# Extending commodity OpenFlow switches for large-scale HPC deployments

**Mariano Benito**[†]
Enrique Vallejo[†]
Ramón Beivide[†]
Cruz Izu[‡]

[†] University of Cantabria
[‡] The University of Adelaide

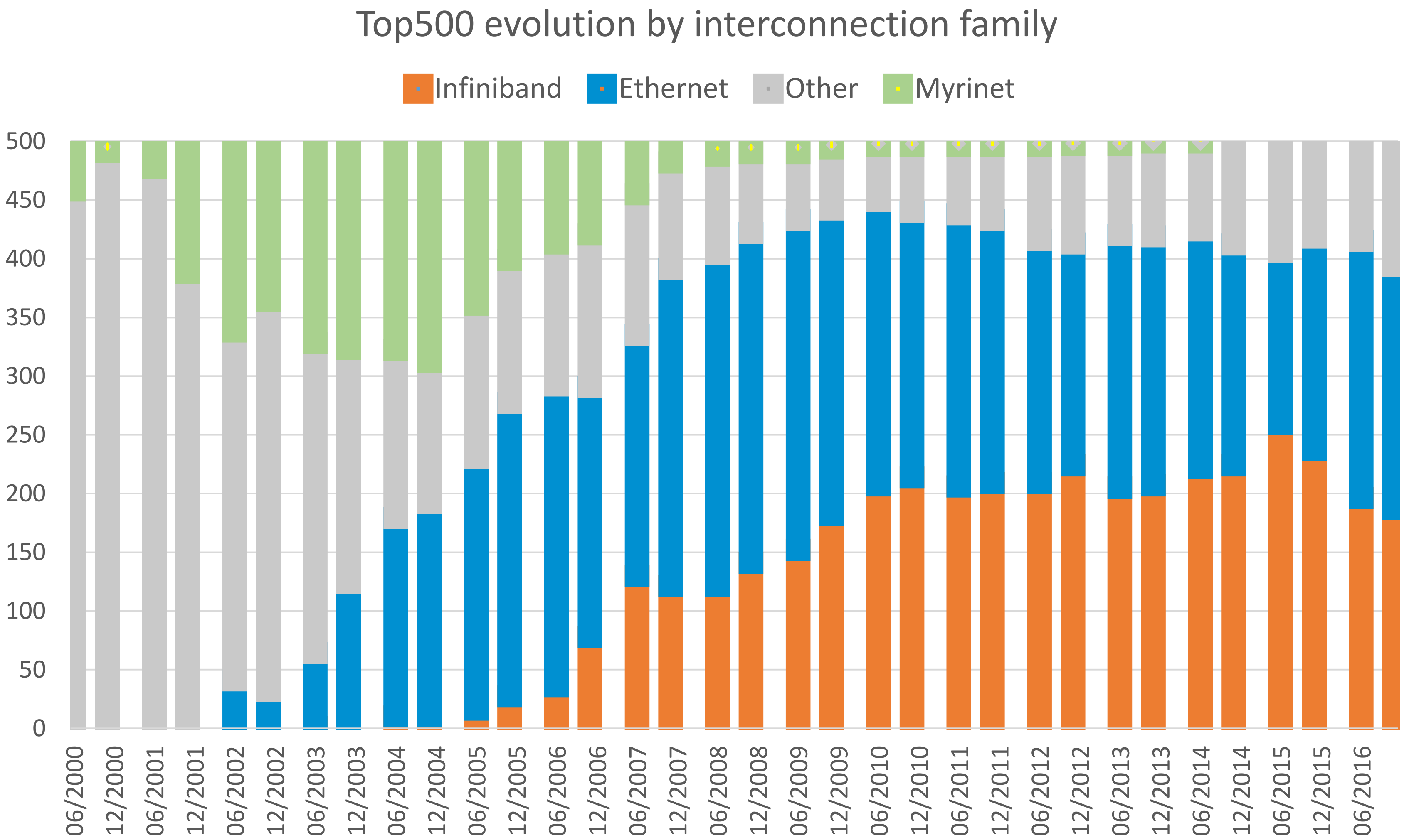
HiPINEB'17, 5 February 2017

# Overview

1. Introduction
    1. Ethernet & Dragonfly
    2. Previous work – Conditional flow rules & limitations
    3. Congestion control indicators
    4. Quantized Congestion Notification
2. Our proposal: QCN-SW
    1. QCN-SW + Source processing
    2. QCN-SW + Feedback comparison
3. Evaluation
4. Conclusions & Future work

# 1.1 Introduction Ethernet in HPC - Now

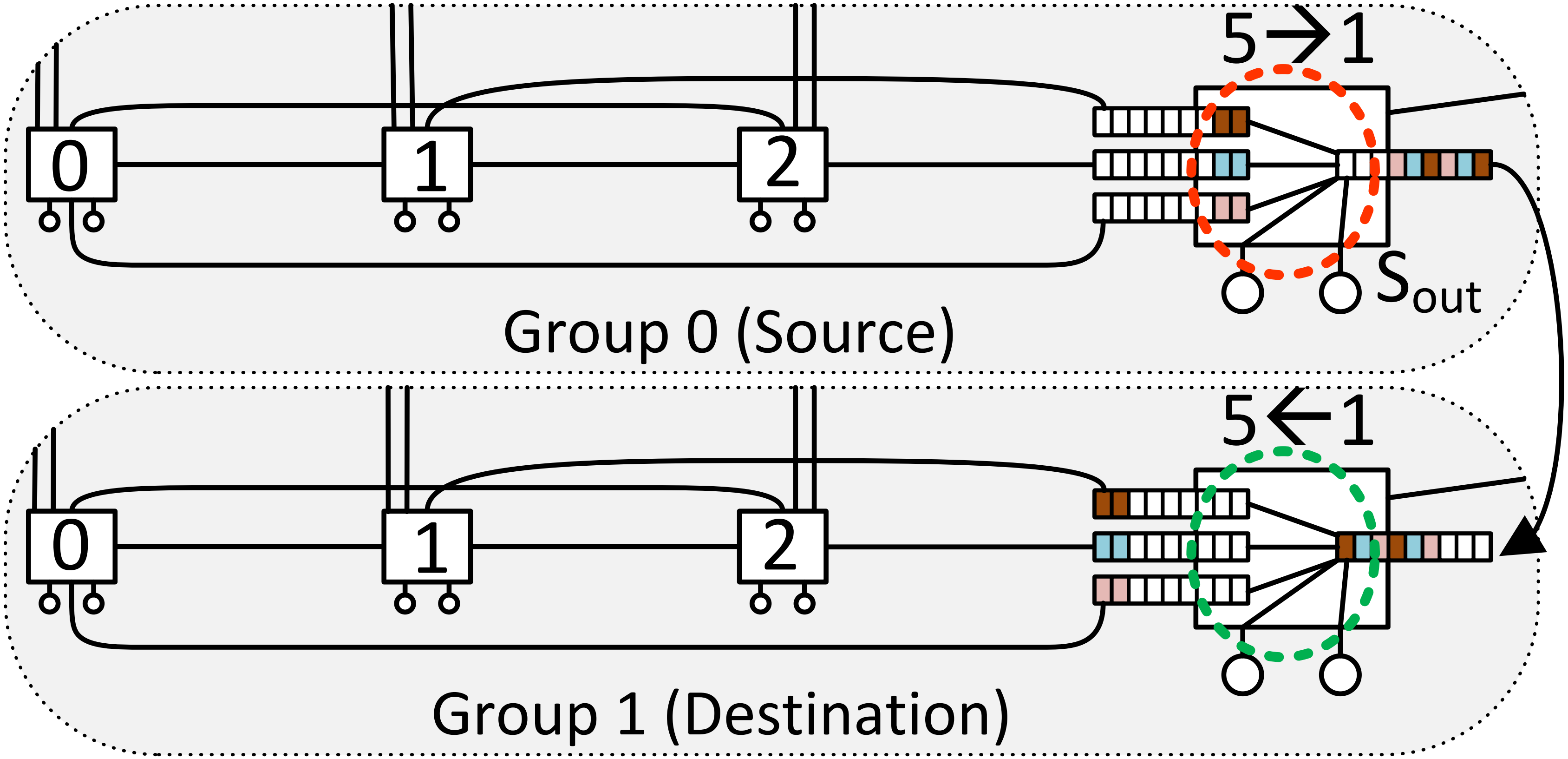
Technology evolution ⇨ Convergence in DC and HPC


