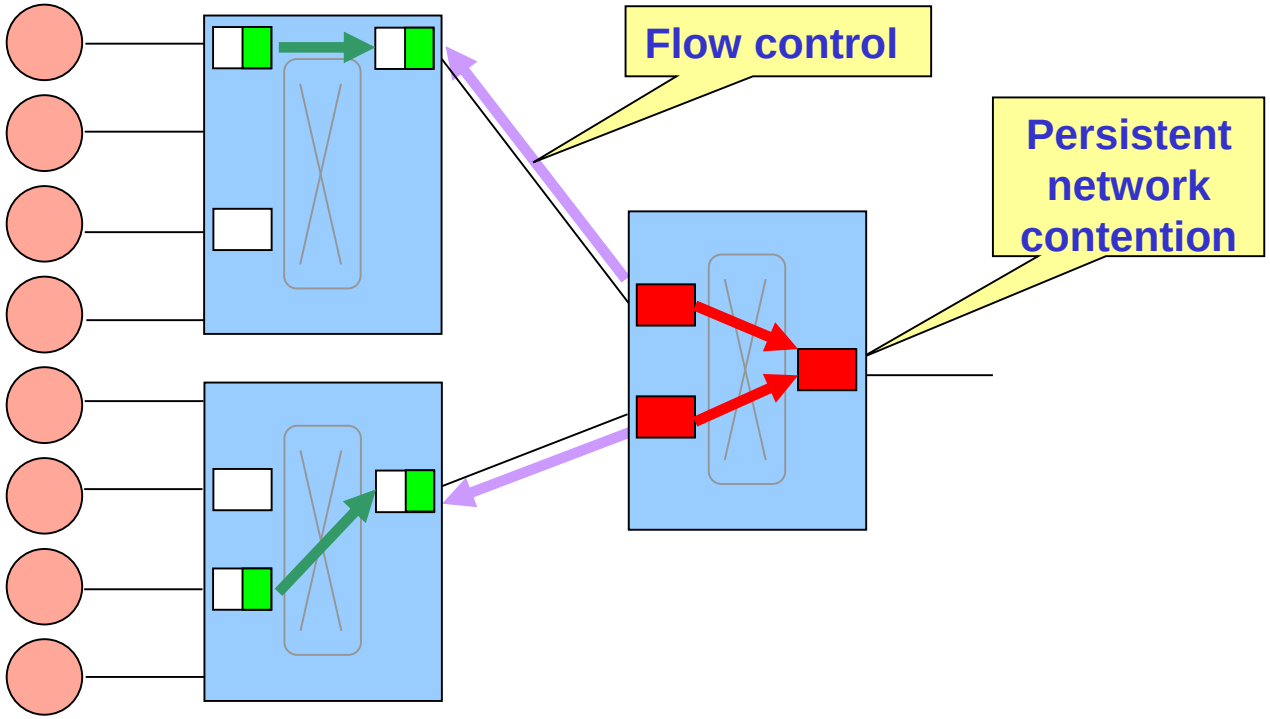
- Persistent contention, mainly in network saturation state
- Buffers containing packets belonging to flows involved in contention become full



Congestion basics

Congestion propagation

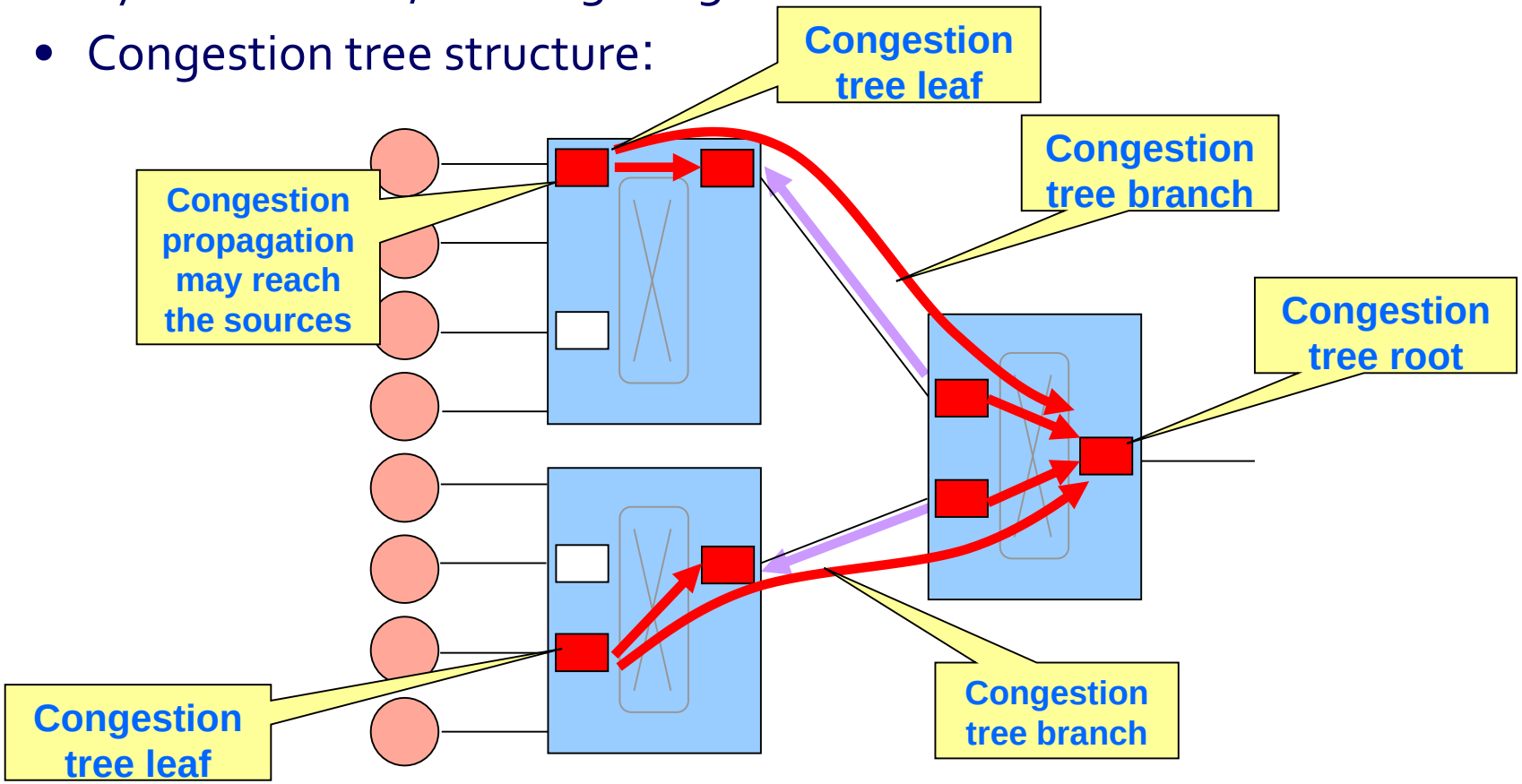
- In saturated lossless networks, congestion is quickly propagated by flow control, forming congestion trees



Congestion basics

Congestion propagation

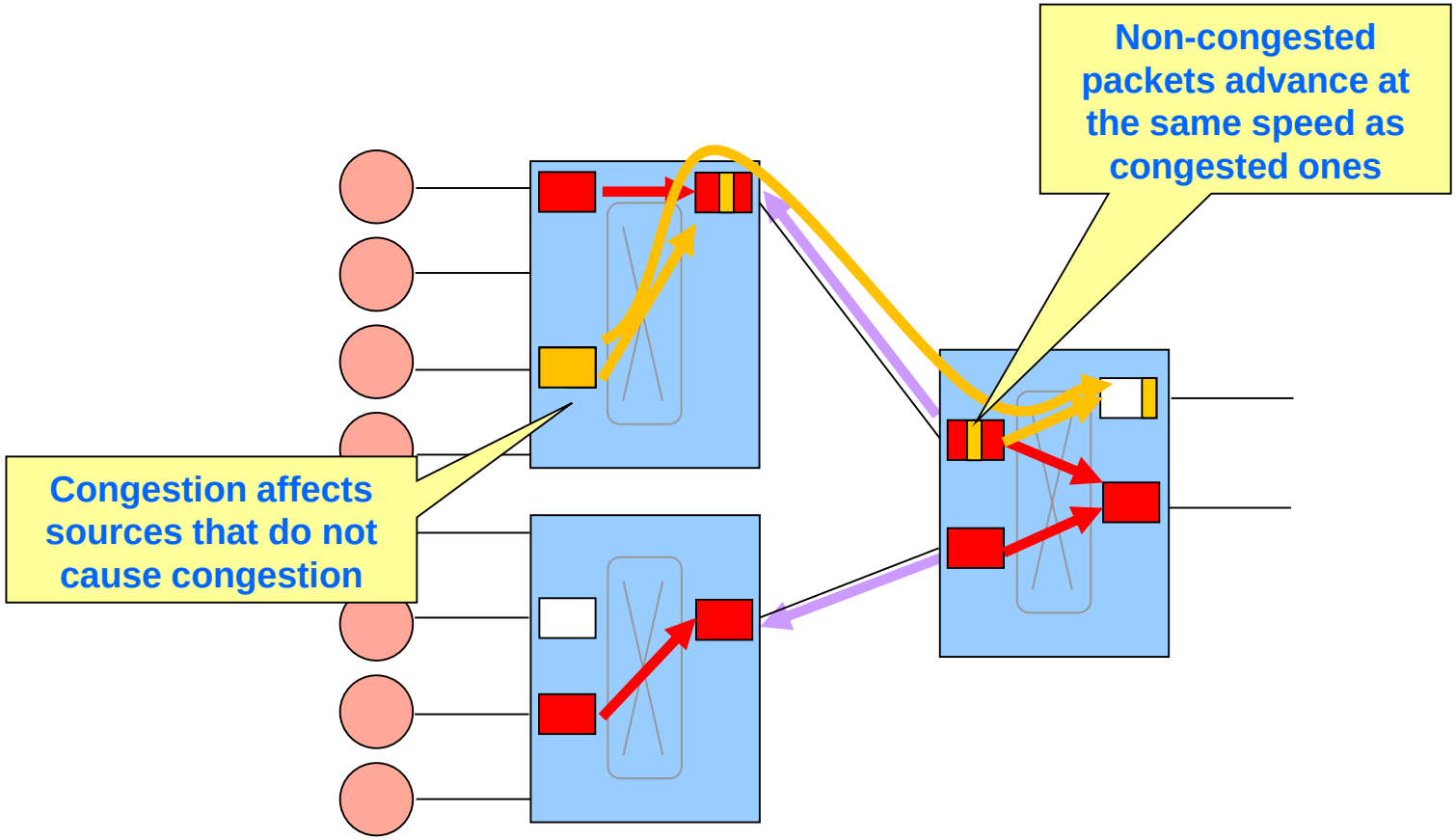
- In saturated lossless networks, congestion is quickly propagated by flow control, forming congestion trees
- Congestion tree structure:



Congestion basics

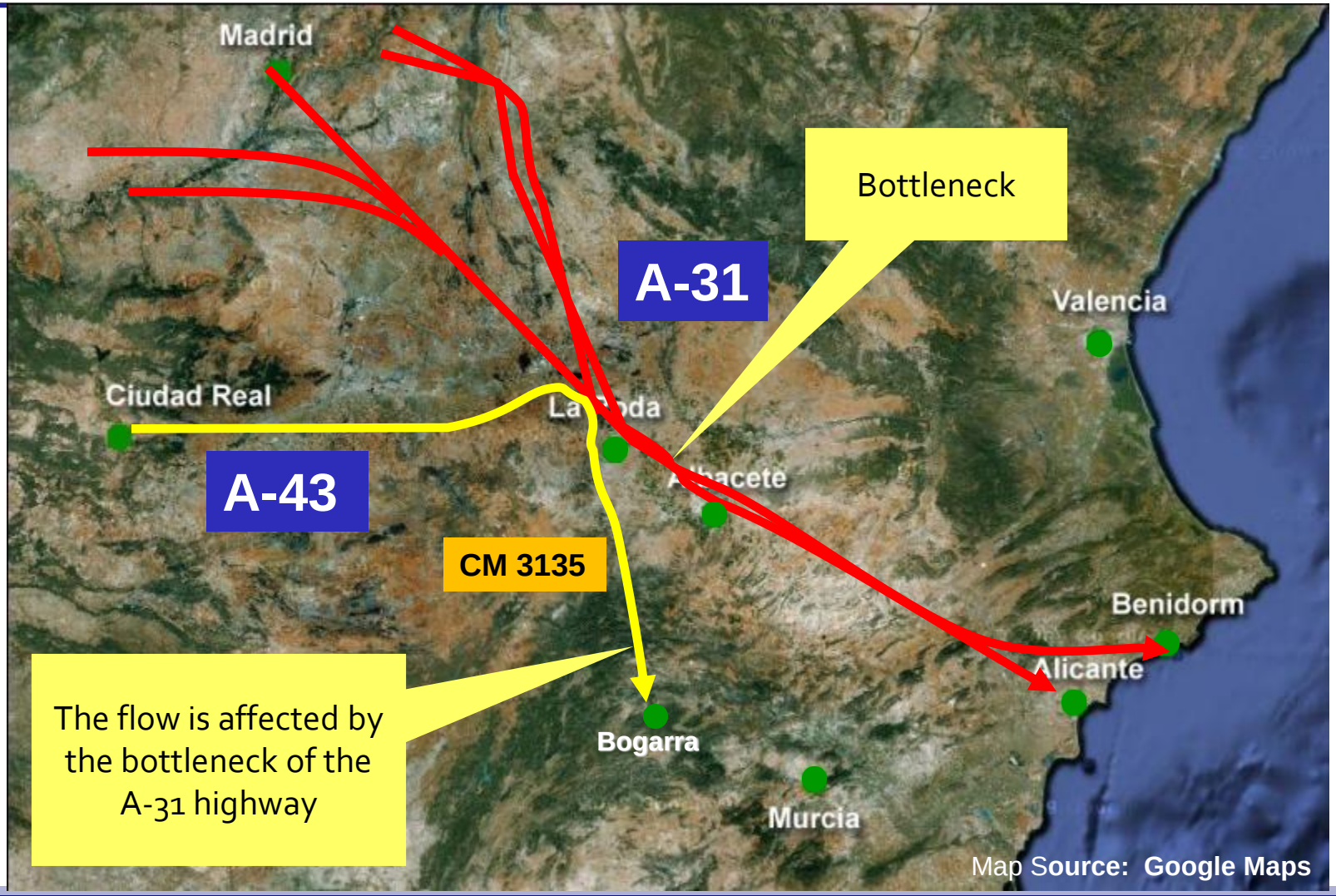
Congestion trees and Head-of-Line blocking

