Strategies to drastically improve congestion control in high performance data centers: next steps for RDMA

Jesus Escudero Sahuquillo (UCLM), Pedro Javier García (UCLM), Francisco J. Alfaro (UCLM), Francisco J. Quiles (UCLM) and Jose Duato (UPV)

E-mail: jesus.escudero@uclm.es

UCLM: University of Castilla-La Mancha, Spain

UPV: Universitat Politècnica de València, Spain

DCN: 1-19-0052-02-ICne

Executive summary

Congestion control strategies are required to reduce the negative effects of congestion, such as Head-of-Line (HoL) blocking. Explicit Congestion Notification (ECN) and some solutions based on it, are the most popular congestion management solutions. Unfortunately, the ECN closed-loop mechanism is not able to react in a smooth way under most congestion scenarios. In this talk, we describe the effects that lead the ECN closed-loop mechanism to generate oscillations when congestion is notified to the NICs. We also propose several ideas to improve ECN based on some research we have conducted in the past years.

Agenda

- Introduction
- Current congestion control
- Consequences
- How can we improve it?
- Conclusions

Introduction

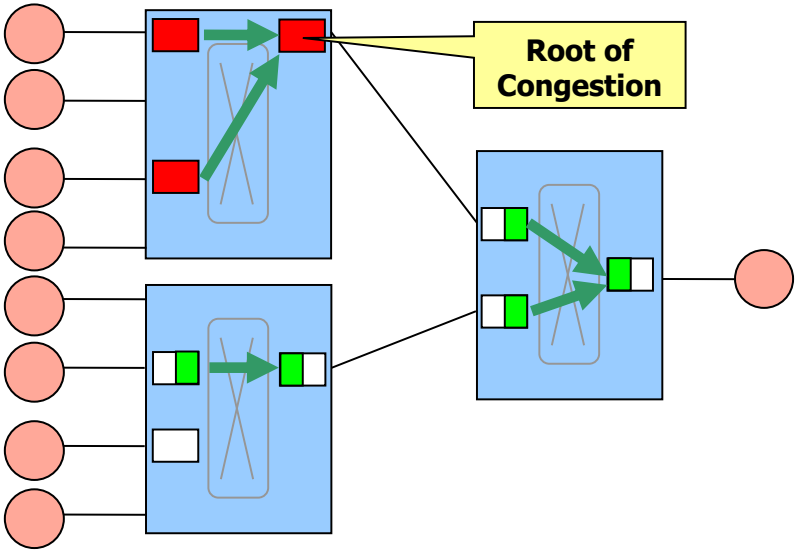
Context in Datacenter Networks (DCNs)

- **Datacenter Use Cases** (OLDI services, Deep Learning, NVMeoF and Cloudification [Congdon18]), require convergent network.
- **RDMA** for high-throughput and low latency communications.
- **Large DCNs** (thousands of server nodes):
  - Topology properties (path diversity and reduced diameter).
  - Efficient routing algorithms (load balancing).
  - Congestion threatens efficient topologies and routings.
- Lossless or low loss: Priority Flow Control + ECN.
- Even in these scenarios **congestion dramatically degrades network performance**, due to its negative effects: HOL blocking.

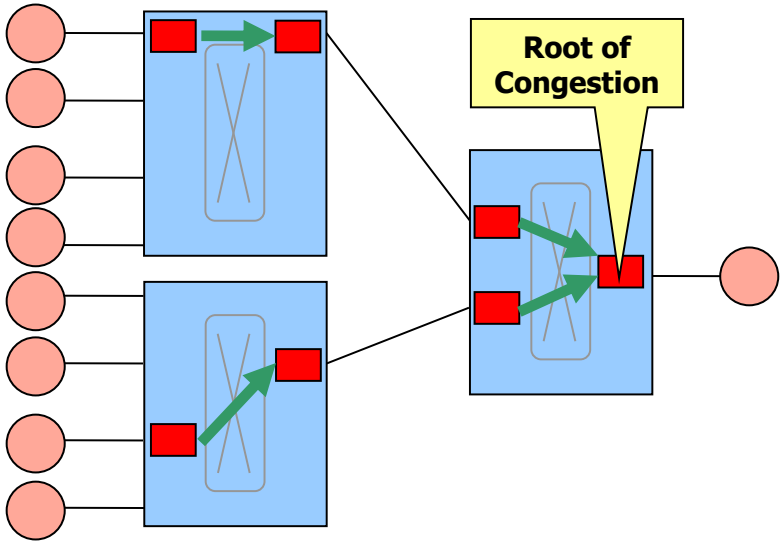
Introduction

Congestion tree dynamics [Garcia05][Garcia19]

- In general, the **switch where congestion originates** could be located at some initial or intermediate stage or be directly connected to end nodes.