Top500 evolution by interconnection family

# 1.1 Introduction Dragonfly topology and routing

- Minimal routing (local – global – local)
  - *Uniform traffic:* optimal throughput and latency
  - *Adversarial traffic: $S_{out}$ is a bottleneck*

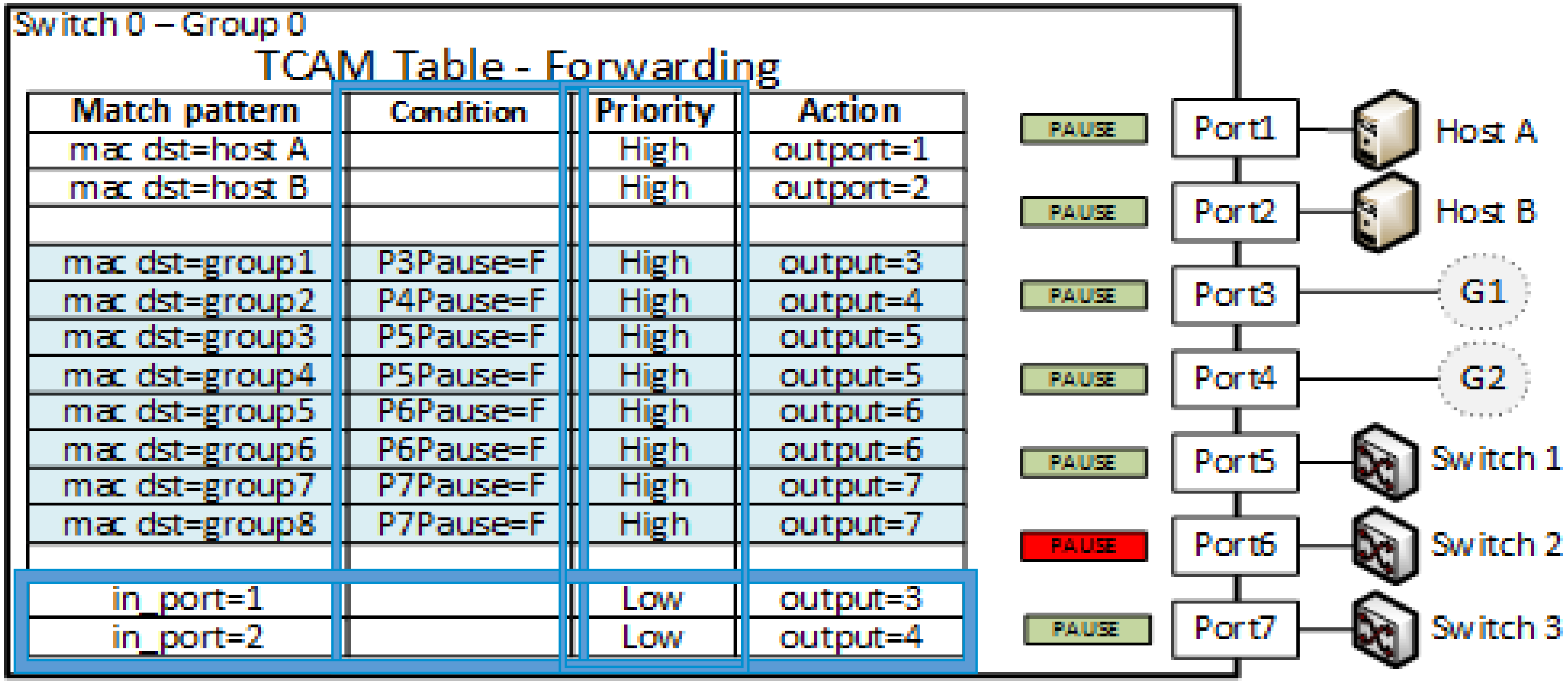
⇨ Non-minimal adaptive routing



Representation of adversarial traffic pattern in a (p=2, a=4, h=2) Dragonfly network.

# 1.2 Previous work Base design - Conditional rules

Add a “condition” to OpenFlow rules (similar to [1]) evaluated locally by switches



