Efficient Dynamic Isolation of Congestion in Lossless DataCenter Networks

Luis Gonzalez-Naharro[*], Jesus Escudero-Sahuquillo[*], Pedro J. Garcia[*], Francisco J. Quiles[*], Jose Duato[†], Wenhao Sun[‡], Li Shen[‡], Xiang Yu[‡], and Hewen Zheng[‡].

*: University of Castilla-La Mancha, Spain.

†: Universitat Politècnica de València, Spain.

‡: Huawei Technologies Co., Ltd., China.

ACM SIGCOMM 2019 NEAT Workshop
August 19, 2019, Beijing, China

OUTLINE

1. MOTIVATION
2. BACKGROUND
3. DVL DESCRIPTION
4. EVALUATION
5. CONCLUSIONS

MOTIVATION

MOTIVATION

- Modern DCs support lots of latency-sensitive applications.
  - Machine / deep learning, big-data, cloud-computing, etc.
- To meet these latency requirements, RoCEv2 (RDMA over Converged Ethernet) is usually employed.
- However, retransmission introduces latency overhead → **Lossless** networks are increasingly used in DCs.
- But, **lossless** networks have congestion problems → Usage of PFC (Priority-based Flow Control) for flow control → Congestion is propagated, performance degrades!
- DC applications often generate bursty, many-to-one traffic which favors congestion.

MOTIVATION

- Usual congestion control approach: **Injection throttling** (such as ECN).
- Drawback: it is slow, and creates oscillations in the injection rate.
- Solution: **Dynamic Virtual Lanes (DVL)**:
  - Congestion isolation locally implemented at every switch → **Very fast response.**
  - Congested flows moved to special queues → **Eliminates HoL blocking.**
  - Propagates congestion information to upstream switches.
  - Only a special queue per port → **Resource saving.**
  - New special queue deallocation and in-order delivery guarantee mechanisms.