In-network congestion

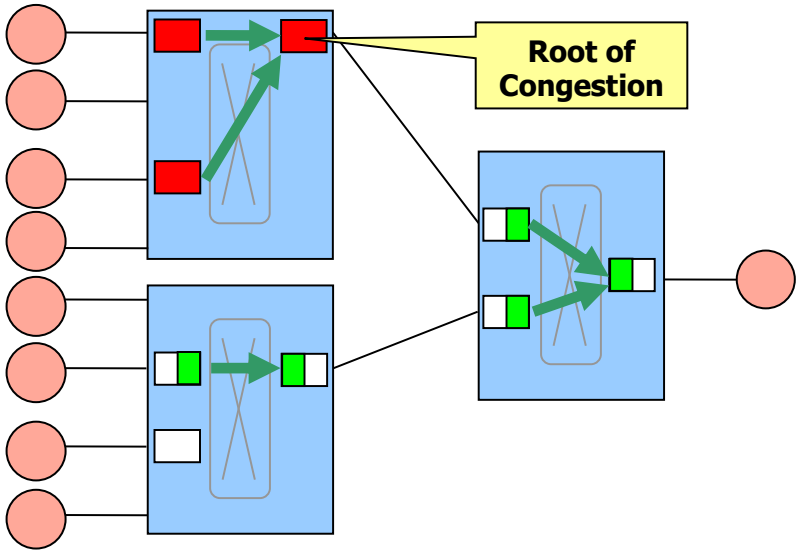


Incast congestion

Introduction

In-Network Congestion

- It usually occurs when **congestion is light** (i.e. it exceeds available link bandwidth by a small integer factor at most).
- There are two basic scenarios:
  1. A few nodes injecting traffic at full rate towards the same destination.
  2. Many nodes injecting traffic at low rates towards the same destination.
- Egress ports of in-network congested switches work at full capacity and may contend with other flows for downstream switches, eventually moving the root of congestion downwards.

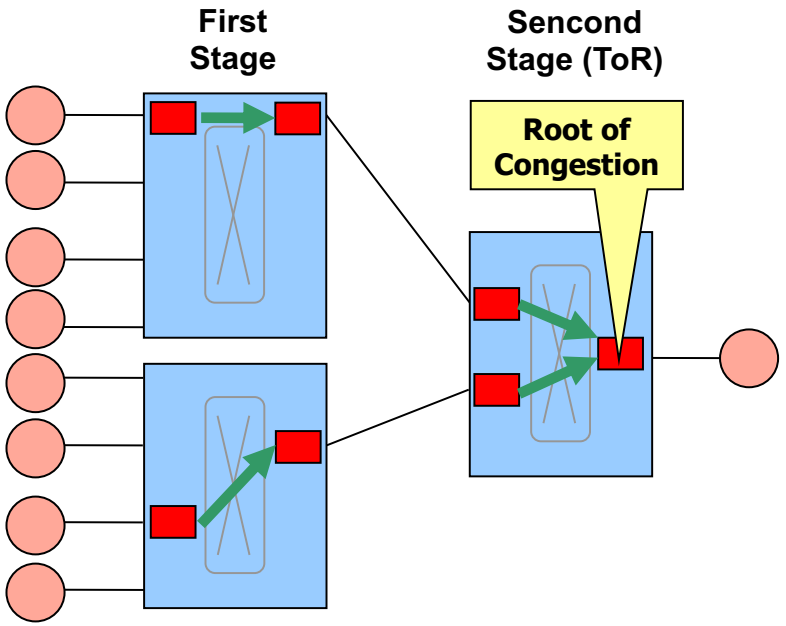


In-network congestion

Introduction

Incast Congestion

- Many nodes start to send packets at full rate towards the same destination, almost at the same time (e.g. OLDI services)
- **Incast congestion** occurs at the ToR switch where the node that multiple parties are synchronizing with is connected, and **grows from ToR switches to upstream switches.**
- Alternatively, in CLOS networks many small congestion trees concurrently appear at first-stage switches, later merging at second-stage switches and finally forming a larger congestion tree.