- Congestion trees may cause Head-of-Line (HoL) blocking



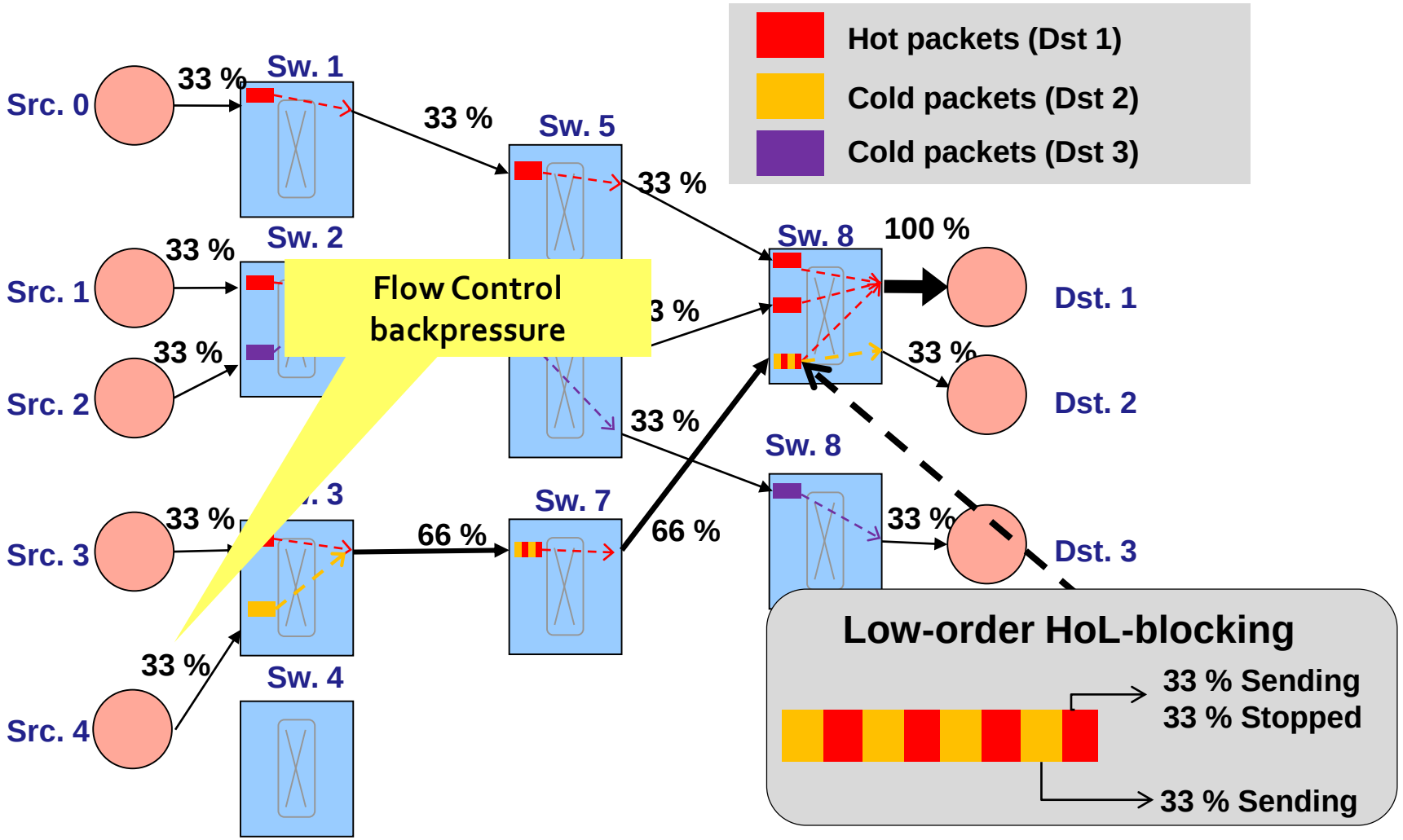
Congestion basics

Example of real-life HoL-blocking: The A-31 highway metaphor



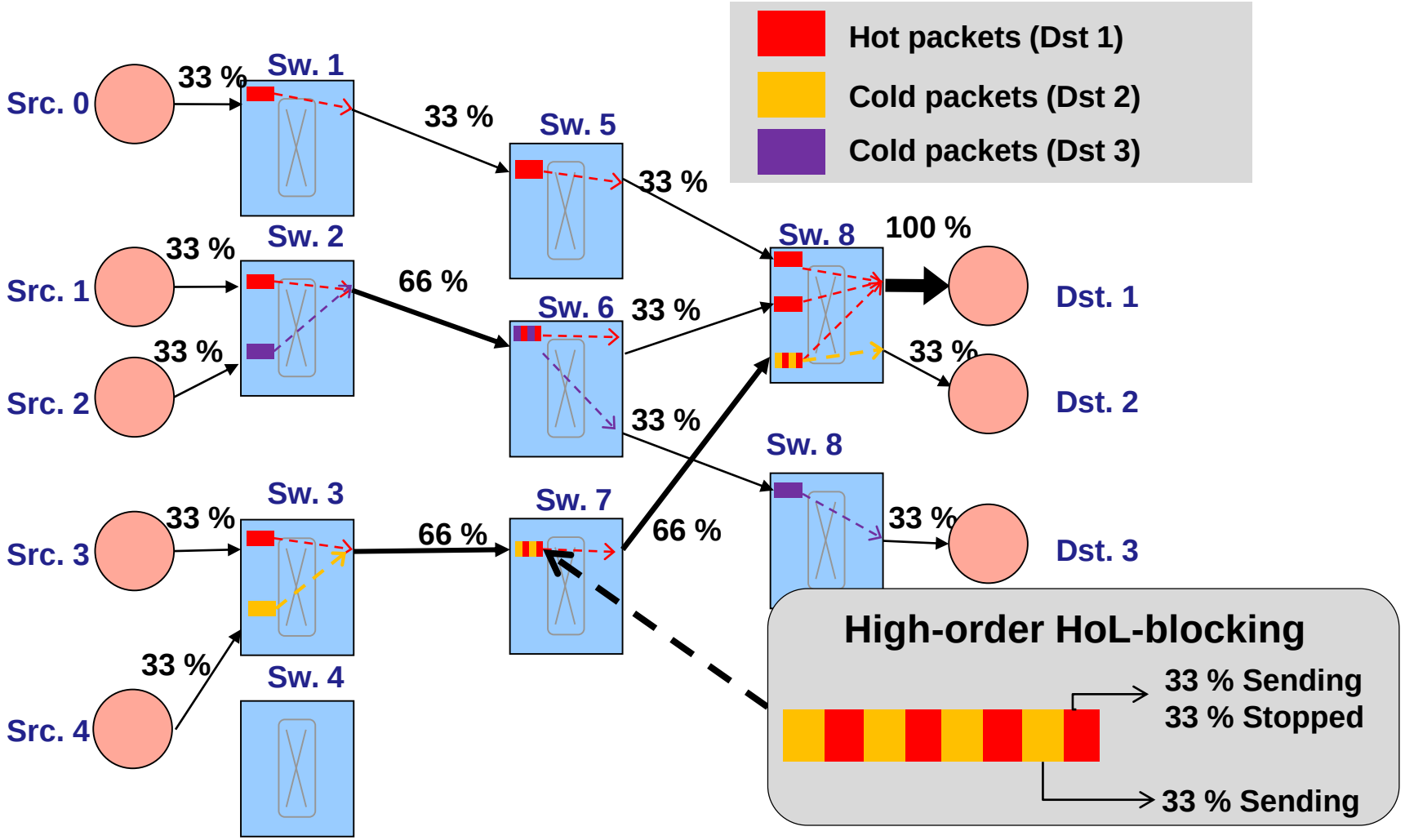
Congestion basics

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



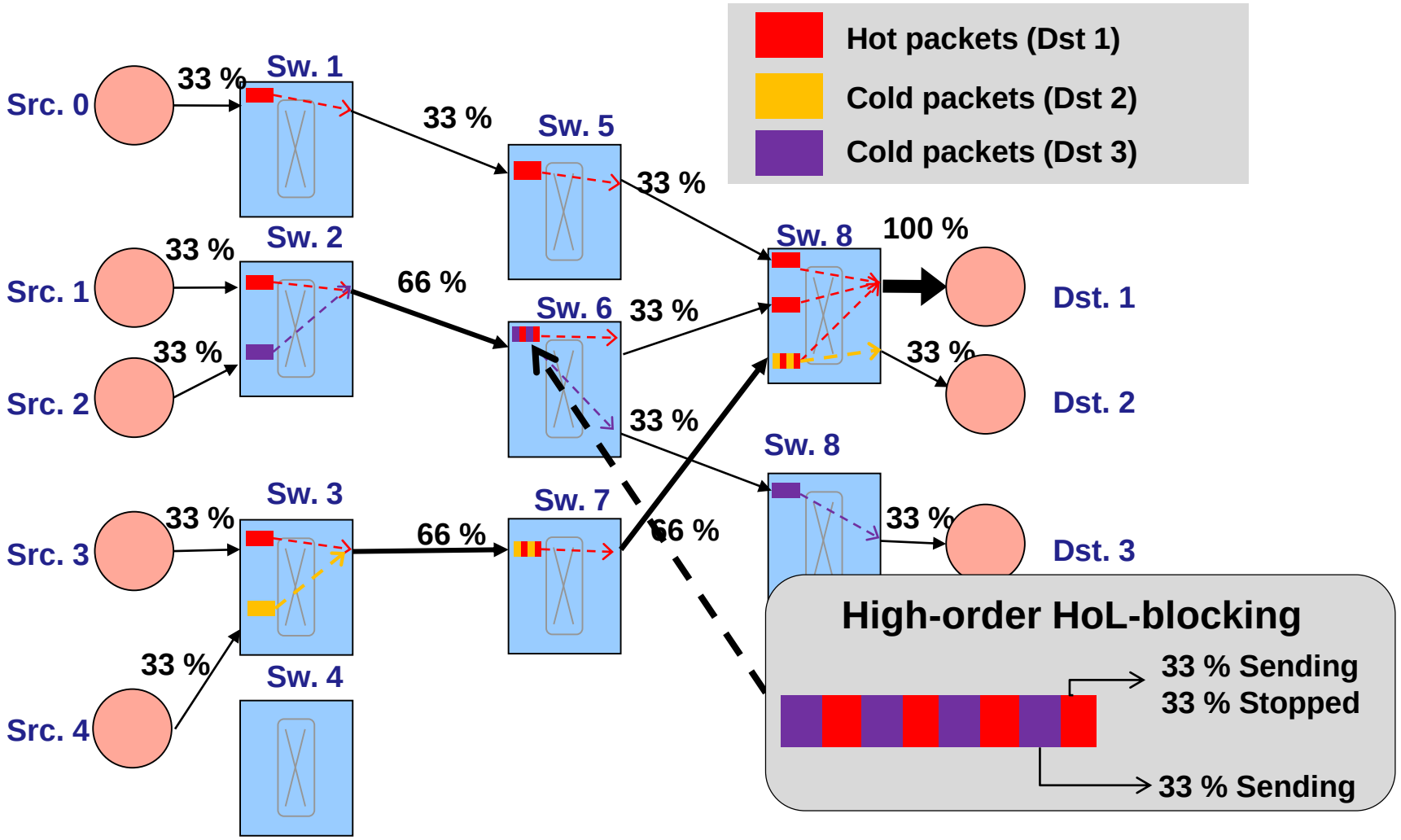
Congestion basics

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



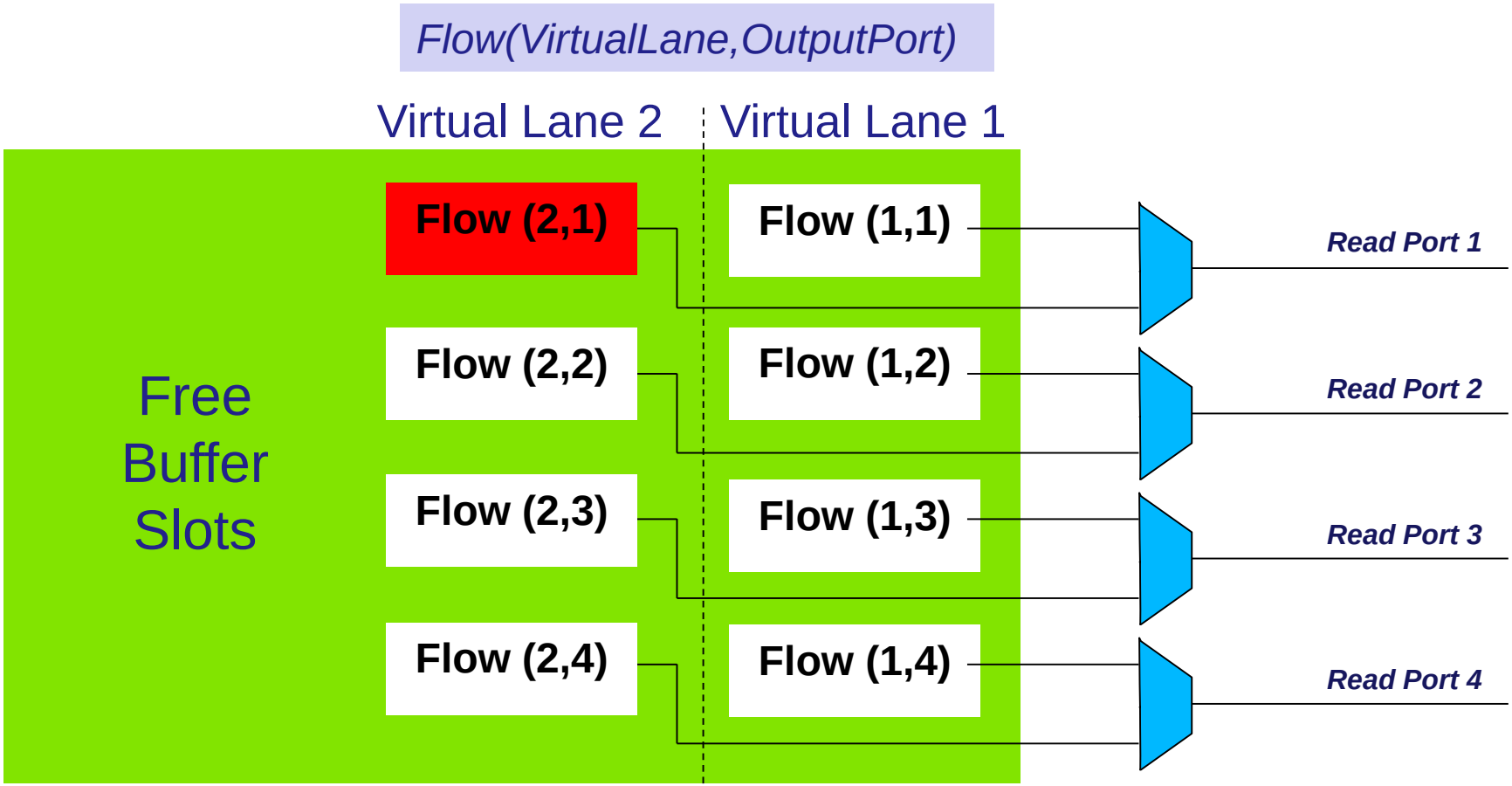
Congestion basics

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



Congestion basics

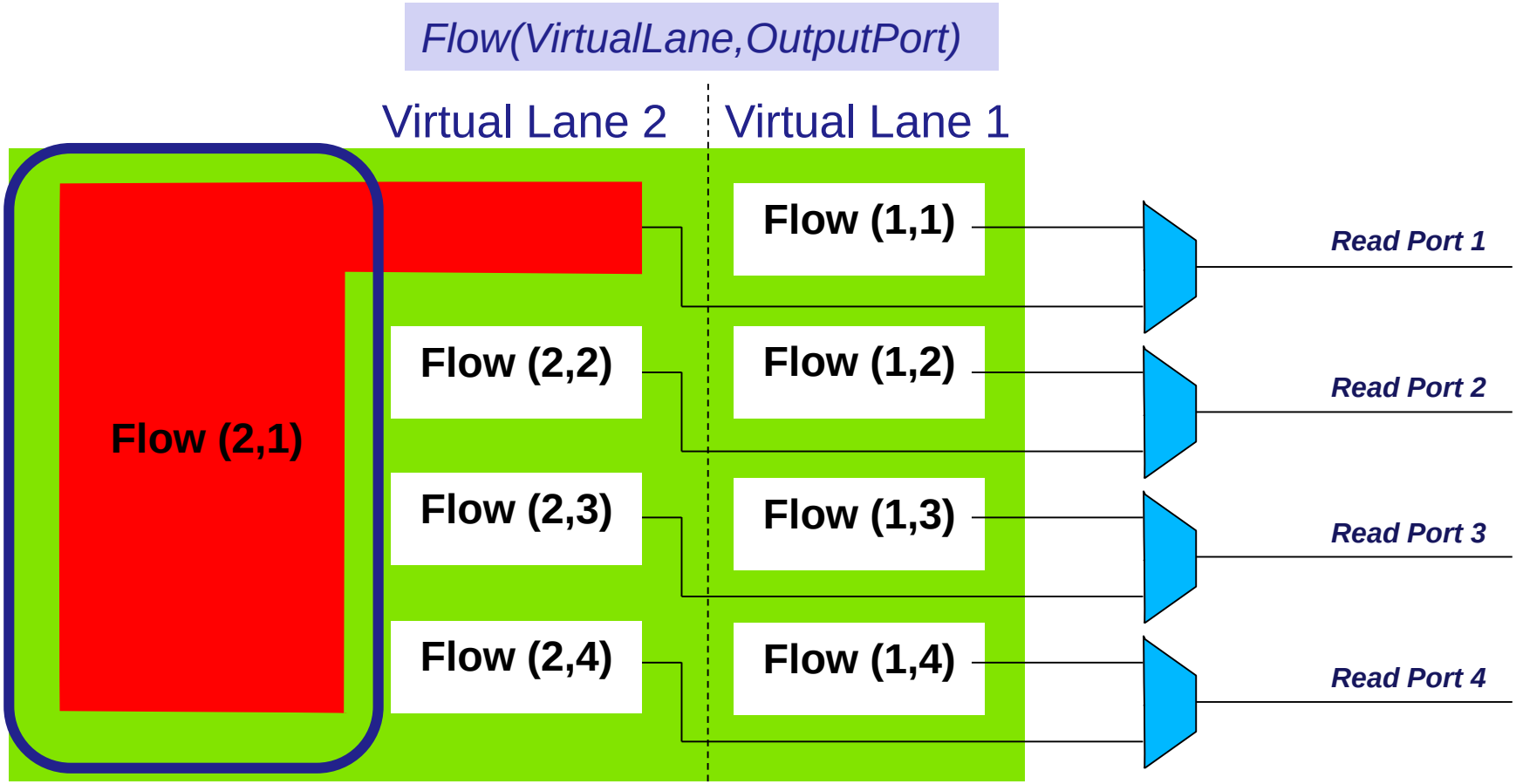
Buffer Hogging / Intra-VL hogging



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