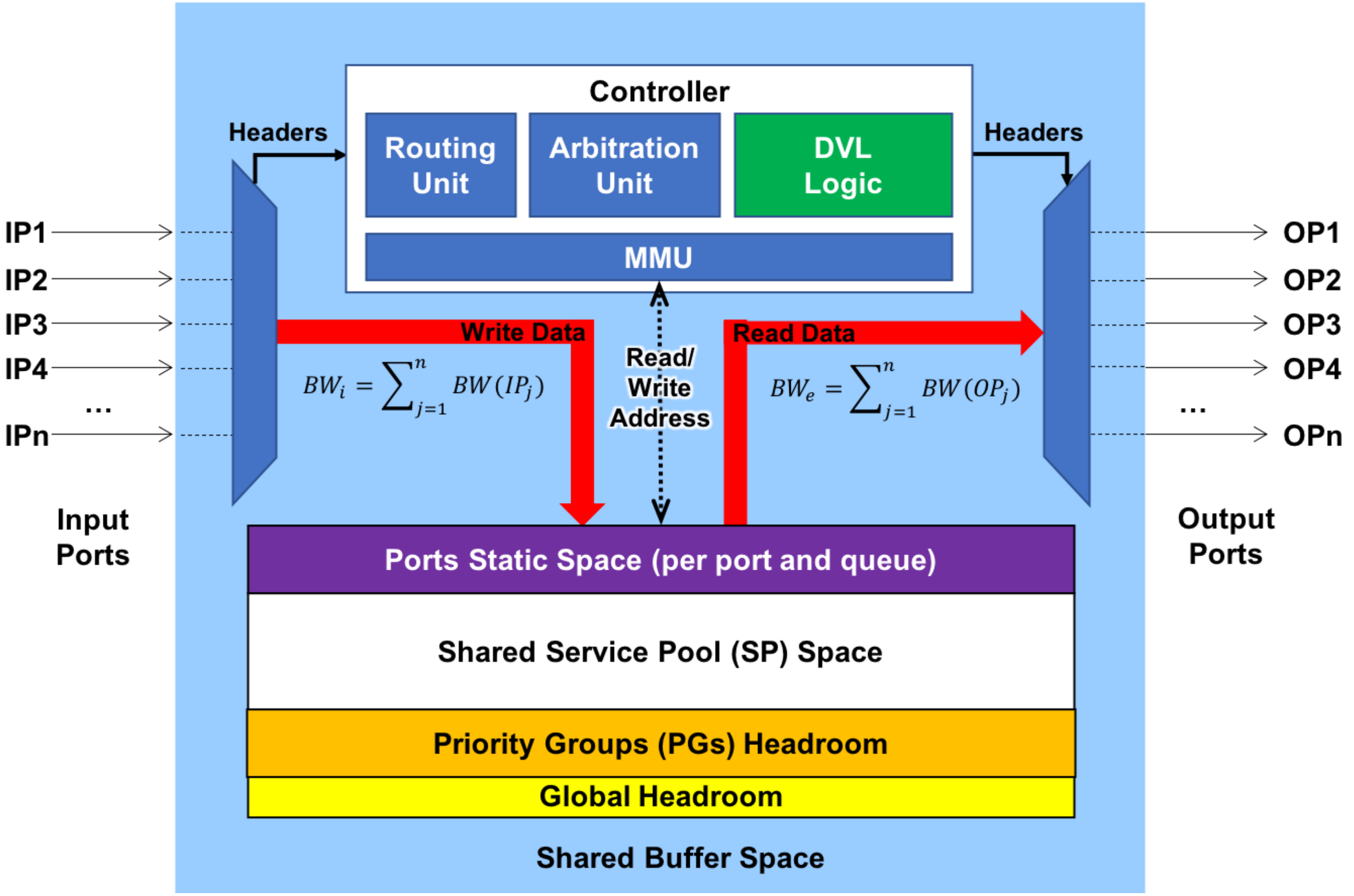
BACKGROUND

BACKGROUND

- Congestion damages performance, since congested flows slow down non-congested flows: **Head-of-Line blocking** (HoL blocking).
- Different techniques to solve congestion:
  - **Load-balancing techniques** (ECMP): packets will eventually meet at the same point.
  - **Injection Throttling** (ECN, QCN): huge time lapse between congestion detection and source reaction.
  - **Destination Scheduling:** also slow, based on end-to-end feedback.
  - **Static queues:** congested and non-congested flows may still share queues.
  - **Dynamic congestion isolation:** theoretically fast reaction time, and HoL blocking is eliminated.

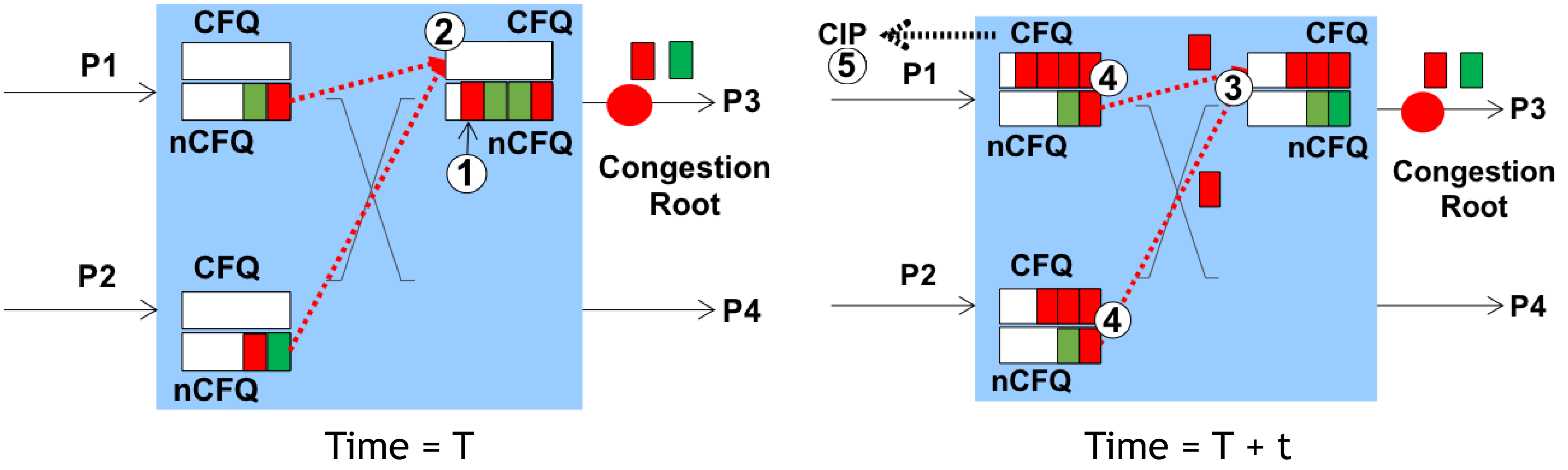
DVL DESCRIPTION

SWITCH ARCHITECTURE



- DVL operates on top of **shared-buffer switches:** packets stored in a centralized memory.
- Memory filling order: Static space → Shared Pool → PG Headroom → Global headroom