Incast congestion scenario

Introduction

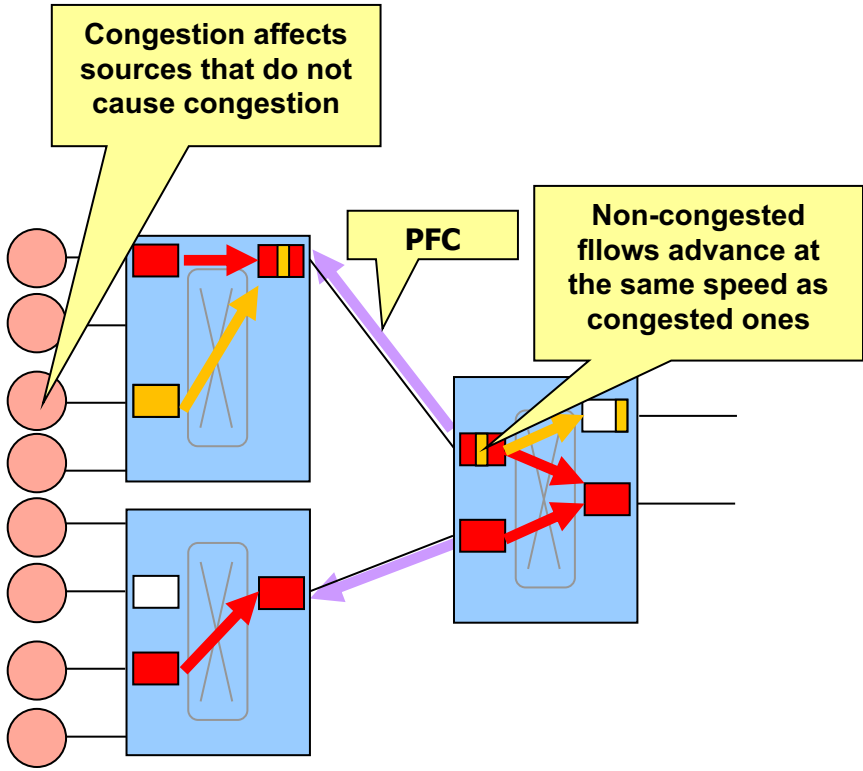
Traditional approaches

- Load balancing: **spread traffic flows across the multiple paths** in order to balance the load and hopefully avoid congestion (load balancing).
- Problems:
  1. Spreading traffic does not take into account whether the selected path is congested, generating **collisions of traffic flows in paths already congested**.
  2. The nature of flows matters: **elephant flows increase the chance of creating in-network congestion**.
  3. Traditional load balancing (e.g. ECMP) **does not work when incast congestion appears**.