Congestion basics

Buffer Hogging / Intra-VL hogging



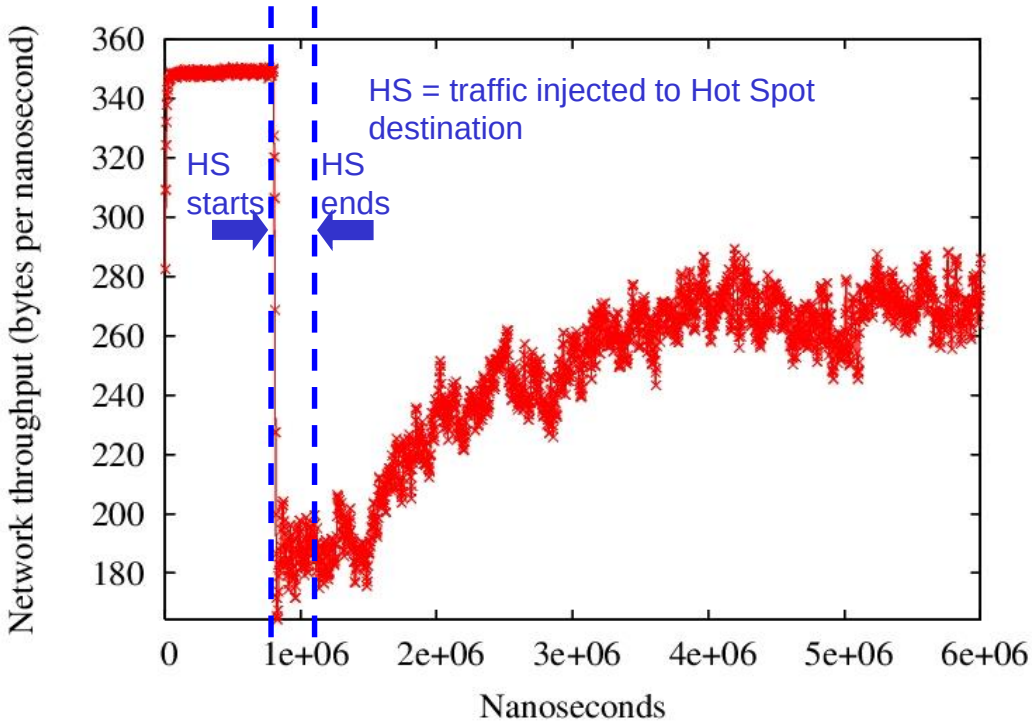
Kenji Yoshigoe: Threshold-based Exhaustive Round-Robin for the CICQ Switch with Virtual Crosspoint Queues. ICC 2007: 6325-6329

Outline

- Introduction
- The context
- Congestion basics
- Should we care about congestion in current and future interconnection networks?
- Solutions: How can congestion be managed?
- Challenges

Should we care about congestion?

Network performance at saturation

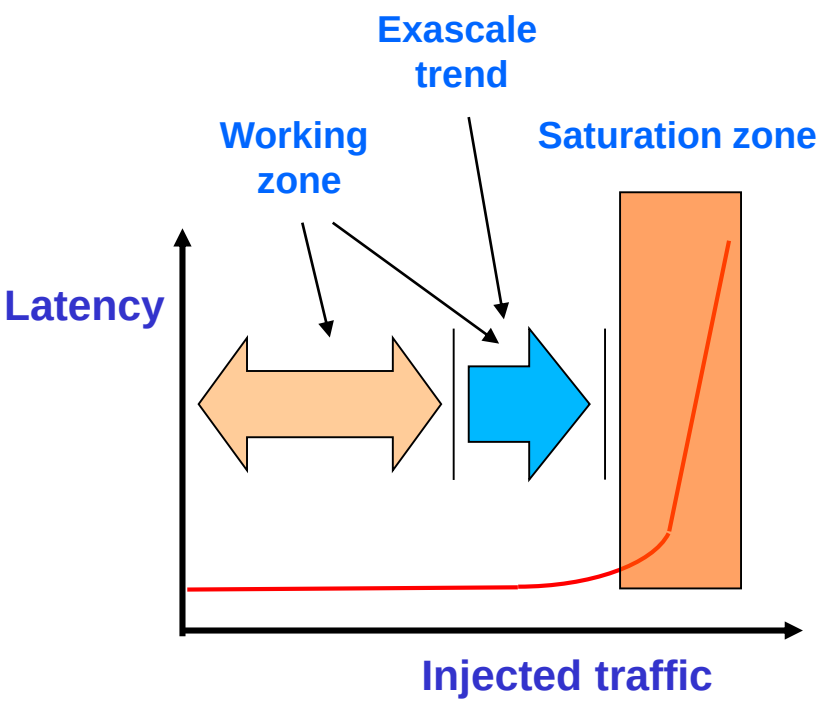


At saturation, **network performance** drops dramatically due to the cumulative effects of congestion situations

Should we care about congestion?