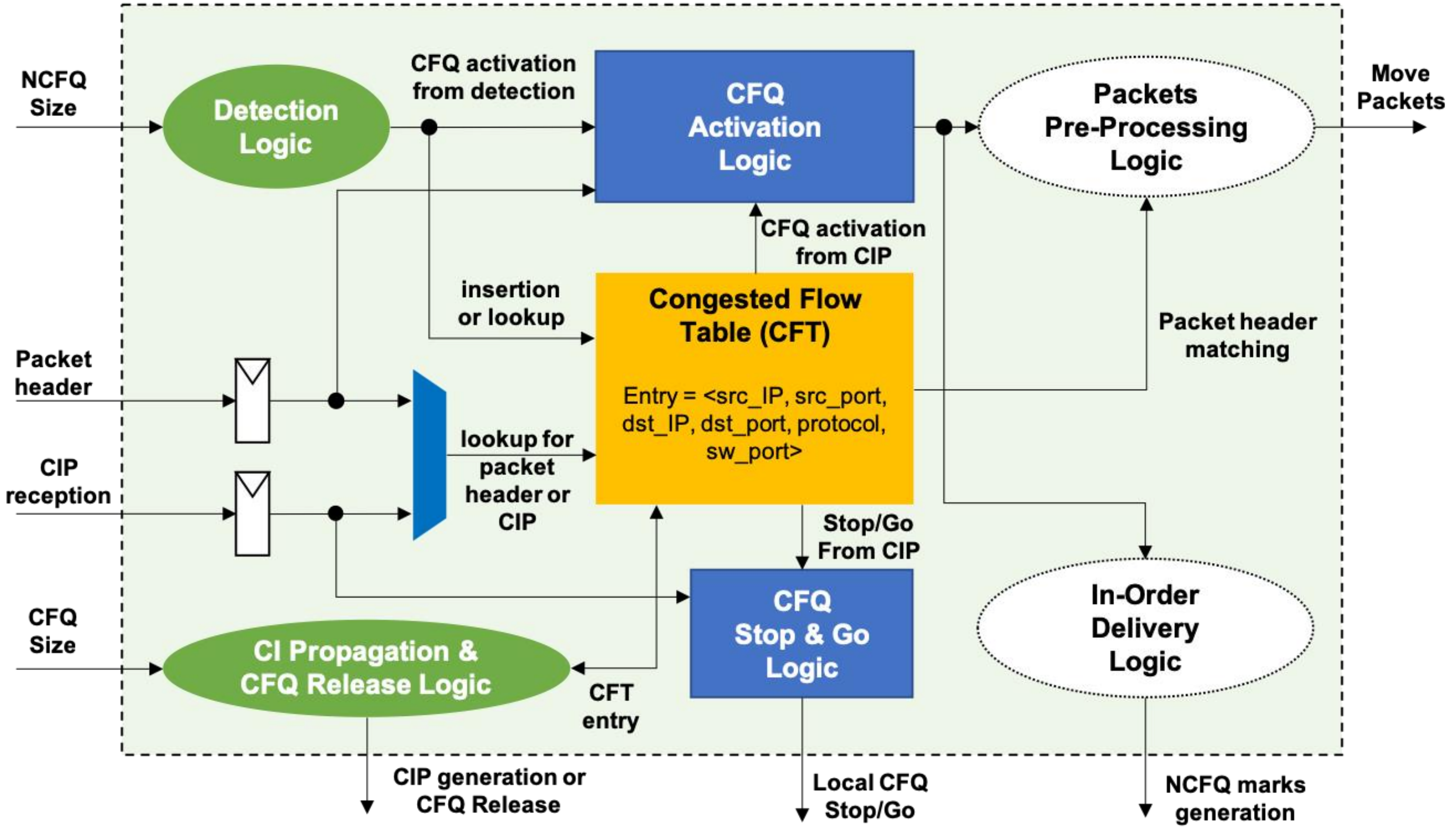
BASIC DVL OPERATION



Congestion detection and isolation mechanism:

- #1: Congestion detected when nCFQ (Non-Congested-Flow Queue) reaches a threshold.
- #2: CFQ (Congested-Flow Queue) is allocated.
- #3: Incoming congested packets (egress) stored at CFQ.
- #4: Egress CFQ grows, CFQ allocated at ingress.
- #5: Ingress CFQ grows, congestion information sent upstream.