Introduction

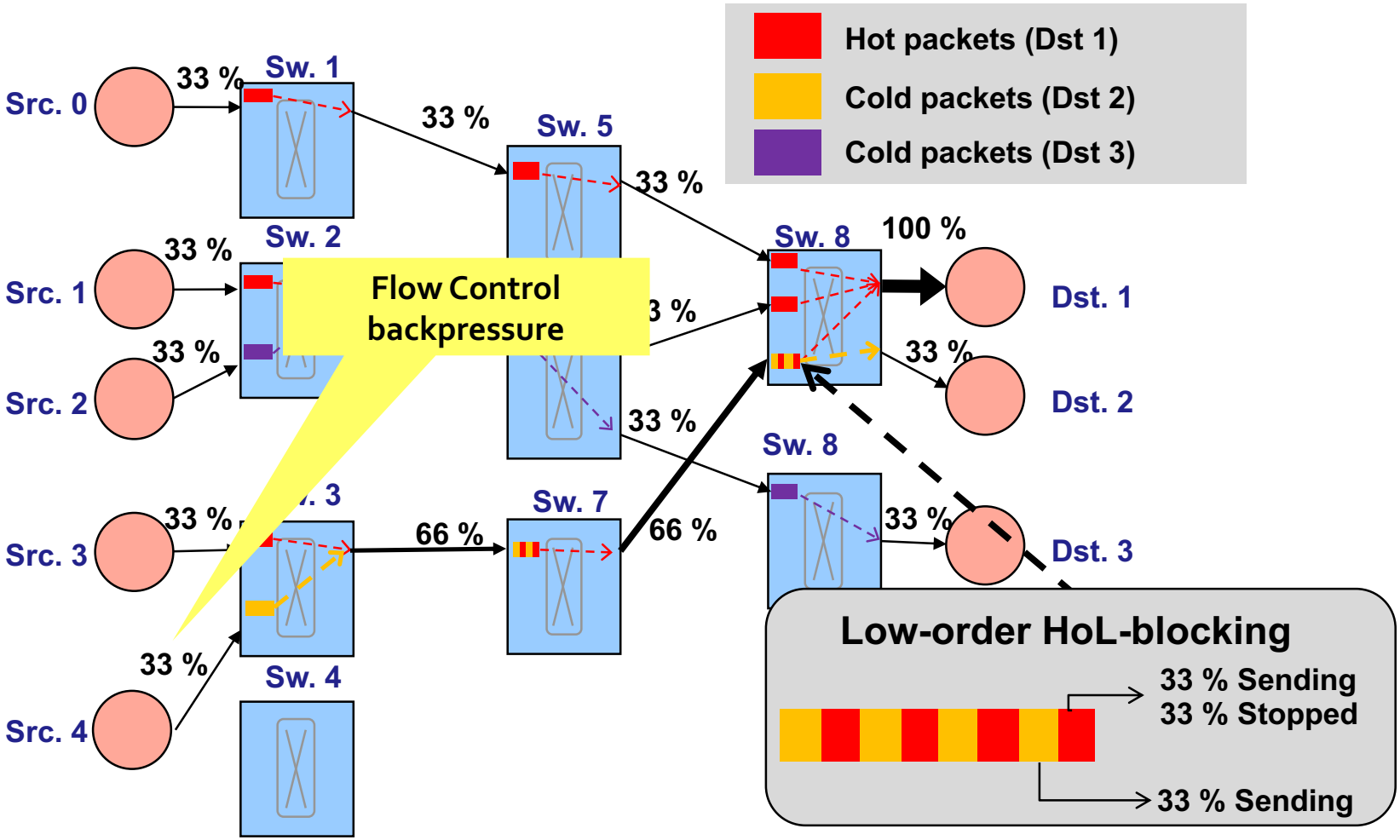
Traditional approaches

- ECN + PFC: The DCN network equipment reacts to congestion using ECN + PFC and smart buffer management in an attempt to eliminate the congestion tree.
- Problems:
  1. Large DCN networks have more hops, increasing the **closed-loop reaction time of ECN**.
  2. More **traffic in flight makes it difficult for ECN to react** to sudden traffic bursts.
  3. PFC generates **HoL blocking** in upstream switches.
  4. **Injection throttling** may be triggered at sources not contributing to congestion.