Switch 0 – Group 0

TCAM Table - Forwarding

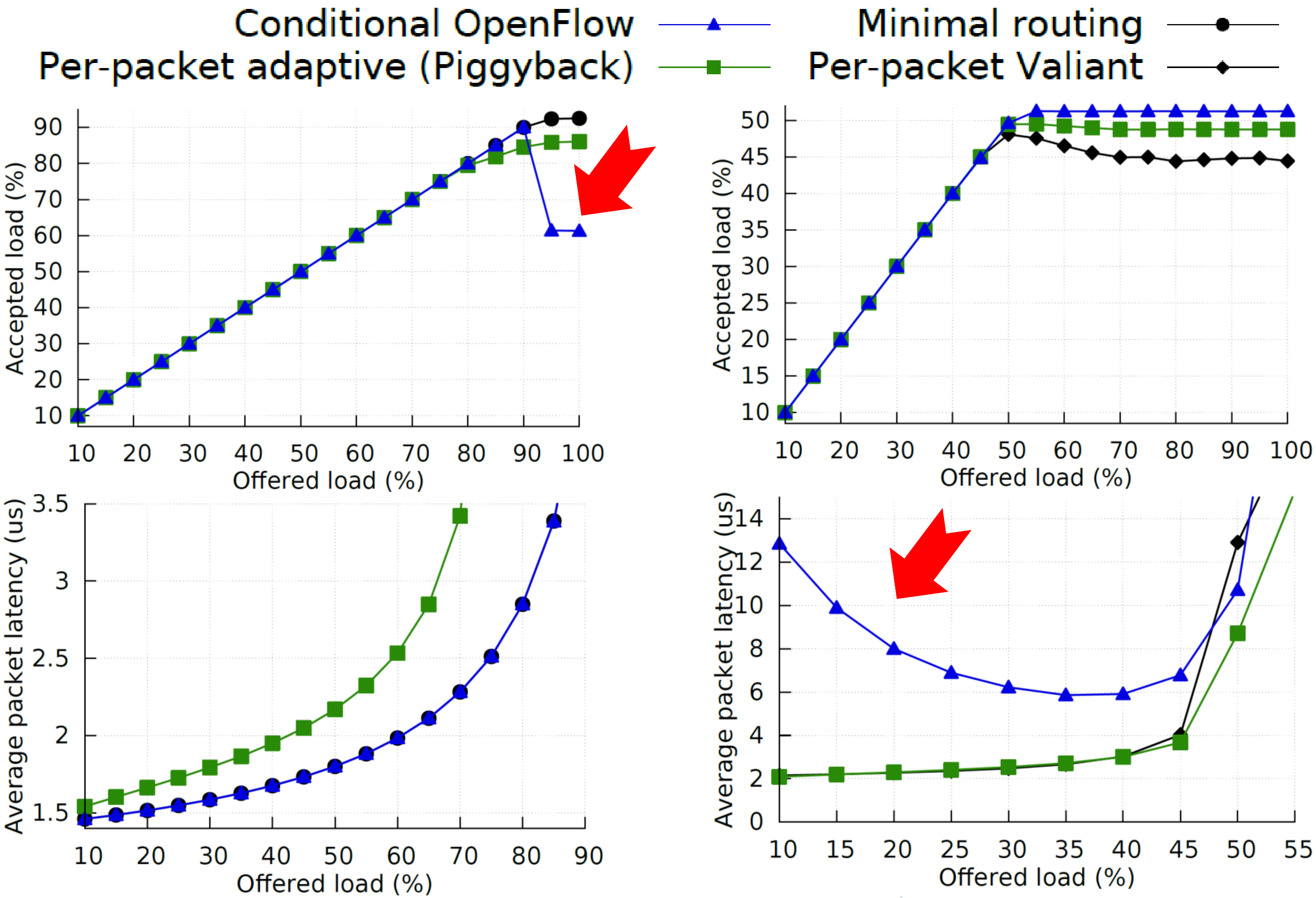
| Match pattern | Condition | Priority | Action |
|---|---|---|---|
| mac dst=host A | | High | outport=1 |
| mac dst=host B | | High | outport=2 |
| | | | |
| mac dst=group1 | P3Pause=F | High | output=3 |
| mac dst=group2 | P4Pause=F | High | output=4 |
| mac dst=group3 | P5Pause=F | High | output=5 |
| mac dst=group4 | P5Pause=F | High | output=5 |
| mac dst=group5 | P6Pause=F | High | output=6 |
| mac dst=group6 | P6Pause=F | High | output=6 |
| mac dst=group7 | P7Pause=F | High | output=7 |
| mac dst=group8 | P7Pause=F | High | output=7 |
| | | | |
| in_port=1 | | Low | output=3 |
| in_port=2 | | Low | output=4 |

Architecture of SW0 in G0 with Conditional Flow Rules for a Dragonfly network.

- ❑ Allows minimal and non-minimal adaptive routing in multipath topologies.
- ❑ Pro-active forwarding without a “central controller”
- ❑ Employs hierarchical addressing for allowing large-deployments in a flat Ethernet domain

[1] S. Shin et al, “AVANT-GUARD: Scalable and vigilant switch flow management in software-defined networks,” HOTI’09

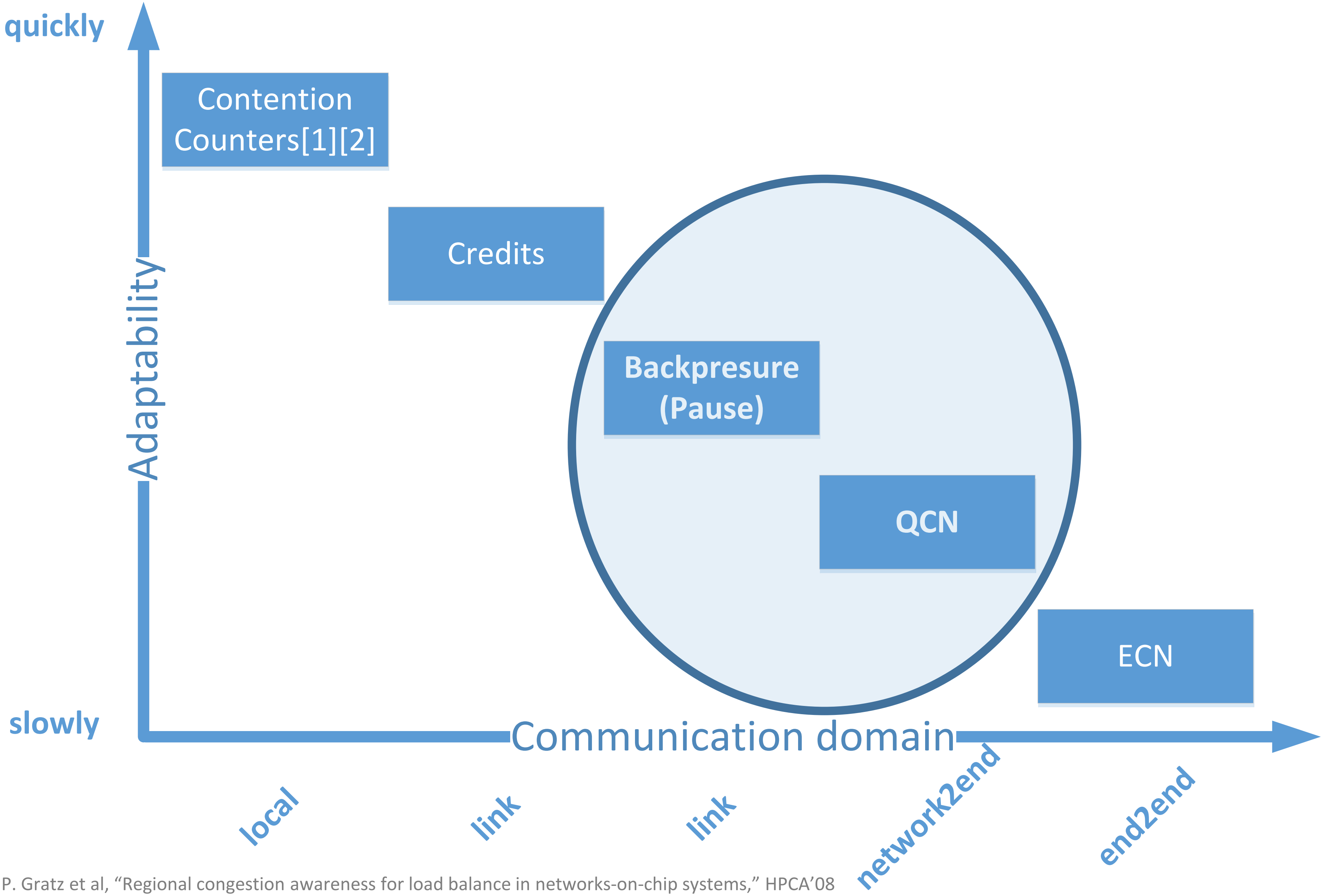
# 1.2 Previous work Base design – Limitations



Uniform traffic

Adversarial traffic

# 1.3 Introduction Congestion control indicators

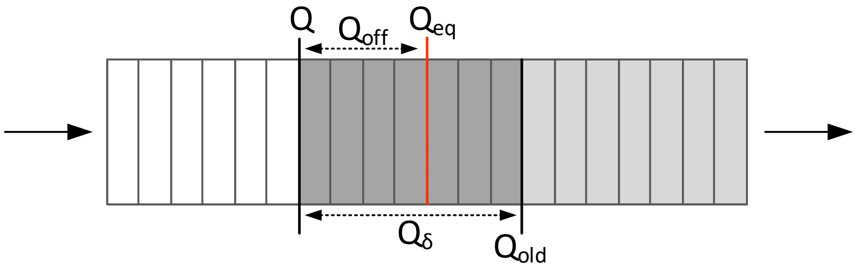


[1] P. Gratz et al, “Regional congestion awareness for load balance in networks-on-chip systems,” HPCA’08