DVL LOGIC



- CFT keeps information regarding congested flows (source and destination IPs and ports, protocol and switch port).
- When congestion is detected, packet header is used to fill a CFT entry.
- Pre-processing: incoming packets matching an entry will be stored in the CFQ.

CFQ DEALLOCATION & IN-ORDER DELIVERY

- CFQ deallocation is local to each switch (enhanced DVL).
- **Deallocation** if occupancy of CFQ+nCFQ < Congestion detection threshold.
- **Markings** used in allocation and deallocation to guarantee **in-order delivery**:
  - They act as a synchronization point between the CFQ and nCFQ.
  - Inserted when a CFQ is allocated or deallocated.
  - If a marking is at the head of a queue, it gets blocked.
  - Markings are deleted when they are at the head of both the CFQ and nCFQ.
  - If marking is active in the CFQ and nCFQ, flow control from downstream switches will pause both queues (enhanced DVL).

EVALUATION

EVALUATION

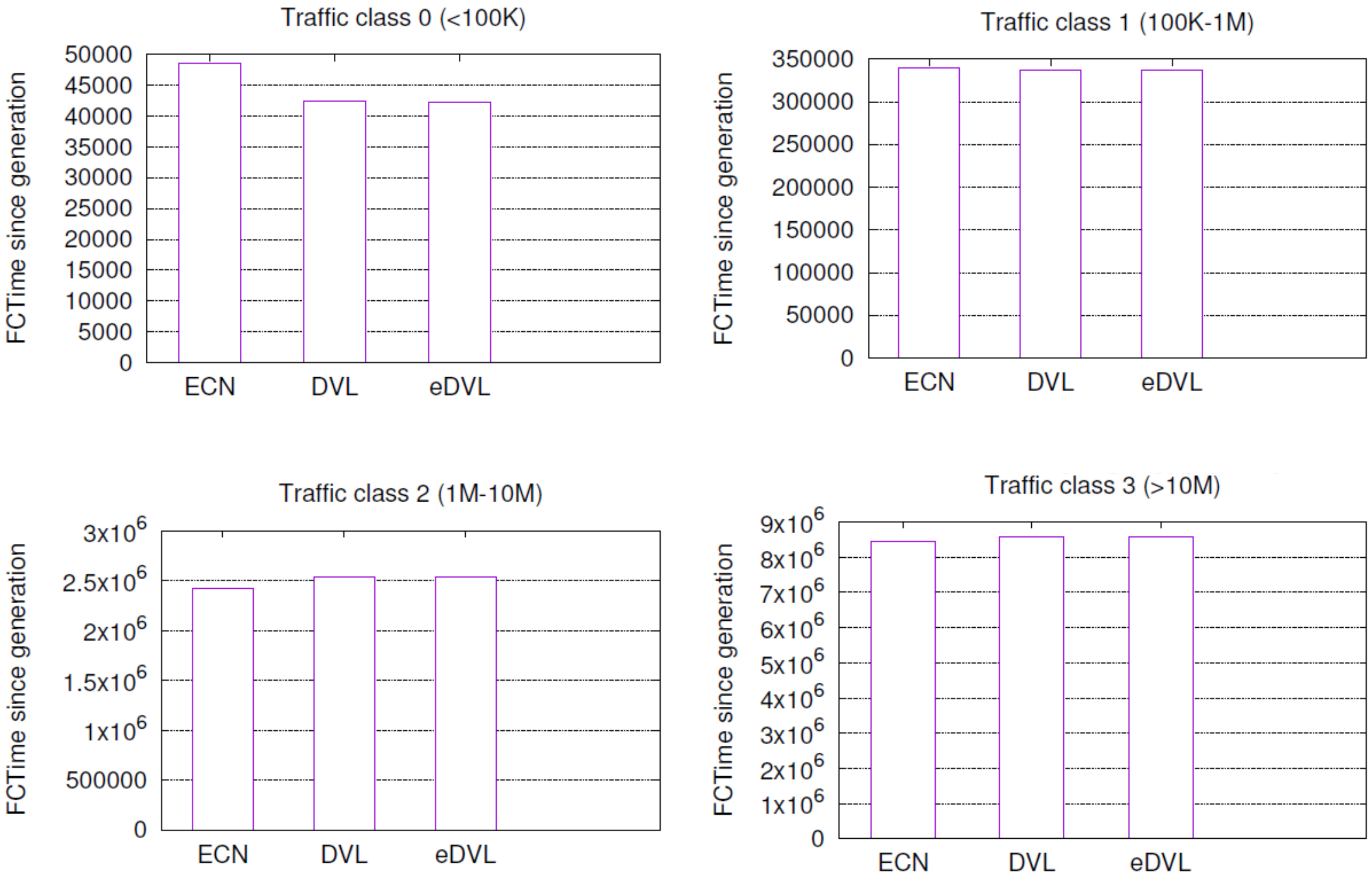
- Experiments carried out with a custom-made event-driven simulator.
- Assumed full-duplex pipelined links with 40 Gbps of bandwidth and 1μs of delay.
- Employed networks:
  - #1: 1024-node CLOS of 3 stages (48 8-port switches).
  - #2: 2048-node CLOS of 2 stages (96 64-port switches).
- 3MB and 24MB shared-buffer switches in each configuration.
- D-mod-K routing, PFC flow control, and 1000-byte MTU.
- Strategies tested: ECN, DVL and enhanced DVL (eDVL).
- NICs with as many queues as destinations in the network.