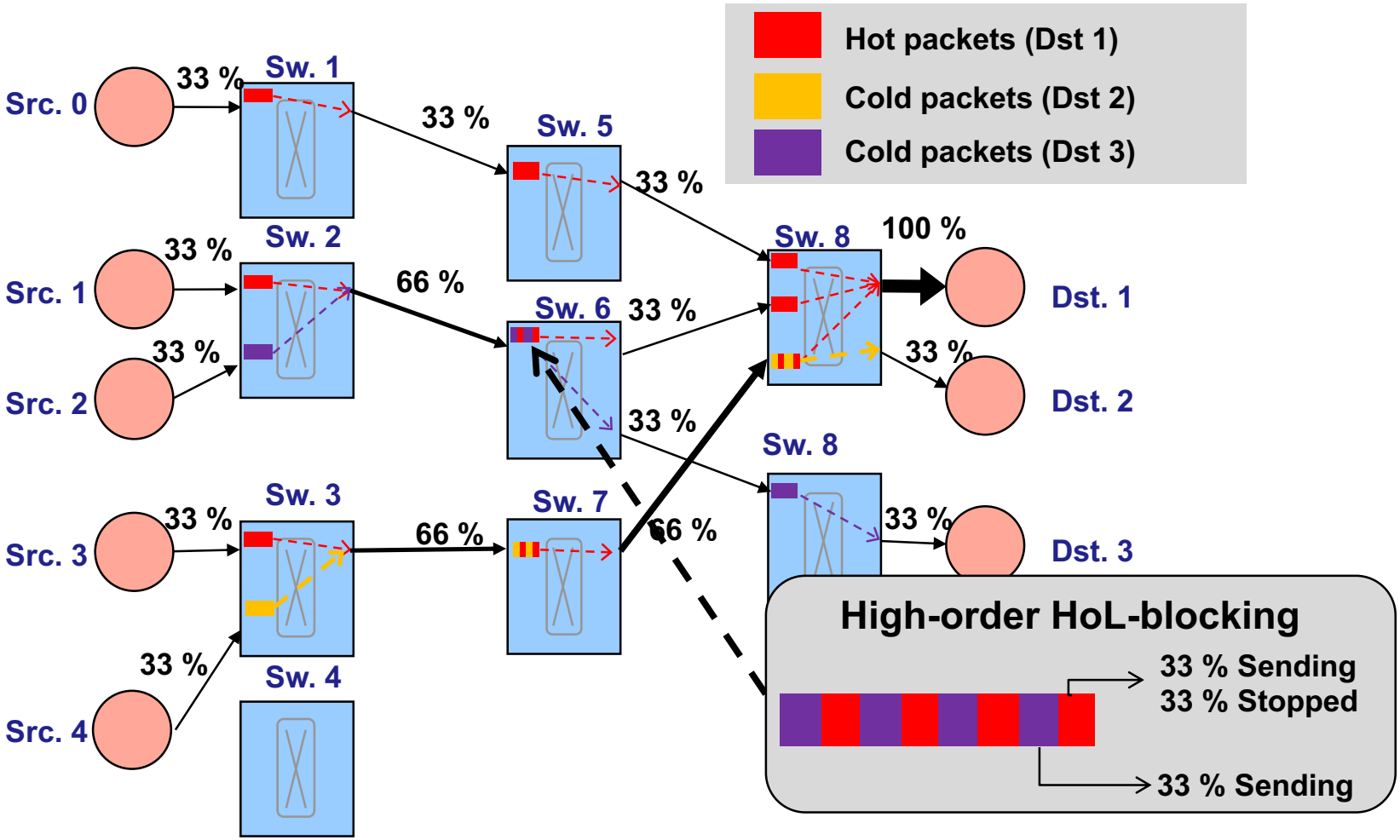
Current congestion control

Head-of-Line (HOL) Blocking problem [Karol87]



Current congestion control

Head-of-Line (HOL) Blocking problem [Karol87]