[2] P. Fuentes et al, “Contention-based Nonminimal Adaptive Routing in High-radix Networks,” IPDPS’15

# 1.4 Introduction Quantized Congestion Notification

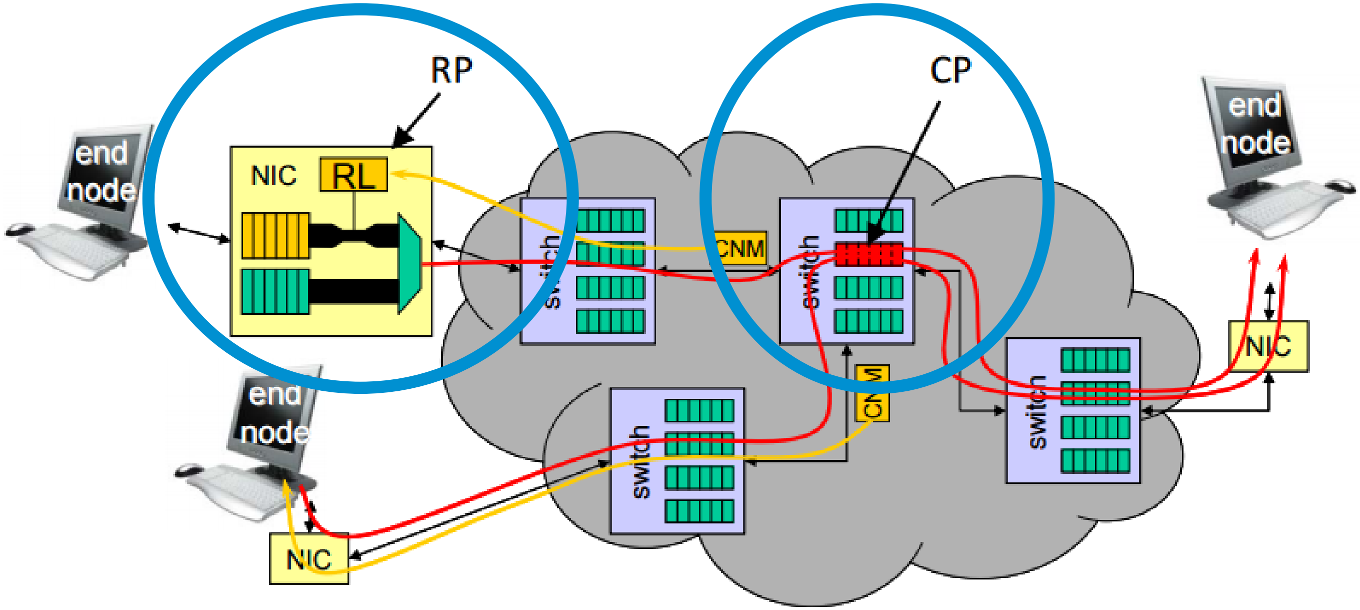
- ❑ QCN – Congestion notification in Layer-2 Ethernet
- ❑ Mainly composed of two elements:
  - Congestion point (CP)
  - Reaction point (RP)
- ❑ CPs generate explicit Congestion Notification Messages (CNMs) with a Fb value
  - Only negative notifications
- ❑ RPs implement injection throttling (AIMD)



$$Q_{off} = Q - Q_{eq}$$

$$Q_{\delta} = Q - Q_{old}$$

$$F_b = -(Q_{off} + wQ_{\delta})$$



Representation of QCN elements in a network. Source: [1] - IBM

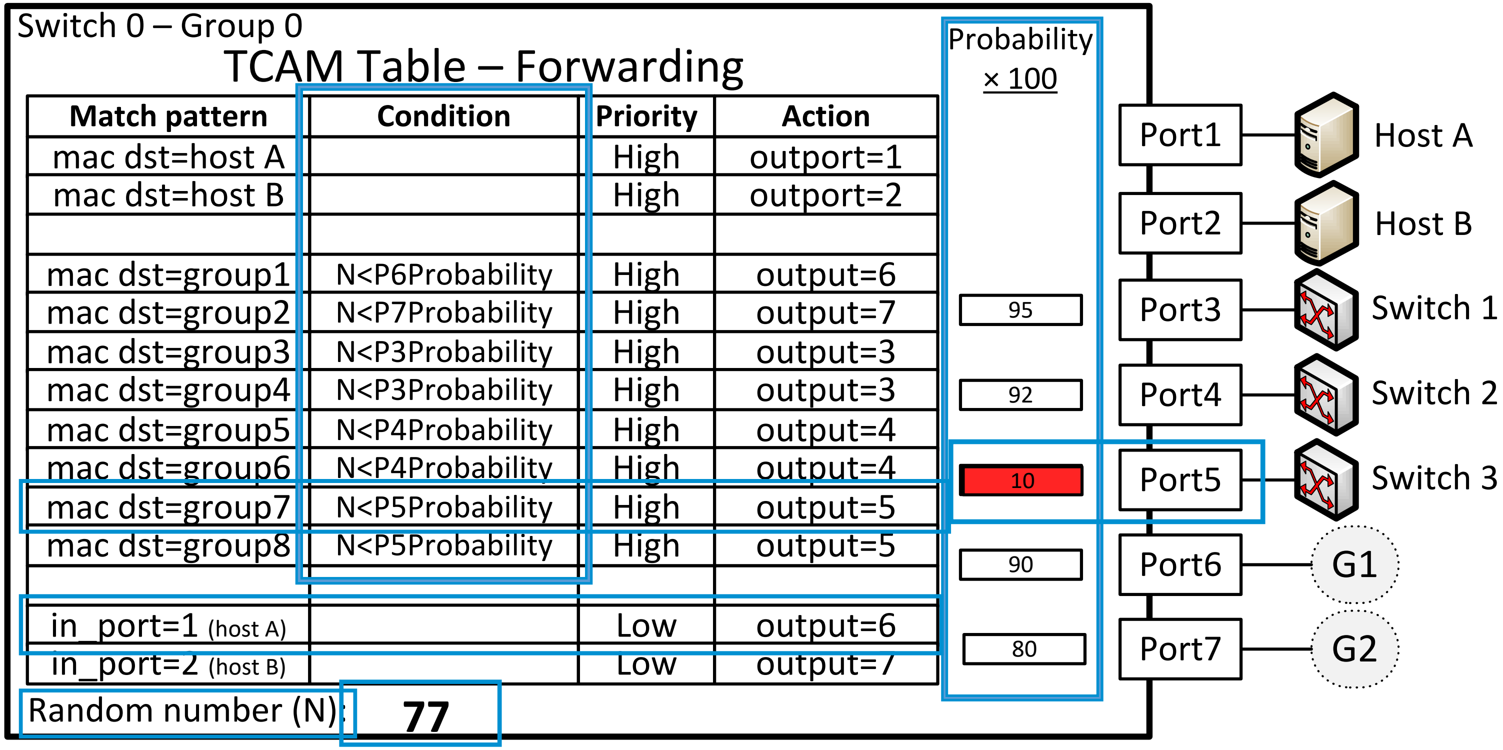
[1] http://www.hoti.org/hoti20/slides/Terabit_CEE_Switches-IBM.pdf

