Extending commodity OpenFlow switches for large-scale HPC deployments

Mariano Benito, Enrique Vallejo, Ramón Beivide
University of Cantabria
Santander, Spain
{mariano.benito, enrique.vallejo, ramon.beivide}@unican.es

Cruz Izu
The University of Adelaide
Adelaide, Australia
cruz.izu@adelaide.edu.au

***Abstract*—Commodity Ethernet networks are used in many HPC systems. Extensions based on OpenFlow have been proposed for large HPC deployments, considering scalability and power consumption concerns. Such designs employ low-diameter topologies to minimize power consumption, such as Flattened Butterflies or Dragonflies. However, these topologies require non-minimal adaptive routing to deal with varying traffic characteristics and avoid pathological behaviors. The solutions to this issue in previous work relies on Ethernet Pauses to adapt minimal or non-minimal routing, depending on the availability (Pause status) of each corresponding output port. Nevertheless, such design provides an undesired high average latency under adversarial traffic patterns and a reduction in peak throughput under uniform traffic.**

This paper identifies the causes of the issues presented above, and presents a preliminary study of alternative solutions based on exploiting commodity congestion notification messages (QCN, 802.1Qau), currently available in Datacenter switches. This work presents the main differences between a congestion control mechanism such as QCN, which performs injection throttling reducing average network load, and an adaptive routing mechanism, which diverts traffic away from the congested area but increases average network load. In particular, it identifies the difficulty of separating the cases of uniform traffic at saturation and adversarial traffic at low loads.

I. Introduction

Ethernet is used in many Datacenters and High-Performance Computing (HPC) systems as the system-level interconnection technology. In the latest Top500 list [1], more than 40% of the supercomputers employ it. High-performance communication technologies require a single Layer-2 Ethernet domain, such as Open-MX [2] and RDMA over Converged Ethernet [3] (RoCE). Ethernet's large economy of scale [4], the availability of simple whitebox switches [5], the possibility of lossless implementations [6], and the ubiquity of Ethernet NICs in SoCs all suggest Ethernet technology will remain as a cost-effective alternative for HPC interconnection.

Multiple topologies have been proposed in recent years for HPC and Datacenter networks, seeking to minimize latency, cost and power consumption. A Dragonfly network comprises multiple groups of switches interconnected in a hierarchical direct topology [7]. These groups are formed by a number a of switches with p nodes per switch, directly connected by means of local links, and different groups are connected by global links (h global links per switch), using a local and global topology respectively. Minimal power consumption and a low 3-hop diameter are obtained when using complete graphs for both local and global topologies, as used in PERCS [8]; this will be the configuration considered in this work.

Adaptive non-minimal routing protocols select between minimal or non-minimal routing in response to traffic conditions. Multiple protocols have been proposed in the past, such as UGAL [7], Piggyback [9] and Opportunistic Local Misrouting (OLM) [10]. As all these approaches estimate per-switch congestion based on its buffer-to-buffer credit count, they cannot be implemented in commodity Ethernet switches.

Previous work in [11] has introduced the first Dragonfly design based on commodity Ethernet OpenFlow switches with a hierarchical MAC address rewriting mechanism to support hierarchical routing. Due to the lack of buffer occupancy information in commodity Ethernet switches, the adaptive routing decision relies on proactive conditional flow rules based on the Pause status of the preferred (minimal) port. This simple non-minimal adaptive routing mechanism suffers performance limitations, such as high latency and throughput drops, which will be described in detail in Section II.

The recent Quantized Congestion Notification standard (QCN, [12]), aimed at converged Datacenter infrastructures, introduces congestion notification messages at the Ethernet level. By snooping on these messages, we can obtain finer-grained congestion information compared to Pauses. Such information could help us improve the mechanisms that select minimal or non-minimal paths.

This paper presents on-going work that seeks to use QCN standard to support non-minimal adaptive routing in Dragonfly networks built from commodity off-the-shelf components. While this paper does not provide a complete solution to the problem studied, it does contain the following contributions:

- We identify the sources of high latency and reduced throughput in Dragonflies when relying on Pauses, namely an excessive rate of minimal routing for some nodes and a positive feedback loop at high loads that increases the use of non-minimal paths.
- We present three approaches for non-minimal adaptive routing mechanisms that exploit QCN notifications to adapt the minimal and non-minimal transmission rates.
- We present an early evaluation of the proposals, observing that the two most elaborated mechanisms independently improve the throughput and fairness of the results.