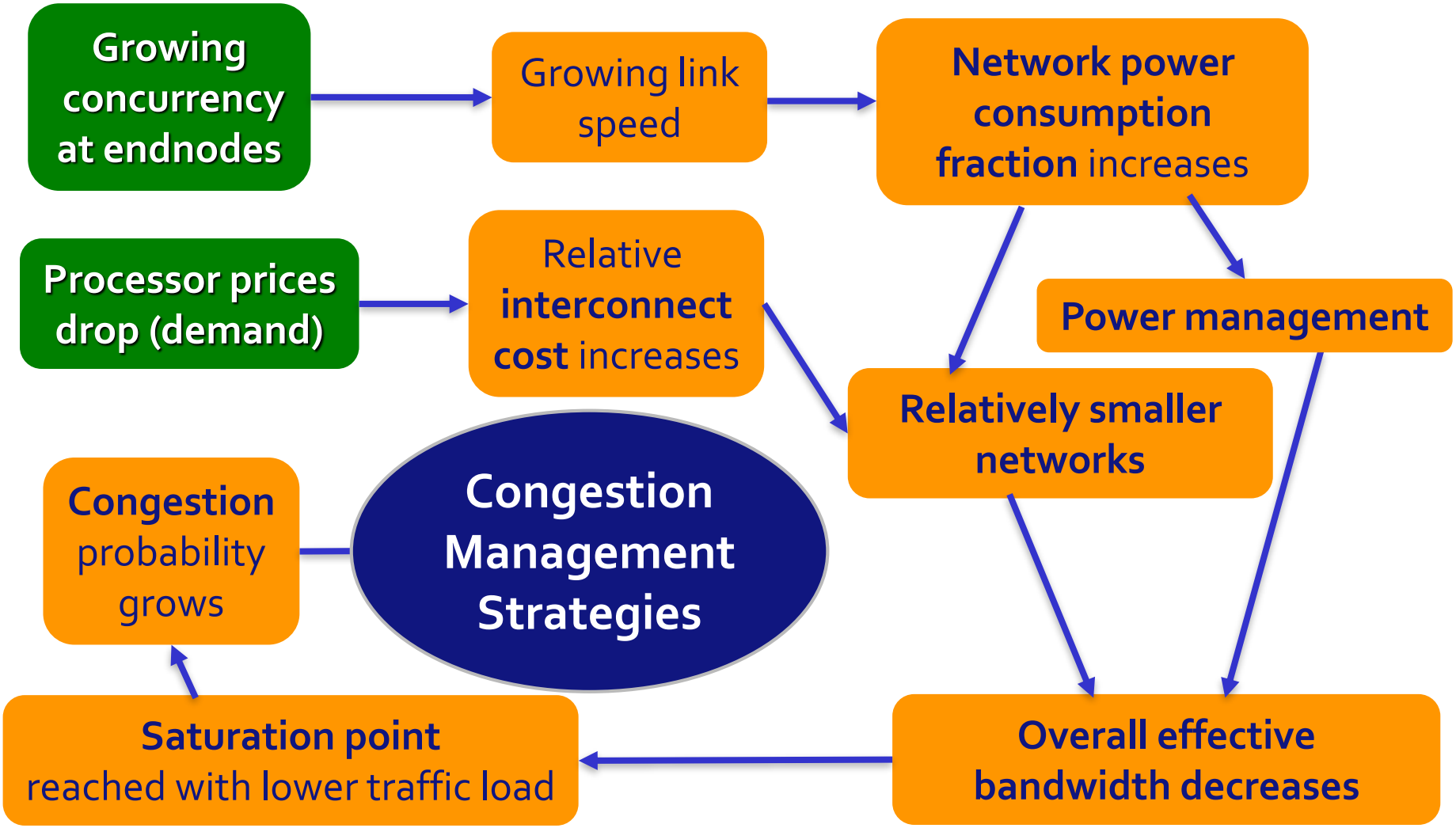
Is congestion likely to appear?

- 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



Should we care about congestion?

The big picture



Outline

- Introduction
- The context
- Congestion basics
- Should we care about congestion in current and future interconnection networks?
- Solutions: How can congestion be managed?
- Challenges

How can congestion be managed?

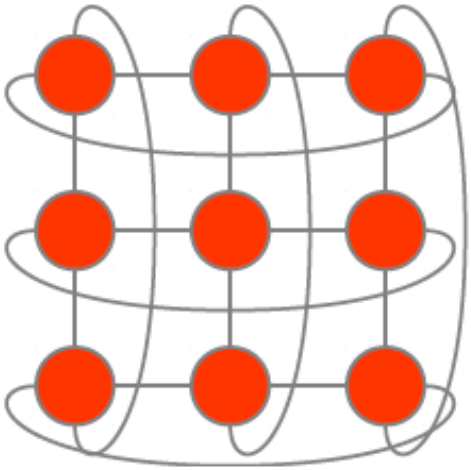
Different approaches

- Different ways to deal with congestion:
  - Appropriate topologies & routings
  - Packet dropping
  - Proactive techniques
  - Reactive techniques
  - HoL-blocking reduction techniques
  - HoL-blocking elimination techniques
  - Hybrid techniques

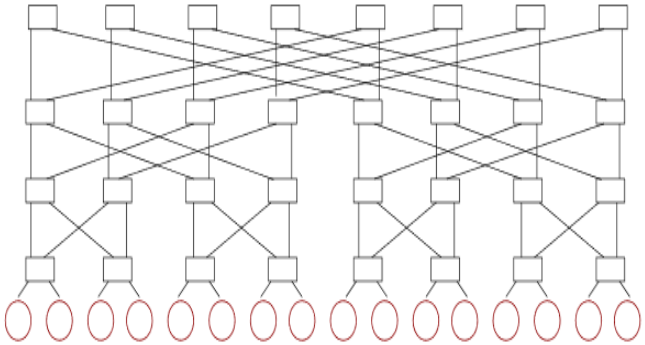
How can congestion be managed?

Appropriate topologies & routings

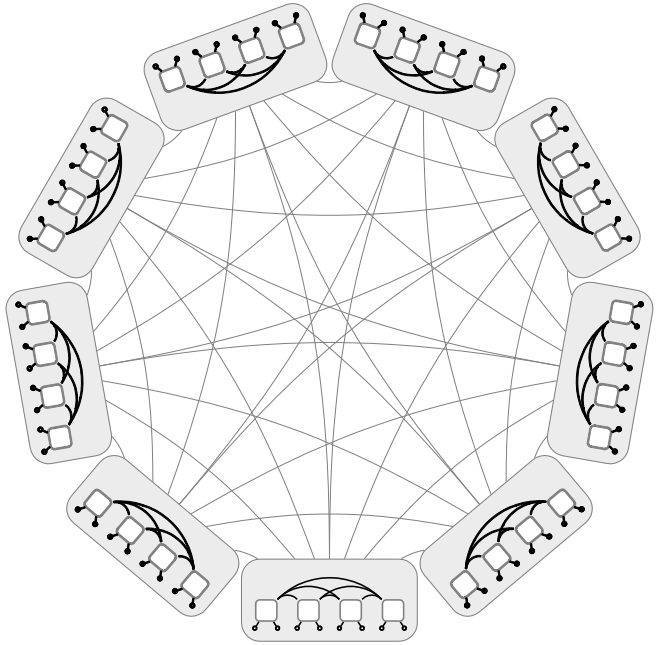
- **Efficient topologies** and their corresponding **routing algorithms** may **reduce congestion probability**
- The keypoint is achieving **traffic balance** throughout the network



Torus



Fat Tree



Dragonfly

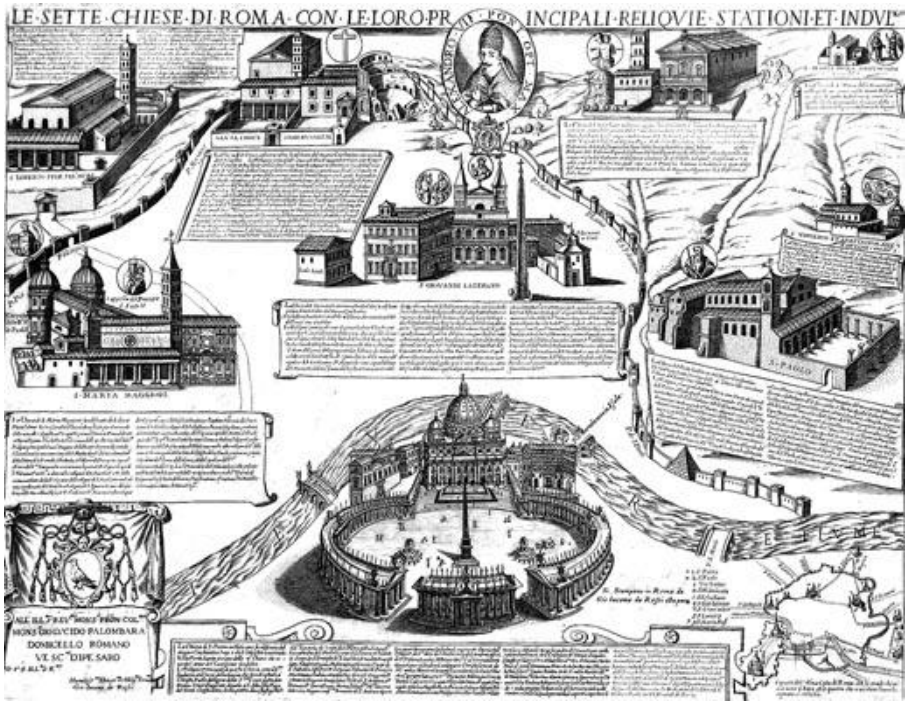
How can congestion be managed?

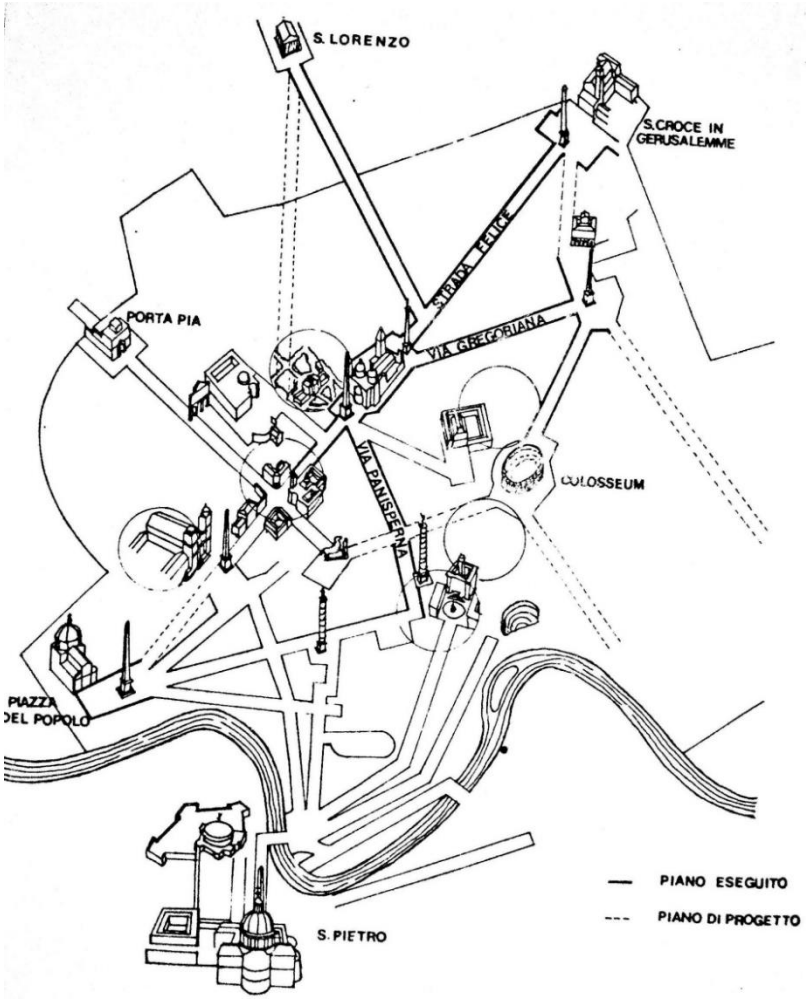
Historical notes: Sixtus V and the topology of Rome

- Many pilgrims (packets) in Rome visiting several churches (endnodes)
- By the end of the XVI century the network made of squares (switches) and streets (links) in Rome was chaotic-> **pilgrim jams**
- **Sixtus V** designed a **new topology** for Rome, trying **to balance pilgrim flows**

How can congestion be managed?