# Overview

1. Introduction
   1. Ethernet & Dragonfly
   2. Previous work – Conditional flow rules
   3. Congestion control indicators
   4. Quantized Congestion Notification
2. Our proposal: QCN-SW
   1. QCN-SW + Source processing
   2. QCN-SW + Feedback comparison
3. Evaluation
4. Conclusions & Future work

# 2. Our proposal : QCN-SW

- Source adaptive routing based on:
  - Assign to each port a probability of sending minimally
  - Take advantage of QCN CNMs for manipulating this probability using AIMD policy
    - Increase probability by a fix % autonomously based on byte counting or timer
    - Reduce probability by a factor R in the range [0.5, 1] when a CNM is received
  - A random value (N) between 0 and 100 for each table lookup
  - Extending Pro-active Conditional OpenFlow rules



Switch 0 – Group 0

TCAM Table – Forwarding

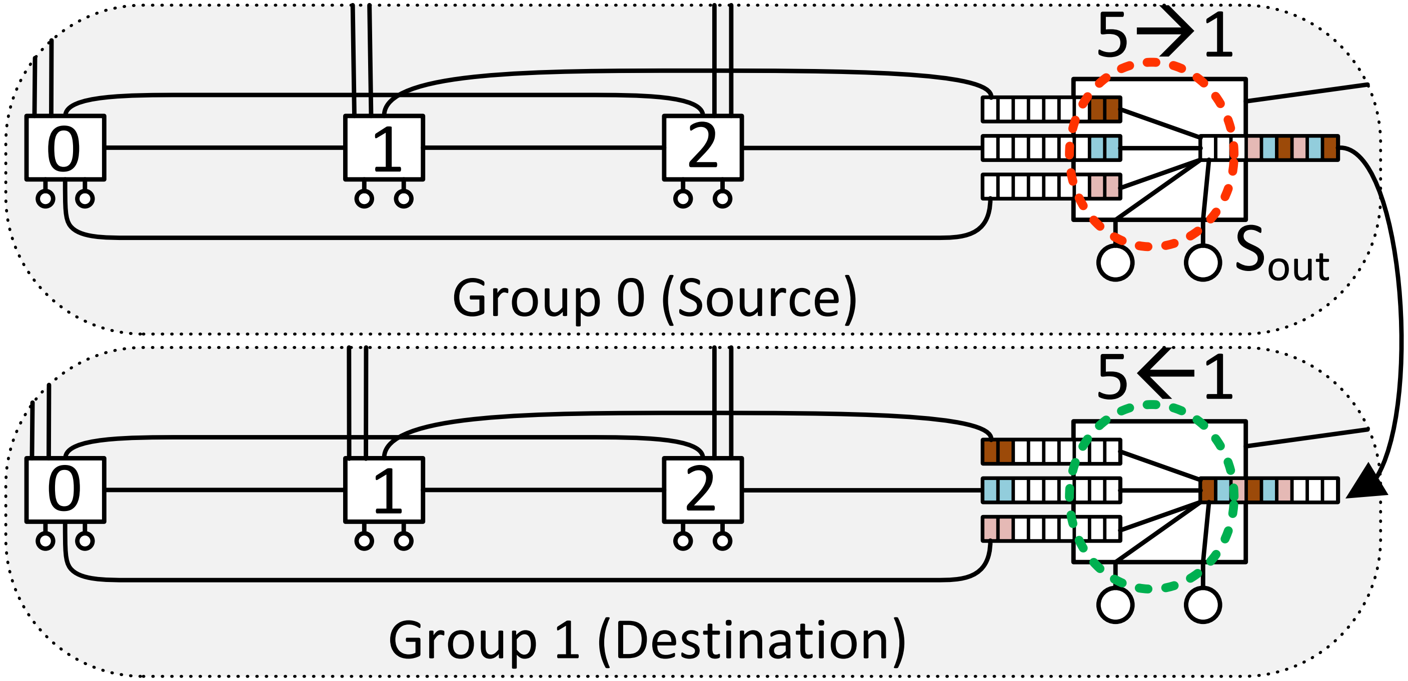
| Match pattern | Condition | Priority | Action |
|---|---|---|---|
| mac dst=host A | | High | outport=1 |
| mac dst=host B | | High | outport=2 |
| | | | |
| mac dst=group1 | N<P6Probability | High | output=6 |
| mac dst=group2 | N<P7Probability | High | output=7 |
| mac dst=group3 | N<P3Probability | High | output=3 |
| mac dst=group4 | N<P3Probability | High | output=3 |
| mac dst=group5 | N<P4Probability | High | output=4 |
| mac dst=group6 | N<P4Probability | High | output=4 |
| mac dst=group7 | N<P5Probability | High | output=5 |
| mac dst=group8 | N<P5Probability | High | output=5 |
| | | | |
| in_port=1 (host A) | | Low | output=6 |
| in_port=2 (host B) | | Low | output=7 |

Random number (N): 77

Architecture of SW0 in G0 with base QCN-SW proposal for a Dragonfly network.

# 2.1 QCN-SW + Source processing

- **Problem:** Unfairness because Sout does not receive CNMs

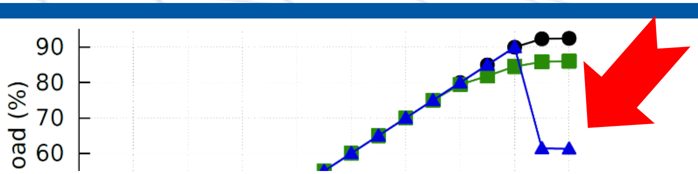


Representation of adversarial traffic pattern in a Dragonfly network.