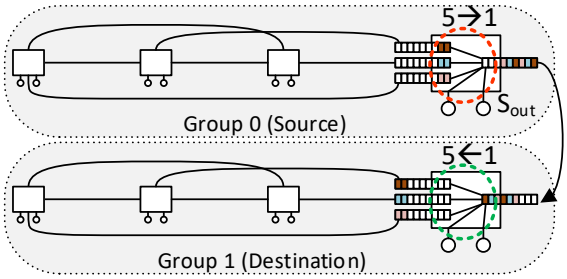


Fig. 1. Representation of adversarial (ADV) traffic in a ($p = 2, a = 4, h = 2$) Drangonfly network. All the traffic from Group 0 goes to Group 1, making the global link that joins them a bottleneck. Since traffic is quickly distributed in the destination group, this link does not pause. Buffers in local ports in S_{out} eventually get full, so these links are paused.

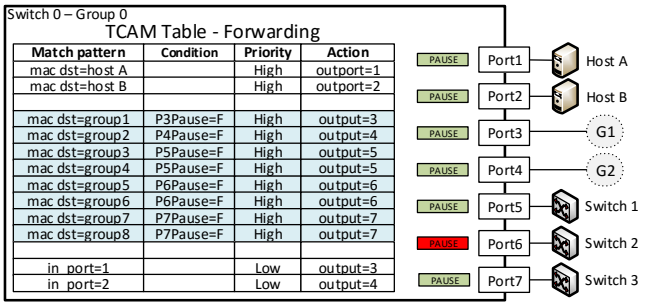


Fig. 2. Architecture of Switch 0 in Group 0 with conditional flow rules from [11] for a ($p = 2, a = 4, h = 2$) Dragonfly network. When the condition fails (when the output port is paused) the high-priority minimal routing rule is ignored, leading to the use of a low-priority rule.

The remainder of this paper is organized as follows: first, we describe background and motivation in Section II. Then, in Section III we present our design, which will be evaluated in Section IV. Some related work will be presented in Section V, concluding this paper in Section VI.

II. Background and Motivation

This section will introduce the requirements for implementing non-minimal adaptive routing in Dragonfly networks, and will describe the first attempt to do so using Pauses, identifying the shortcomings of that approach. This analysis will drive our new implementation using QCN.

A. Dragonfly topology and routing

Minimal routing in the Dragonfly is hierarchical, first to the destination group (a local hop, followed by a global hop), and then making a local hop to the destination switch. Under *uniform* (UN) traffic patterns, in which the destination of each frame is any terminal in the network, minimal routing provides optimal throughput and latency. When all the traffic from nodes in a group is sent using minimal routing to the same destination group, the only global link between these two groups, denoted as S_{out} in Figure 1, becomes the bottleneck. Such *adversarial* (ADV) traffic patterns should use non-minimal paths to avoid limiting network performance.

Valiant routing [13] randomizes network traffic by sending each packet first to a random intermediate network switch, and then to its final destination. This doubles average network distance and halves maximum throughput, but makes any traffic pattern behave as uniform traffic. Therefore, it improves network throughput for adversarial traffic patterns when compared to minimal routing.

Non-minimal adaptive routing adjusts to network conditions by alternating the use of minimal and non-minimal paths: when congestion is low most packets are sent using minimal paths. However, as congestion increases in the minimal paths, a higher percentage of messages should be diverted to less congested non-minimal paths.

B. Base design using Commodity Ethernet in HPC

Our previous work in [11] extended OpenFlow switches to support non-minimal adaptive routing in low-diameter topologies. Due to the lack of detailed congestion information such as buffer credits in Ethernet, routing decisions relies on the Pause status of the minimal port. Note that congestion control in commodity networks has been traditionally implemented as part of TCP/IP [14].

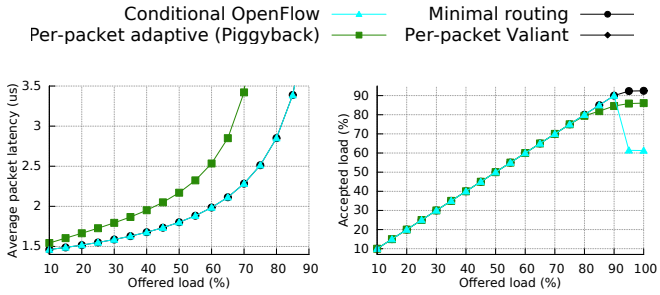
In that previous work, source adaptive routing, which we denoted *Conditional OpenFlow*, is implemented locally in each switch without contacting the remote controller, by extending OpenFlow rules to support conditions. Condition codes rely on the "Pause" status of each output port, as depicted in Figure 2. In this example, port 6 is paused, so the associated high-priority conditional rules that forward towards this port are disabled, and then, traffic is forwarded non-minimally using a low-priority rule.