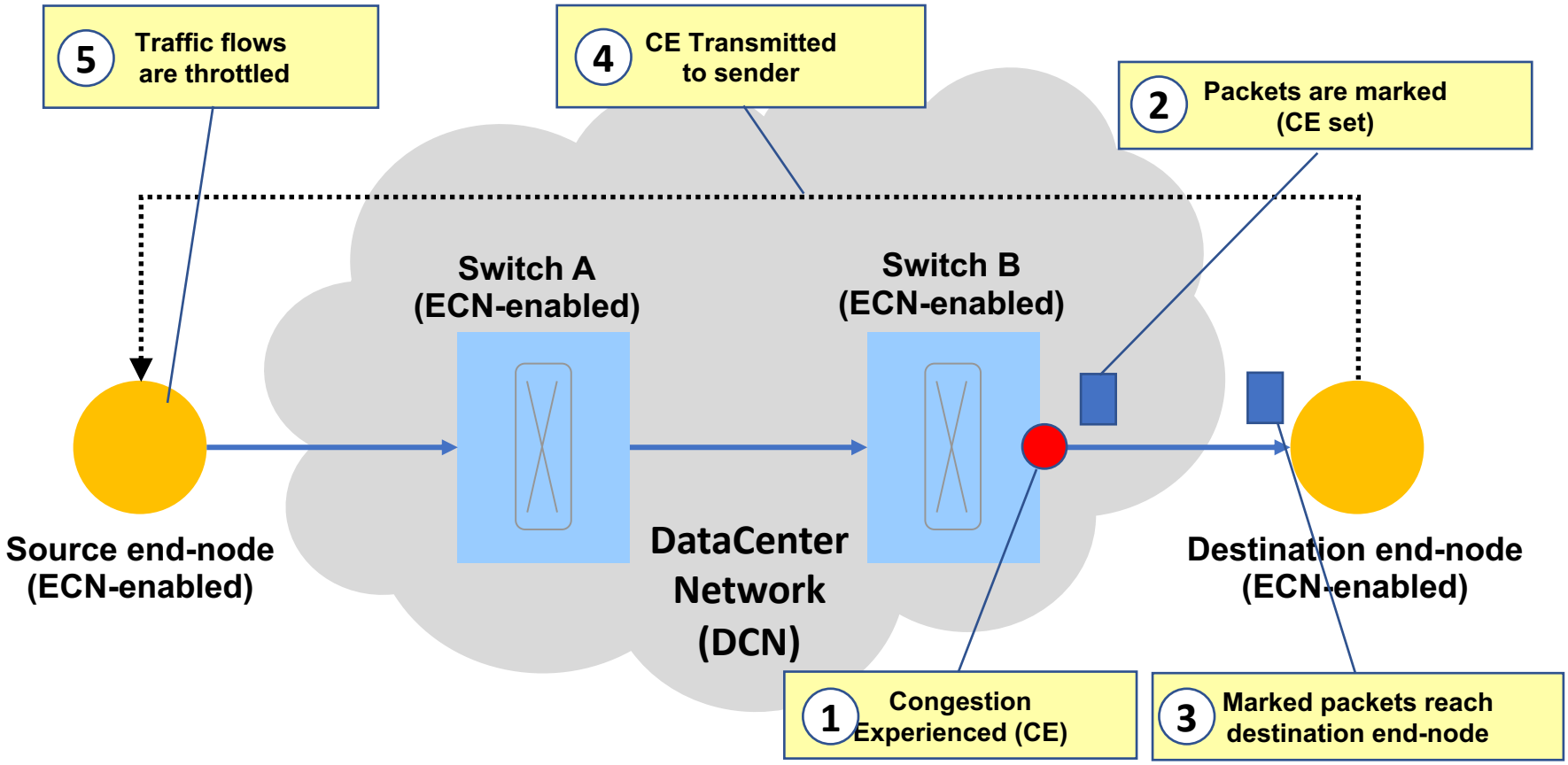
Current congestion control

Head-of-Line (HOL) Blocking problem [Karol87]

- The **problem is not the congestion itself** but the HOL blocking that spoils the traffic flows performance (throughput and latency).
- By preventing the HOL blocking, congestion becomes harmless.
- A successful strategy for reducing congestion impact on network performance should be focused on **preventing HOL blocking immediately, while congestion trees are drained** (e.g. by an ECN-like approach).

Current congestion control

Explicit Congestion Notification (ECN) [RFC 3168]



Current congestion control

Explicit Congestion Notification (ECN) [RFC 3169]

- Packets are just marked, based on a queue threshold that triggers the congestion detection.
- Long notification delays: once marked, packets have to reach destination, be processed at the destination NIC, and that NIC has to send notification to the source NIC. After a while, congestion notification reaches the source NIC, and then the actual throttling happens.
- Injection throttling may be based on obsolete information due to congestion dynamics and long notification delays.
- **ECN reduces HOL blocking but it does not directly approach this problem.**

Consequences

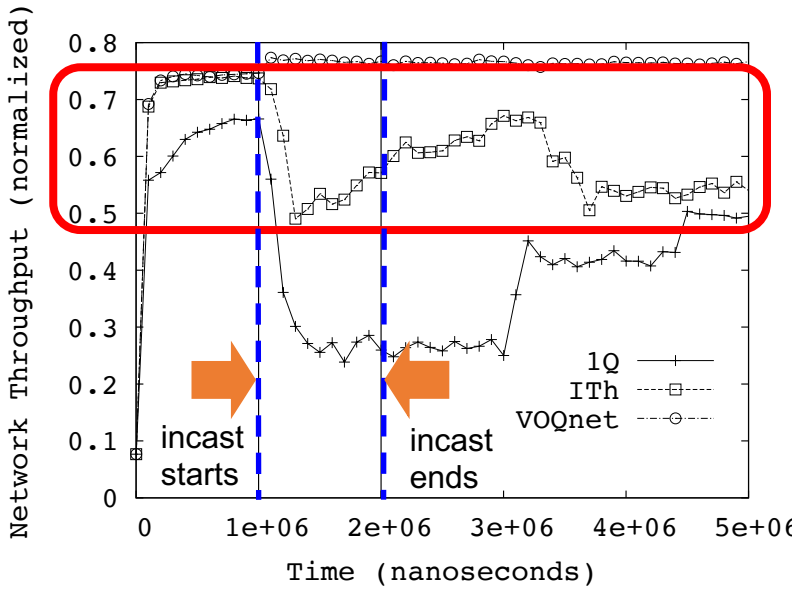
- Closed-loop control system with long delay in the feedback chain:
  - The delay from congestion detection to actual traffic throttling can be very long.
  - Depending on the DCN topology diameter and routing, this delay may be even worse.
- By the time the source NIC reacts, the congestion tree has grown significantly:
  - When delayed congestion notifications reach the NIC, then the throttling has to be more aggressive to reduce the already formed congestion tree.