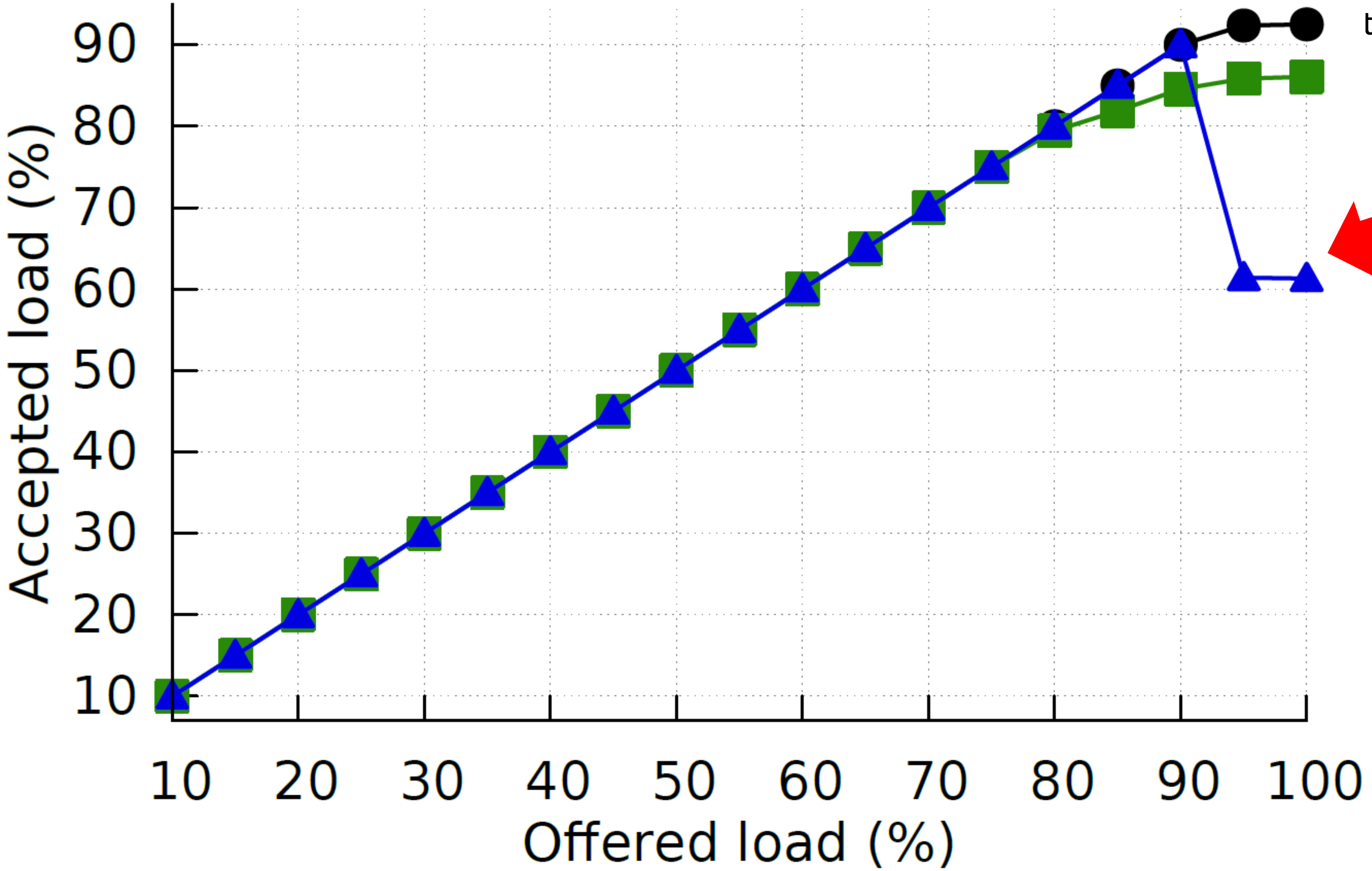
- **Proposed solution:** QCN-SW + source-processing
  - Add source-processing of CNMs generated by Sout
  - Switches snoop their own generated CNMs and change their routing table

# 2.2 QCN-SW + Feedback comparisson

- **Problem:** Throughput drop under *Uniform* traffic at high load



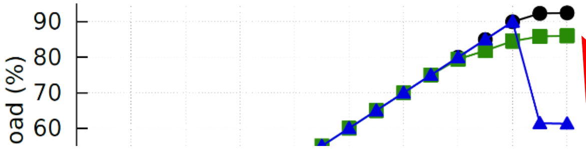
- **Proposed solution:** QCN-SW + feedback comparison



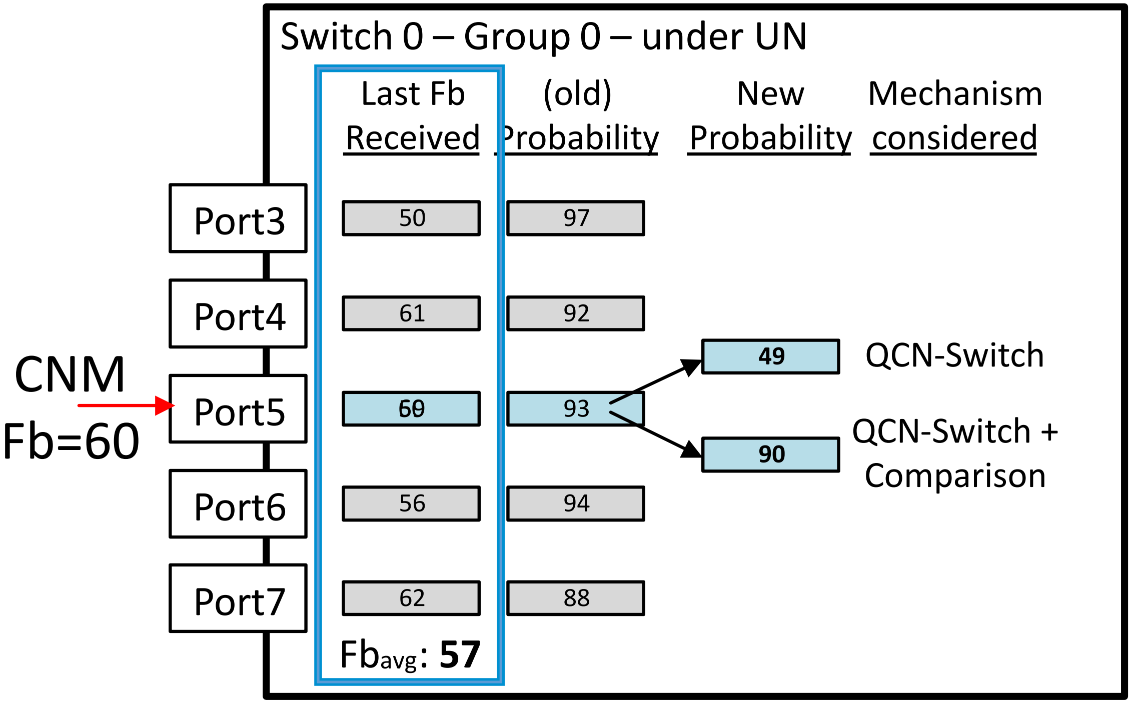
tch

under uniform traffic for a switch in a Dragonfly network.

# 2.2 QCN-SW + Feedback comparisson



- ❑ **Problem:** Throughput drop under *Uniform* traffic at high load
- ❑ **Proposed solution:** QCN-SW + feedback comparison
  - Add an average feedback value which represents the average congestion of ports of a switch
  - When a CNM is received, Fb value is compared with this Fbavg and if:
    - Fb < Fbavg → Probability is increased as in base mechanism
    - Fb > Fbavg → Probability is reduced by R= 1 – Lf * (Fb – Fbavg)



Switch 0 – Group 0 – under UN

CNM
Fb=60

| | Last Fb Received | (old) Probability | New Probability | Mechanism considered |
|---|---|---|---|---|
| Port3 | 50 | 97 | | |
| Port4 | 61 | 92 | | |
| Port5 | 60 | 93 | 49<br>90 | QCN-Switch<br>QCN-Switch + Comparison |
| Port6 | 56 | 94 | | |
| Port7 | 62 | 88 | | |

$Fb_{avg}$: **57**

Sample update of probability values when a CNM with Fb equal to 60 arrives, under uniform traffic for a switch in a Dragonfly network.