EVALUATION

- Synthetic traffic with the following Traffic Class (TC) distribution, obtained from [1]:
  - TC0: 1-100KB messages, 69.52% of overall traffic.
  - TC1: 100KB-1MB messages, 25,3% of overall traffic.
  - TC2: 1-10MB messages, 3% of overall traffic.
  - TC3: 10-30MB, 2.18% of overall traffic.
- 10,000 and 50,000 flows generated in 2ms for networks #1 and #2, respectively.
- Flow completion time between flow generation and flow last packet injection recorded as metrics

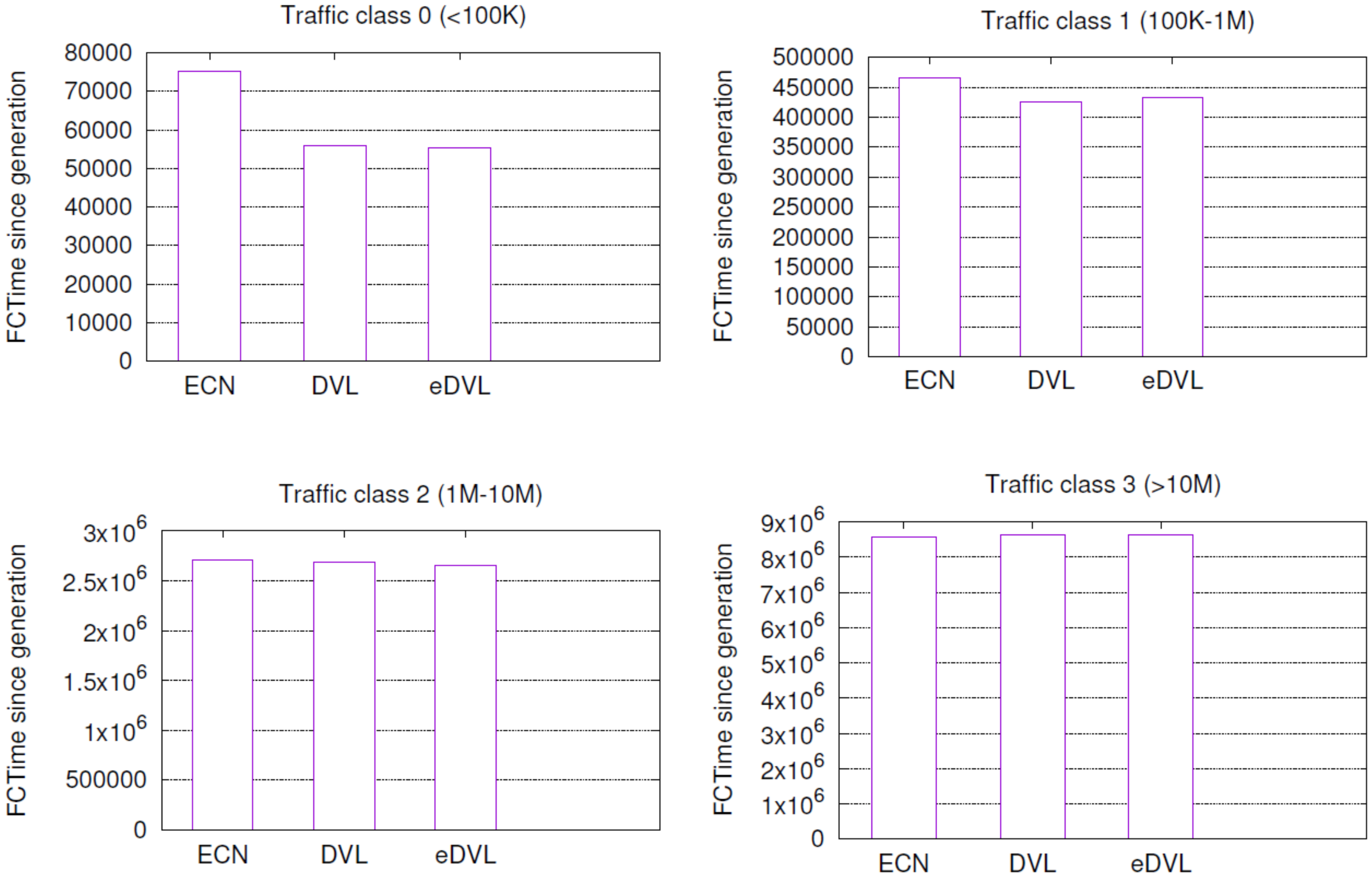
[1]: Jasmeet Bagga, George Porter, Arjun Roy, Hongyi Zeng and Alex C. Snoeren. 2015. Inside the Social Network's (Datacenter) Network. In Proceedings of SIGCOMM '15, August 17–21, 2015, London, United Kingdom.