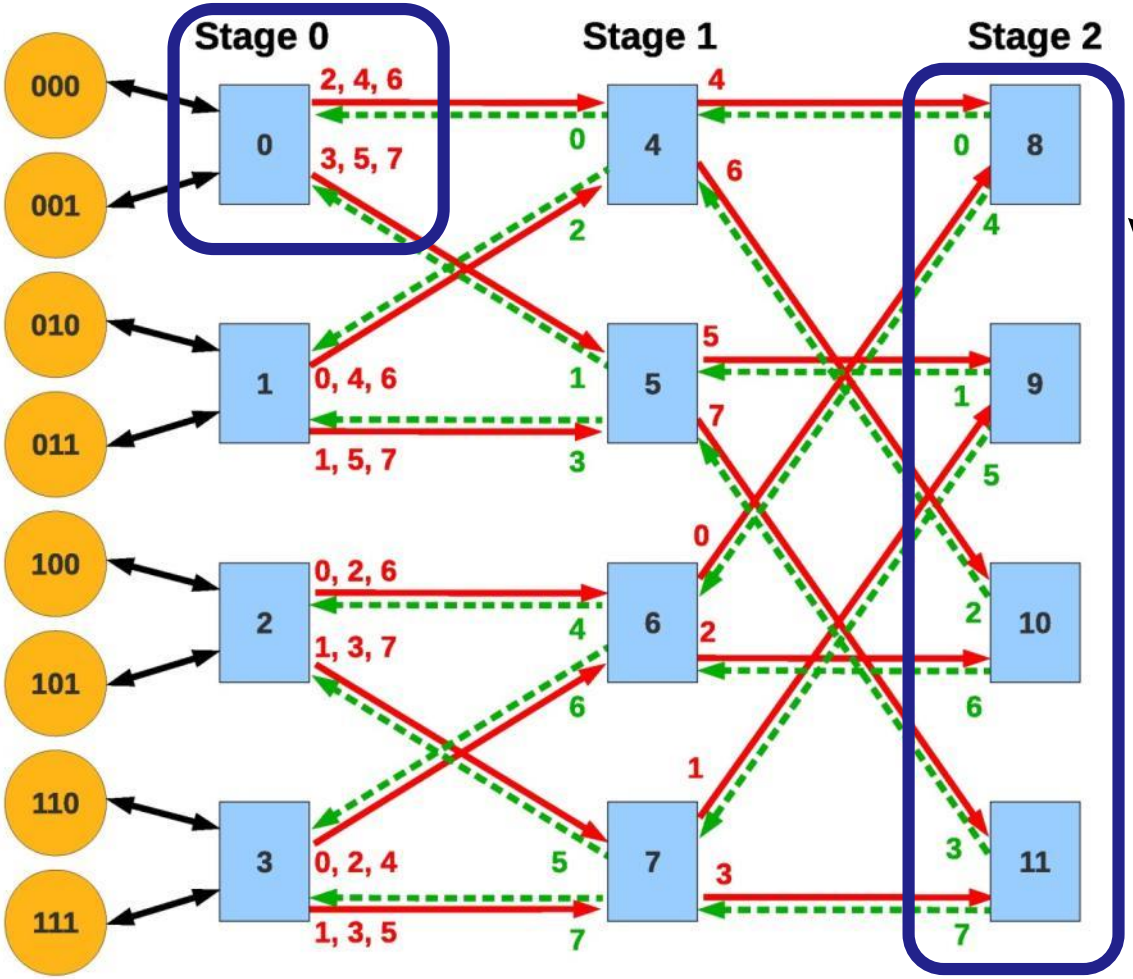
Historical notes: Sixtus V and the topology of Rome





How can congestion be managed?

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



Balances the use of links by different paths

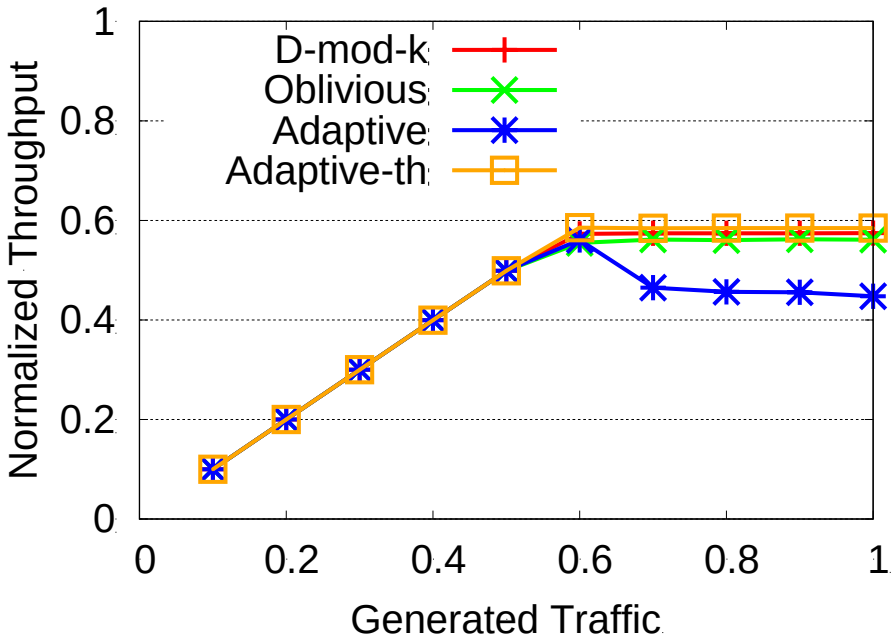
How can congestion be managed?

Deterministic, Oblivious and Adaptive Routing

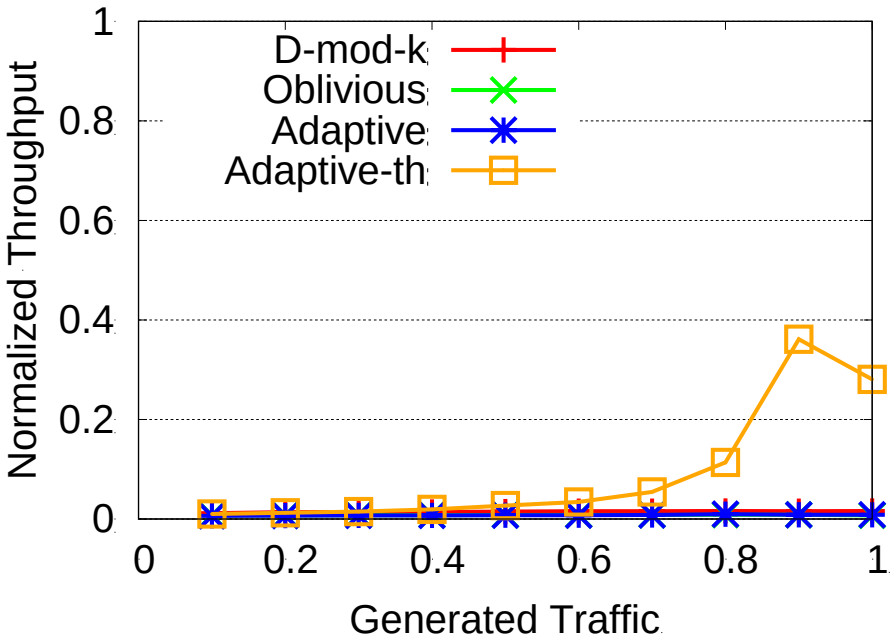
- In contrast with **deterministic** routing, **oblivious** and **adaptive** routings may use several paths between any source and destination:
  - Oblivious: routing **independent** of traffic status
  - Adaptive: routing decisions **based on** network conditions
- Adaptive and oblivious routings may help to deal with congestion, but:
  - Problems regarding in-order packet delivery
  - Congested points may vary
  - Moving paths may spread congestion over more links
  - Unable to solve satisfactorily heavy congestion situations

How can congestion be managed?

Deterministic, Oblivious and Adaptive Routing



11664-node real-life fat-tree, random traffic



11664-node real-life fat-tree, 10% hotspot

How can congestion be managed?

Appropriate topologies & routings

- Even the most efficient topologies and routings may end up suffering the congestion negative effects
- **Topologies and routings may help to delay the occurrence of congestion or to reduce congestion probability but cannot manage congestion by themselves**
- **Specific techniques to manage congestion are still required**

How can congestion be managed?

Different approaches

- Different ways to deal with congestion:
  - Appropriate topologies & routings
  - Packet dropping
  - Proactive techniques
  - Reactive techniques
  - HoL-blocking reduction techniques
  - HoL-blocking elimination techniques
  - Hybrid techniques

How can congestion be managed?

Packet dropping

- Packets in congested buffers are discarded
- Suitable for computer networks (like the Internet) but not suitable for most current HPC applications
- Both congested and non-congested packets may be discarded
- Discarded packets must be retransmitted, thus increasing final packet latency
- High latency variability
- Not suitable for HPC

How can congestion be managed?