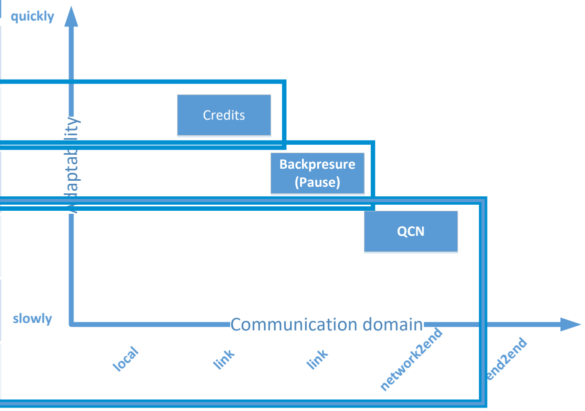
# Overview

1. Introduction
   1. Ethernet & Dragonfly
   2. Previous work – Conditional flow rules
   3. Congestion control indicators
   4. Quantized Congestion Notification
2. Our proposal: QCN-SW
   1. QCN-SW + Source processing
   2. QCN-SW + Feedback comparison
3. Evaluation
4. Conclusions & Future work

# 3. Evaluation

| Network parameters | | |
|---|---|---|
| Dragonfly topology | Input-Output Queue Switch | 16 Ports @40 Gbps |
| 1056 hosts | Packet Size = 1Kbyte | Switch Latency=200 ns |
| 4 CoS levels | Local/Global link latency=40/400 ns | QCN CP sampling at input queues [1] |

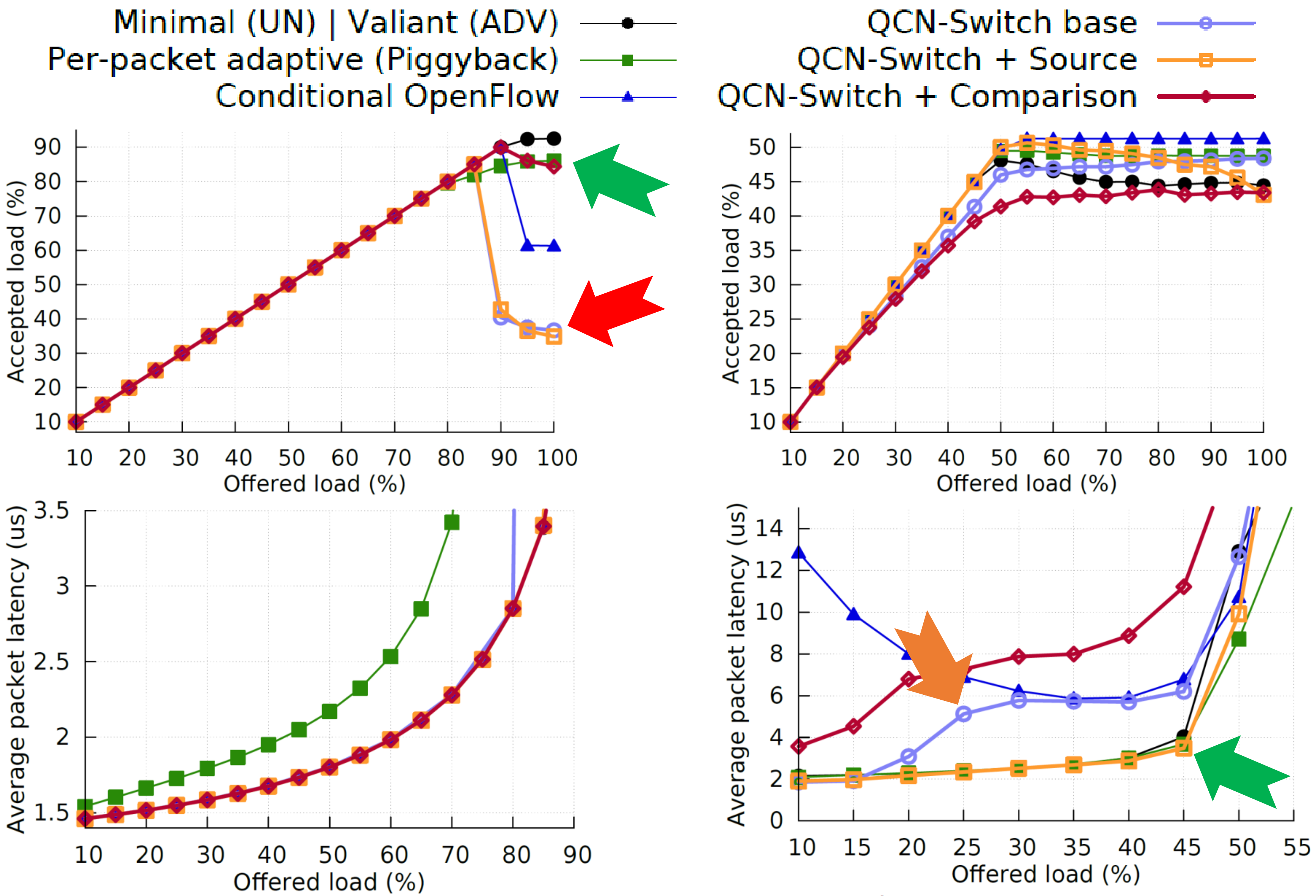
| Routing algorithms | |
|---|---|
| Minimal (UN) / Valiant (ADV) | Oblivious → No congestion estimation |
| Adaptive piggyback [2] | Credits |
| Conditional OpenFlow | Backpresure (Pauses) |
| QCN-Switch base | QCN CNMs |
| QCN-Switch + Source processing | QCN CNMs |
| QCN-Swtich + Feedback comparison | QCN CNMs |



[1] F. D. Neeser et al., "Occupancy sampling for terabit cee switches" HOTI'12

[2] N. Jiang et al, "Indirect adaptive routing on large scale interconnection networks," ISCA'09