Consequences

- Reaction based on possibly obsolete information:
  - Throttling at NICs may be disproportional with the network traffic status.
  - By the time the closed-loop mechanism actually reacts at NICs the congestion trees may have evolved (movable roots and branches).
- Overreaction due to delayed notifications
  - Aggressive throttling causes that flows throughput may be reduced excessively when congestion trees are disappearing or moving.
  - Inadequate reaction in case of transient congestion.
- ECN works work end-to-end (e2e), so that **transient congestion may still spoil network performance**.

Consequences

- The result are the oscillations in the injection rate, so that the DCN obtains low throughput and poor latency:
  - ECN is **very difficult to tune** as it depends on the traffic pattern [Escudero11].



Simulation results:

- *64-node 3-level CLOS*
- *75% Uniform/Random traffic*
- *25% Hot-spot traffic*
- *4 incast situations [1ms-2ms]*
- *Network configurations:*
  - *1Q: one queue per buffer*
  - *ITh: ECN-like technique*
  - *VOQnet (ideal): one queue per server node (64 queues)*

How can we improve it?

- By providing **more detailed feedback from the switches and packet headers.**
- By **distinguishing in-network from incast congestion.**
- By **speeding up notifications.**
- By implementing **fast-response mechanisms in the switches.**

How can we improve it?

More detailed feedback

- More accurate detection of packets really contributing to congestion **at switches** (separate them from victim packets):
  - Packet at the queue head, packet with longest accumulated delay.
  - Avoid false positives.
- To record accumulated packet delay in the **packet headers** and include this information in the notifications:
  - The longer the packet latency, the more intense the congestion, and more aggressive the injection throttling needs to be.

How can we improve it?

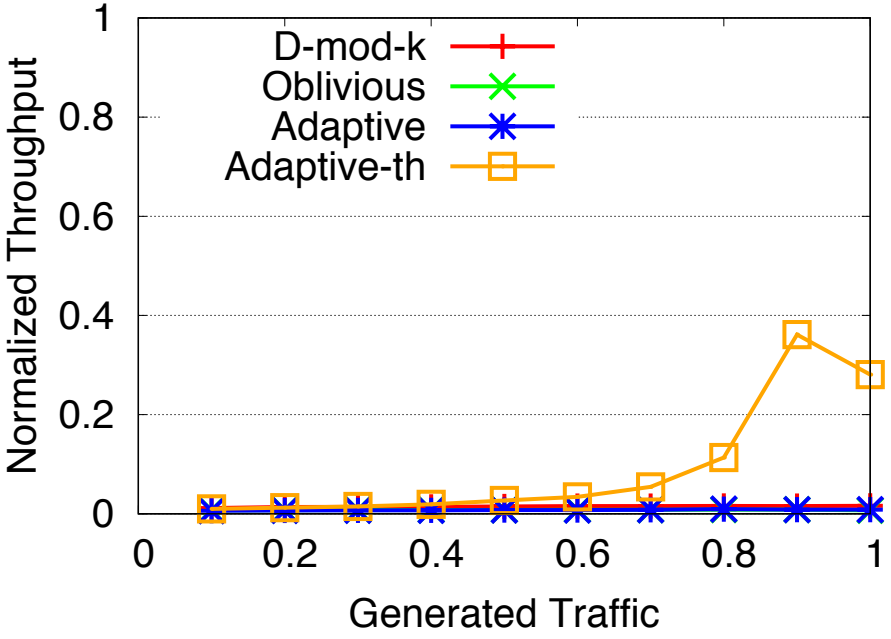
Distinguishing in-network from incast congestion

- Load balancing alleviates in-network congestion, but it does not work properly with incast congestion.
- ECN reduces incast scenarios but has several issues: signaling delays, oscillations, etc.
- **Congestion isolation** can be combined not only with ECN, but other techniques for load balancing and destination scheduling to work together.