Different approaches

- Different ways to deal with congestion:
  - Appropriate topologies & routings
  - Packet dropping
  - Proactive techniques
  - Reactive techniques
  - HoL-blocking reduction techniques
  - HoL-blocking elimination techniques
  - Hybrid techniques

How can congestion be managed?

Proactive congestion management

- Path setup before data transmission
- Used in ATM, computer networks (QoS)
- Optimal performance requires to know in advance:
  - Resource requirements of each transmission
  - Network status
- Knowledge about network status is not always available
- High overhead, high setup time, poor link utilization
- Too slow for HPC

P. Yew , N. Tzeng, D.H. Lawrie, "Distributing Hot-Spot Addressing in Large-Scale Multiprocessors", IEEE Transactions on Computers, 36(4): 388–395, 1987.

How can congestion be managed?

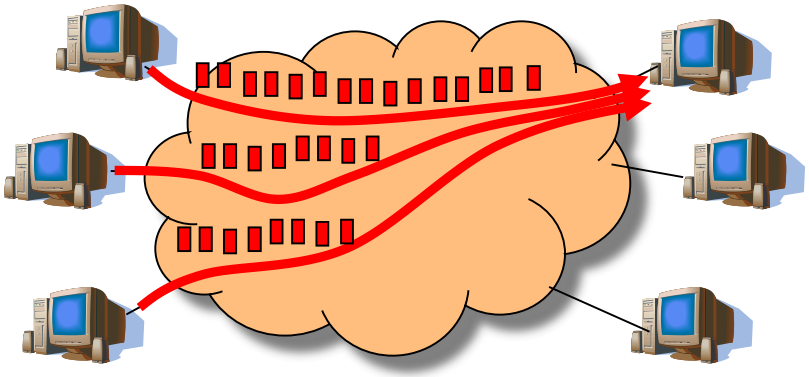
Different approaches

- Different ways to deal with congestion:
  - Appropriate topologies & routings
  - Packet dropping
  - Proactive techniques
  - Reactive techniques
  - HoL-blocking reduction techniques
  - HoL-blocking elimination techniques
  - Hybrid techniques

How can congestion be managed?

Reactive congestion management

- 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)



How can congestion be managed?

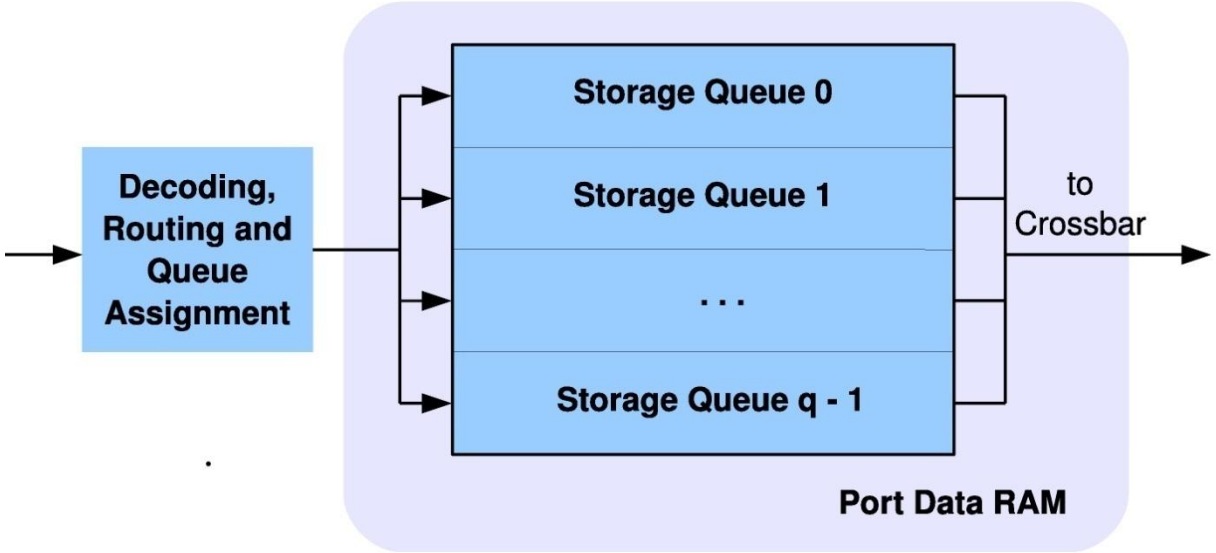
Different approaches

- Different ways to deal with congestion:
  - Appropriate topologies & routings
  - Packet dropping
  - Proactive techniques
  - Reactive techniques
  - HoL-blocking reduction techniques
  - HoL-blocking elimination techniques
  - Hybrid techniques

How can congestion be managed?

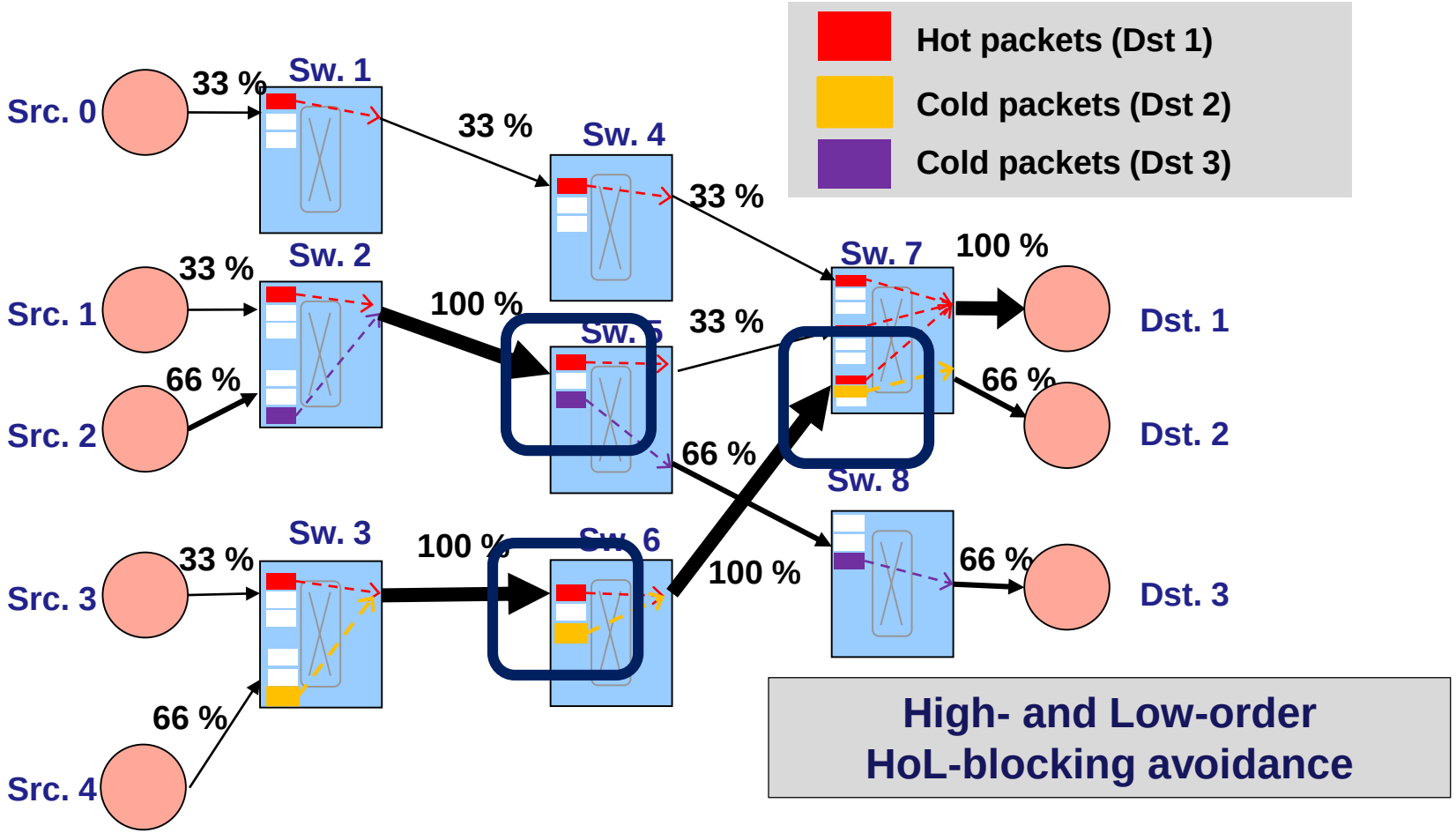
HoL-blocking reduction techniques

- 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



How can congestion be managed?

Static mapping of 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*

How can congestion be managed?

Generic Queuing Schemes

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

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

How can congestion be managed?

Generic Queuing Schemes

- **DBBM** (Destination-Based Buffer Management)
  - Several groups of destinations are defined
  - A separate queue for each group at every port (*q* queues per port)
  - Packets with destinations in the same group are stored at the same queue

Selected_Queue = Packet_Destination MOD q

  - Does not completely eliminate HoL-blocking
  - Effectiveness depends on the number of queues, topology and traffic pattern

T. Nachiondo, J. Flich, J. Duato, "Buffer management strategies to reduce HoL-blocking", IEEE Transactions on Parallel and Distributed Systems, vol. 21 (6), pp. 739–753, 2010.

How can congestion be managed?

Topology- & Routing –Aware Queuing Schemes

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

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

How can congestion be managed?

Different approaches

- Different ways to deal with congestion:
  - Appropriate topologies & routings
  - Packet dropping
  - Proactive techniques
  - Reactive techniques
  - HoL-blocking reduction techniques
  - HoL-blocking elimination techniques
  - Hybrid techniques

How can congestion be managed?

HoL-blocking elimination techniques

- Queue mapping schemes reduce HoL-blocking as much as possible with the available queues, **but do not eliminate it completely.**
- A complete effectiveness in solving these problems would require **allocating dynamically extra queues to set aside packets that generate 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, similar to D-mod-K)

How can congestion be managed?