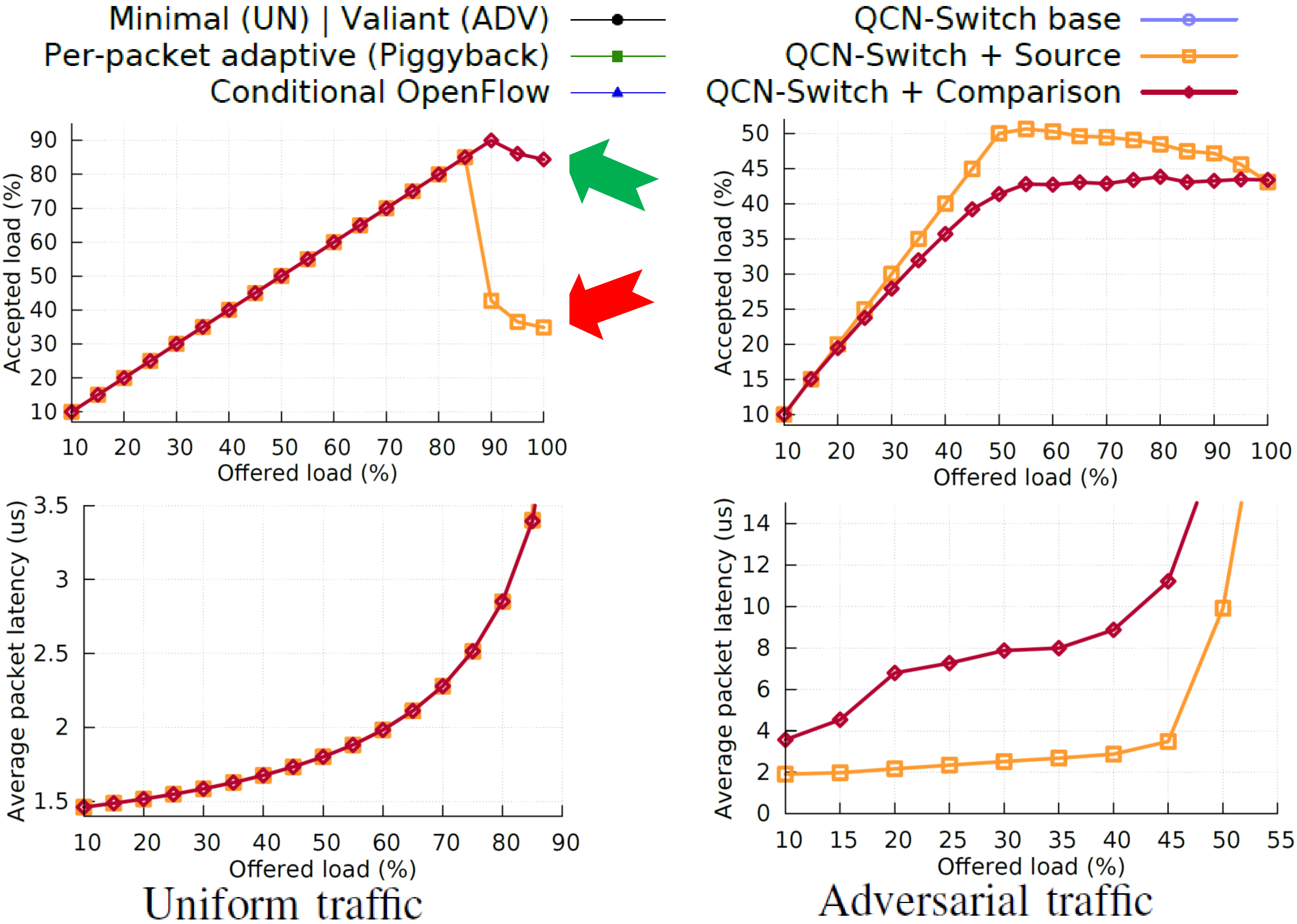
# 3. Evaluation



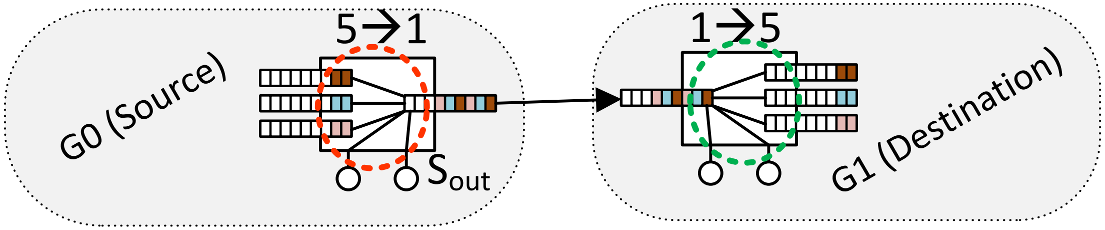
Uniform traffic

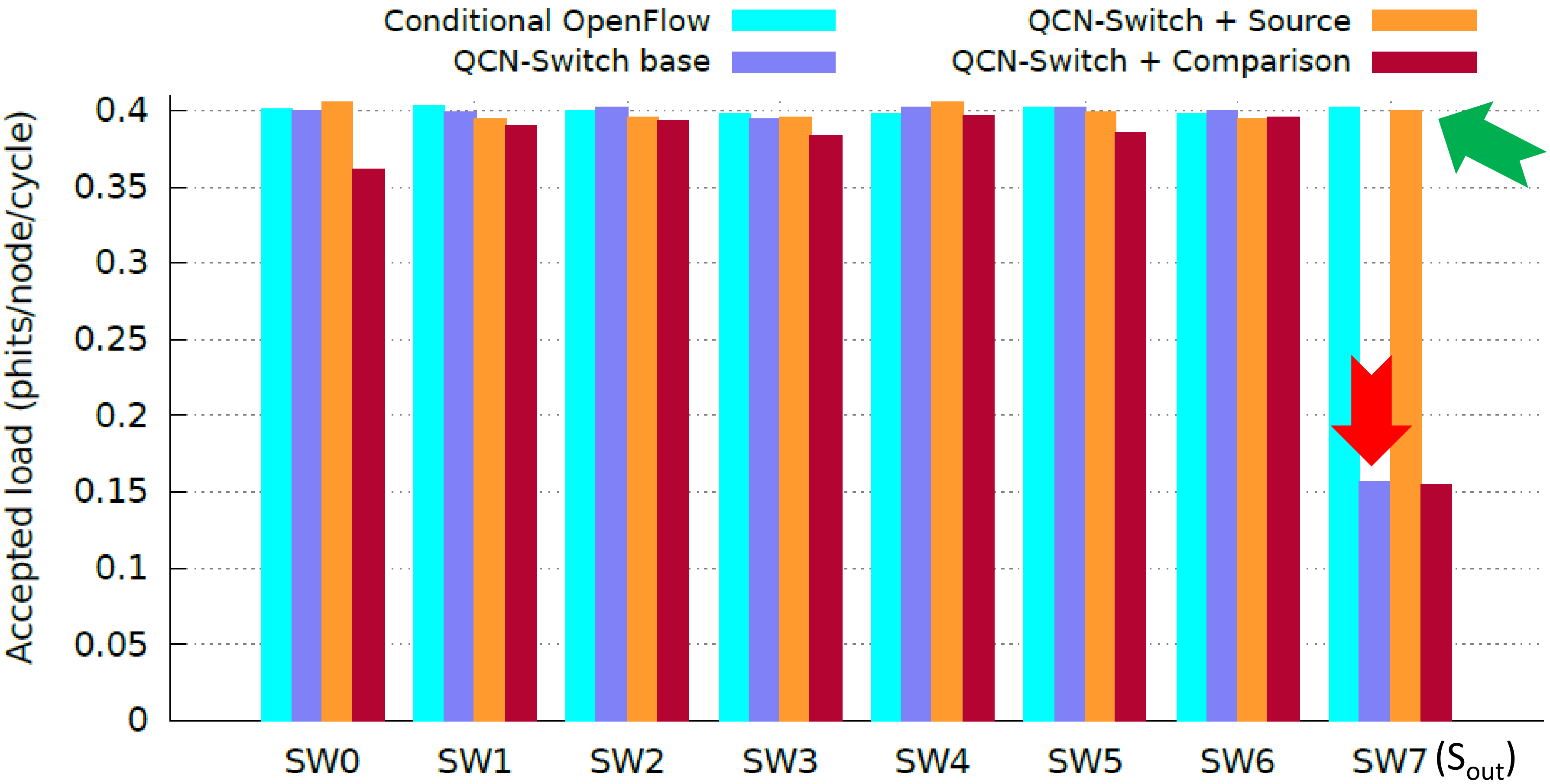
Adversarial traffic

# 3. Evaluation



# 3. Evaluation





Throughput accepted in each switch of group 0 for different routing mechanisms under ADV traffic with load 0.4 phits/node/cycle.

# 4. Conclusions and Future Work

## Conclusions:

❑ Our proposal relies on:

- Conditional OpenFlow rules
- QCN Congestion Notification messages
- Per-port probability to avoid oscillations

❑ Leveraging QCN information to build a non-minimal adaptive routing is not trivial:

- Identify two problems our base implementation
- Propose a solution for each problem with an add-on mechanism
- The on-going results in isolation of two add-ons proposed are good

❑ Exploring them in isolation allows us to identify their impact individually and clearly

## Future work:

❑ Define a mechanism joining QCN-SW base + source processing + feedback comparison

❑ Implement QCN Sampling at output buffers + feedback comparison

❑ Different policies for the increase and decrease probability should be analyzed

# Extending commodity OpenFlow switches for large-scale HPC deployments

Thank you!

**Mariano Benito**[†]
Enrique Vallejo[†]
Ramón Beivide[†]
Cruz Izu[‡]

✉ mariano.benito@unican.es

@m1n0x88

HiPINEB'17, 5 February 2017

# Extending commodity OpenFlow switches for large-scale HPC deployments

## BACKUP SLIDES