EVALUATION



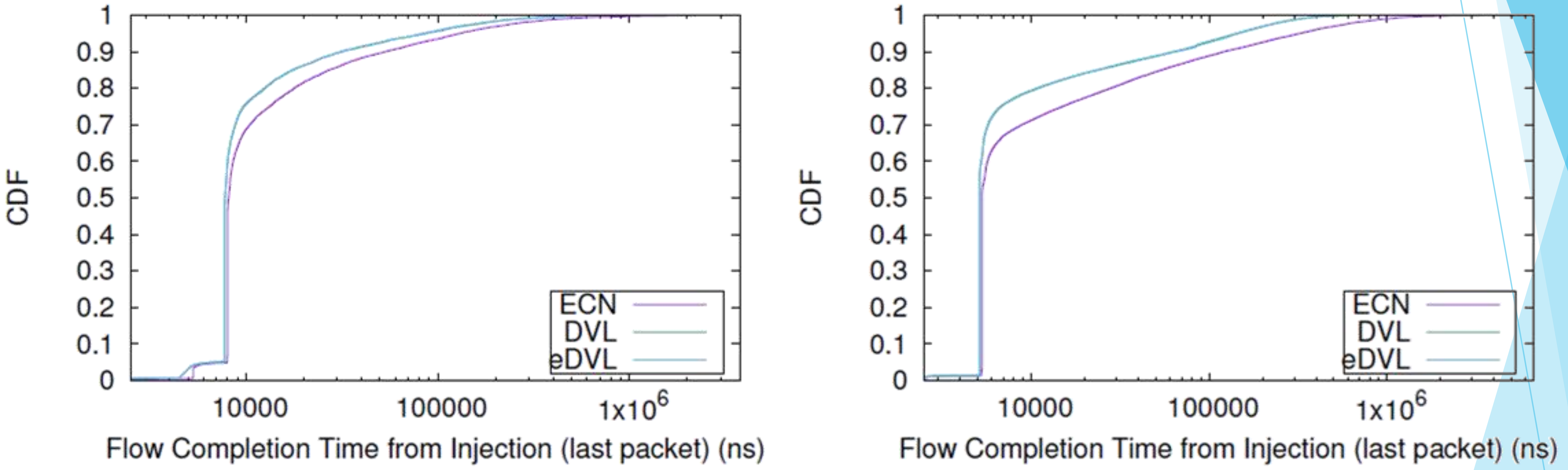
Flow completion times (generation) for network #1

EVALUATION



Flow completion times (generation) for network #2

EVALUATION



(a) Network Configuration #1. **(b) Network Configuration #2.**

Cumulative Distribution Function (CDF) of FCT

CONCLUSIONS

CONCLUSIONS

- Traditional solutions for solving congestion in DCs are not suitable for latency requirements.
- DVL reacts locally and immediately to congestion situations, isolating the congested flows and so eliminating HoL blocking.
- DVL uses resources more efficiently than previous proposals.

THANK YOU!
ANY QUESTIONS?