C. Performance and limitations of the base design

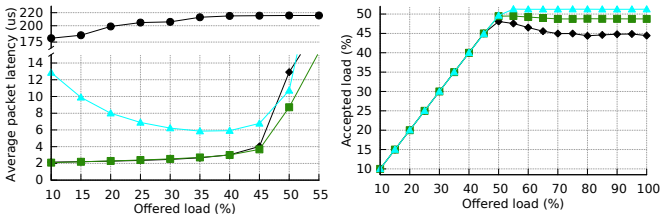
Figure 3 presents latency and throughput results from this approach[1] under both UN and ADV traffic patterns. The base design exhibited two performance issues which we will examine next in detail.

1) Throughput drops under UN traffic at saturation: the drop shown in Figure 3a is caused by a "positive control loop" introduced by adaptive non-minimal routing. When UN traffic reaches saturation, many ports get paused due to network congestion; such Pauses disable the minimal-forwarding rules, so traffic is injected non-minimally in those switches. Such non-minimal forwarded packets increase the amount of network traffic by using longer paths, which in turn further increases the amount of paused ports until most of the traffic is forwarded non-minimally. This problem arises because of the difficulty of differentiating the cases of UN traffic at saturation and ADV traffic (even at low levels) analyzing only the Pause status of the first-hop output port.

[1]The parameters used to obtain these results are detailed in Section IV-A, and slightly differ from those used in [11]; in any case, the same trends are observed.



(a) Average Latency and Throughput under uniform (UN) traffic.



(b) Average Latency and Throughput under adversarial (ADV) traffic.

Fig. 3. Average latency and throughput of the base *Conditional OpenFlow* mechanism under uniform (a) and adversarial (b) traffic patterns.

2) High average latency under ADV traffic: average latency values for the base approach are high compared to either Valiant or Piggyback, as shown in Figure 3b, particularly at low loads. In the base mechanism, traffic is sent minimally as long as a Pause is not received, this is, as long as there is some free space in the next buffer of the minimal path. This implies that the minimal path to the destination group is always saturated, and this minimally routed traffic suffers a very high latency.

The amount of traffic that can be forwarded minimally is very low; in the configuration used in this paper it is lower than 10% and it decreases with the network size. The rest of the traffic is forwarded non-minimally, with a much lower latency. As the offered load increases, the amount of traffic forwarded non-minimally increases, while the amount of minimal traffic remains constant. For this reason, *average* latency decreases with the traffic load, because the weight of non-minimal traffic increases. A similar effect has been observed with other routing mechanisms, such as UGAL with large buffers [7].

This latency issue appears under any adversarial traffic that merges multiple flows into one or a few global links. Keep in mind these high latency values at low loads are still 10 to 20 times lower than when all flows were using their minimal paths.

D. Quantized Congestion Notification

Quantized Congestion Notification (QCN, [12]) implements congestion notification in Layer-2 Ethernet Datacenter Networks. It is mainly composed of two elements, congestion points (CP, in switches) and reaction points (RP, rate limiters at NICs). Congestion points generate explicit Congestion Notification Messages (CNMs) when a given buffer suffers a

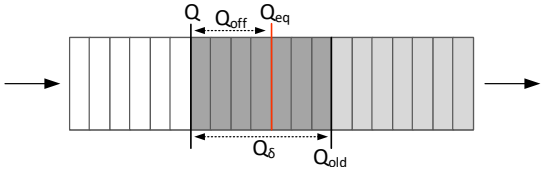


Fig. 4. QCN feedback calculation at congestion point.

congestion situation. To characterize this, two state variables are combined; position (Q_{off}) and velocity (Q_δ).

Figure 4 illustrates the feedback calculation mechanism. Each buffer is assigned a reference length denoted Q_{eq}, or equilibrium point. Q denotes the instantaneous buffer length sampled every 100 packets and Q_{old} denotes the buffer length when the last CNM was generated. Then, a feedback value (Fb) is calculated combining both values according to the following expression:

$$Fb = -(Q_{off} + w \times Q_\delta)$$

where $Q_{off} = Q - Q_{eq}$, $Q_\delta = Q - Q_{old}$, and w is a constant weight value set to 2 in the baseline implementation.

The generated CNM containing the Fb value quantized to 6 bits will be sent to the source of one packet sampled in the switch buffer (a source NIC). NICs implement injection throttling, based on an Additive-Increase, Multiplicative-Decrease (AIMD) policy. When (negative) congestion notifications are used, the feedback value Fb is used to divide the injection rate, adjusted by a factor L_f. QCN does not implement positive notifications (lack of congestion), so the additive increase policy is applied when no notifications are received for a period corresponding to 100 frames.

III. Design using QCN

Based on the concerns from Subsection II-C in this section we introduce a source adaptive routing based on Conditional OpenFlow rules which take advantage of QCN messages for deciding between minimal and non-minimal path for a packet.