How can we improve it?

Distinguishing in-network from incast congestion

- Load balancing is counterproductive when incast scenarios appear. It is better to limit this balancing under certain scenarios [Rocher17].



Simulation results:

- *11664-node 3-level CLOS,*
- *90% Uniform/Random traffic*
- *10% incast traffic (one congestion tree)*
- *Network configurations:*
  - *D-mod-k: Deterministic routing.*
  - *Oblivious: Random/exploits path diversity.*
  - *Adaptive: Fully adaptive routing.*
  - *Adaptive-th: Limits adaptivity.*

How can we improve it?
Speeding up congestion notifications

- The closed-loop approach is slow, as marked packets have to cover the path length to reach the destination end-node, and then the congestion notifications (e.g. CNPs, TCP ACKs, etc.) need to travel backwards to reach the source end-nodes:
  - Notifications directly from switches backwards to other switches and end-nodes will speed up the congestion notification to both switches and NICs.
- The congestion notification mechanism can be in sync with ECN and congestion isolation, so that they leverage CNPs to work better.

How can we improve it?

Fast-response mechanisms in the switches

- Fast congestion detection mechanisms.
- Fast reaction once congestion is detected, by isolating the congested flows at switch buffers:
  - While ECN reacts, HoL blocking can be immediately prevented if the congested flows are isolated.
- ECN reaction will be focused only on congested flows, already isolated in special buffer space at switches:
  - The reaction time of ECN is not so critical anymore.
  - ECN helps congestion isolation to release the resources used to isolate congested flows, making them available for future congestion.

Conclusions

- **End-to-end solutions incur in delays**, due to the closed-loop approach when congestion appears.
- We have analyzed the pros and cons of ECN, in order to **improve its design**:
  - More detailed feedback.
  - Distinguish in-network from incast congestion.
  - Speeding up congestion notifications.
  - Fast-response congestion mechanisms at switches.
- ECN can be **combined with fast-response mechanisms at switches** to prevent HoL-blocking immediately while the congestion tree is reduced due the injection throttling.

References

[Congdon18] Paul Congdon et al: **The Lossless Network for Data Centers**. NENDICA "Network Enhancements for the Next Decade" Industry Connections Activity, IEEE Standards Association, 2018.

[Garcia05] P. J. Garcia, J. Flich, J. Duato, I. Johnson, F. J. Quiles, and F. Naven, "**Dynamic Evolution of Congestion Trees: Analysis and Impact on Switch Architecture**," in High Performance Embedded Architectures and Compilers, ser. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Nov. 2005, pp. 266–285.

[Garcia19] Pedro Javier Garcia, Jesus Escudero-Sahuquillo, Francisco J. Quiles and Jose Duato, "**Congestion Management for Ethernet-based Lossless DataCenter Networks**" DCN: 1-19-0012-00-Icne.

[Karol87] M. J. Karol, M. G. Hluchyj, S. P. Morgan, "Input versus output queuing on a space-division packet switch", *IEEE Trans. Commun.*, vol. COM-35, no. 12, pp. 1347-1356, Dec. 1987.

[RFC 3168] K. Ramakrishnan et al. **The Addition of Explicit Congestion Notification (ECN) to IP**. RFC 3168, Year 2001**:** https://tools.ietf.org/html/rfc3168.

[Congdon19Qcz] Paul Congdon: P802.1Qcz – Congestion Isolation. **Standard for Local and Metropolitan Area Networks — Bridges and Bridged Networks — Amendment: Congestion Isolation.** PAR approved 27 Sep 2018.

[Escudero11] 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.

[Rocher17] Jose Rocher-Gonzalez, Jesús Escudero-Sahuquillo, Pedro Javier García, Francisco J. Quiles: **On the Impact of Routing Algorithms in the Effectiveness of Queuing Schemes in High-Performance Interconnection Networks**. Hot Interconnects 2017: 65-72.

[Escudero19] Jesús Escudero-Sahuquillo, Pedro Javier García, Francisco J. Quiles, José Duato: **P802.1Qcz interworking with other data center technologies**. IEEE 802.1 Plenary Meeting, San Diego, CA, USA July 8, 2018 (cz-escudero-sahuquillo-ci-internetworking-0718-v1.pdf)