Dynamic-Mapping Queuing Schemes: Basics

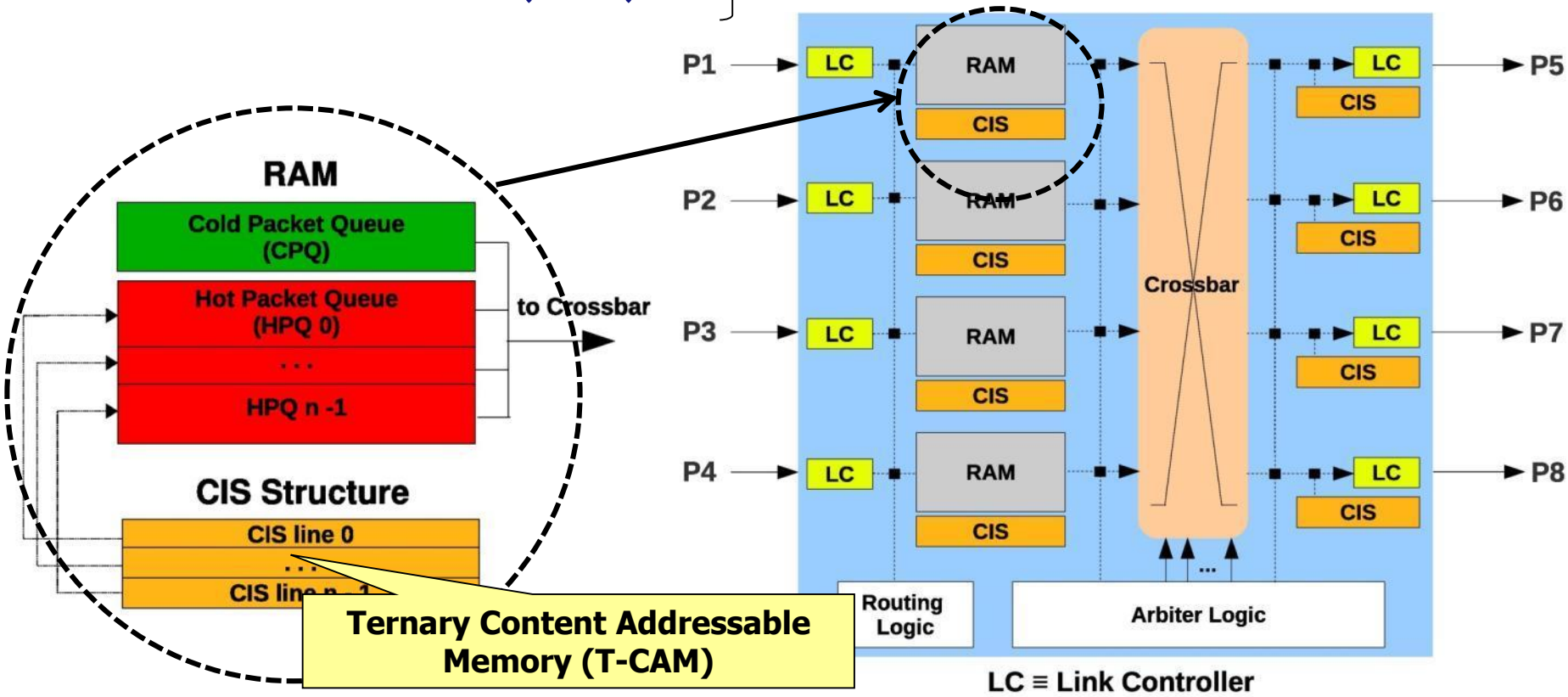
- **Congested points are detected** at any switch port of the network by measuring **queue occupancy**
- The **location** of any detected congested point is stored in a **control memory (a CAM or T-CAM line)** at any port forwarding packets towards the congested point
- A **special queue** associated 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**

How can congestion be managed?

Example of Dynamic-Mapping Queuing Scheme: DRBCM

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

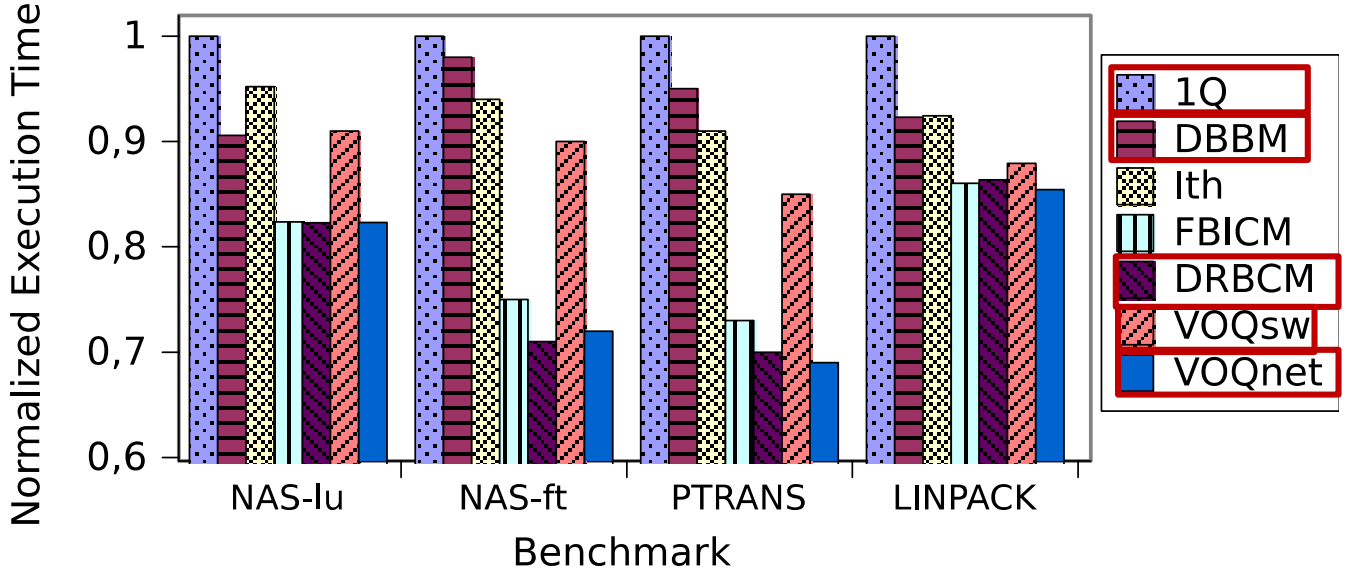
} To **Isolate** hot flows from cold ones



How can congestion be managed?

Example of Dynamic-Mapping Queuing 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.*

How can congestion be managed?

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

How can congestion be managed?

Different approaches

- Different ways to deal with congestion:
  - Appropriate topologies & routings
  - Packet dropping
  - Proactive techniques
  - Reactive techniques
  - HoL-blocking reduction techniques
  - HoL-blocking elimination techniques
  - Hybrid techniques

How can congestion be managed?

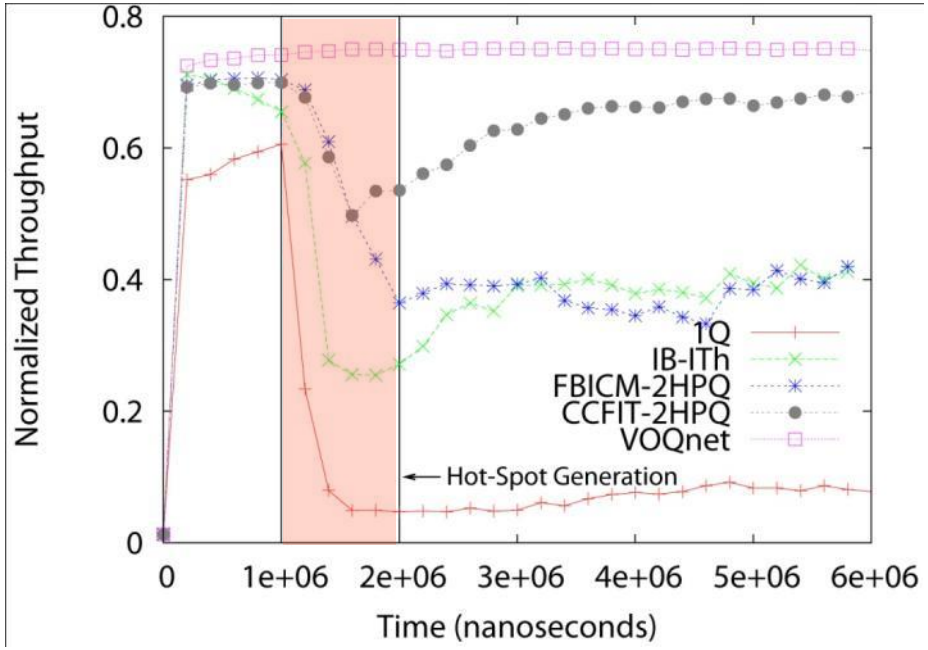
Hybrid Congestion Management Strategies

- **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)

How can congestion be managed?

Example of Hybrid Congestion Management: CCFIT

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



4-ary 4-tree
256 nodes

How can congestion be managed?

Summary

- Different ways to deal with congestion:
  - Appropriate topologies & routings: Not enough, but essential
  - Packet dropping: High and variable latency. Not suitable for HPC
  - Proactive techniques: High setup time. Too slow for HPC
  - Reactive techniques: Difficult to tune, lead to oscillations.
  - HoL-blocking reduction techniques: Scalable, excellent performance/complexity ratio
  - HoL-blocking elimination techniques: Highest effectiveness, complex to implement
  - Hybrid techniques: Pros and cons of the combined approaches

Outline

- Introduction
- The context
- Congestion basics
- Should we care about congestion in current and future interconnection networks?
- Solutions: How can congestion be managed?
- Challenges

Challenges

- To develop congestion management techniques that react ***locally*** and ***immediately*** when congestion arises
- To make congestion management techniques truly ***scalable***
- To achieve **coordination among end nodes** without explicit communication among them
- To **eliminate** instabilities and **oscillatory** responses
- To **minimize the number of extra resources** needed to handle congestion
- To make congestion management compatible with **adaptive routing**

Challenges

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

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS

UCLM UNIVERSIDAD DE CASTILLA-LA MANCHA

Thanks!!
Any question?

Francisco José Quiles Flor
Universidad de Castilla-La Mancha
SPAIN
Francisco.Quiles@uclm.es

Style Powered by: ACDA