Our design extends the basic conditional OpenFlow rules introduced in [11] and presented in Section II. In other previous mechanisms [7], [9], [10], routing is adapted on a packet-by-packet basis. However, this is not possible when relying on QCN, because the temporal granularity of congestion notifications is much larger than packet transmission time, so it would generate abrupt oscillations of traffic. Therefore, instead of using a single flag to enable or disable each conditional rule, the new approach considers a probability of sending minimally for each transit port (local and global) of a switch, as depicted in Figure 5. The goal of this mechanism is to balance the use of different paths depending on the congestion status of the network paths.

To implement this mechanism, a random value (N) is used for each table lookup; this value can be generated using a pseudo-random sequence which is updated periodically, or generated from the incoming frame (for example, by selecting some bits from its CRC). If the random number exceeds the

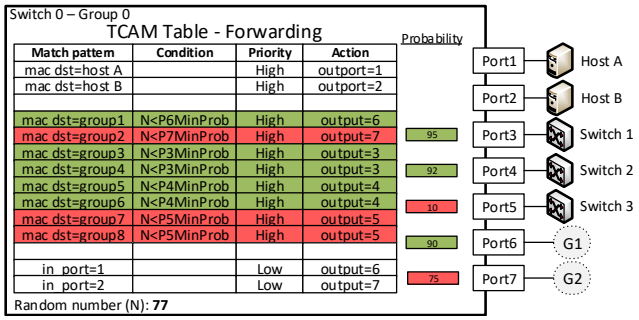


Fig. 5. Switch architecture with base QCN-Switch proposal for a ($p = 2, a = 4, h = 2$) Dragonfly network. When the condition fails (when N is lower than the probability of send minimally by associated port) the high-priority minimal routing rule is ignored, leading to the use of a low-priority rule. This diagram portrays Switch 0 of Group 0 under adversarial traffic pattern, rules highlighted in red are disabled because N is bigger than the probability of associated output ports.

probability of a matching conditional rule, this rule is not executed and a lower-priority rule is followed instead. With this mechanism, load can be balanced between minimal and non-minimal paths without relying on an estimation for their delays.

The probability of each conditional rule needs to be adapted to the congestion level in the corresponding path, which is estimated from the feedback values received in the QCN congestion notification messages. In our proposal, standard QCN Reaction Point would be disabled and switches intercept QCN notification messages (sent according to QCN standard by the network switches) and process them in order to update the probability associated to each forwarding rule. We have considered three different mechanisms for calculating the probability of each conditional rule, which are explained in the following subsections.

A. QCN-Switch base

This is the base reference in which the mechanism reduces the probability value associated to the output port when a CNM is received through it. An AIMD policy is used, modulated by the Feedback (Fb) value indicated in the message. Since QCN only contains negative congestion notification messages (but it does not notify the absence of congestion), we implement the protocol as follows:

- Upon reception of a CNM with feedback Fb, the probability value is reduced by a factor:

$$R = 1 - L_f \times Fb$$

  where L_f is a limiting factor that determines the extent to which the probability decreases, for example, with $L_F = 1 \,/\, 128$, R will be in the range between 0.5 and 1.
- We use a counter to keep the packets transmitted by each transit port between CNM messages. The counter is reset every time a CNM message is received and it is incremented for every transmitted packet. If it reaches a threshold PC_l, the probability value associated with that port is increased by $A_i\%$.

Tuning parameters L_f and A_i determines how quickly the mechanism reacts to changes in congestion. This base QCN-Switch mechanism only attacks the problem of queues being always full of minimal traffic, which is the cause of high latency under adversarial traffic as explained in Subsection II-C.

However, this mechanism introduces an unfairness problem because the switch S_{out} does not receive QCN notifications. Indeed, the receiving switch in the destination group quickly forwards received packets, so its buffer occupancy[2] and measured congestion level is very low. By contrast, queues associated to local input ports in S_{out} get full, generating CNMs for the rest of switches in the source group.

B. QCN-Switch + source-processing

The objective of this mechanism is to make S_{out} aware of the congestion associated to the minimal port under adversarial traffic, so its own traffic is forwarded non-minimally and unfairness is mitigated.

When a CNM is generated by a QCN Congestion Point in a switch due to a congestion situation in a *local* input port buffer, this is also processed in that same switch. Then, the feedback of this QCN message is used to decrease the probability of rules associated to the output port used to forward the "victim packet", which has been selected randomly from the (congested) input buffer by CP according to the QCN standard.

C. QCN-Switch + feedback comparison

This mechanism aims to prevent the switch reverting to non-minimal routing when *many* ports are congested, which is an indication of high-load uniform traffic (minimal routing should be used in that case for optimal performance).

This alternative maintains an average feedback value Fb_{avg} which represents the average congestion of transit ports of a switch. This value is calculated from the most recent feedback values $Last\ Fb_i$ received on each of its transit ports according to the following expression:

$$Fb_{avg} = \frac{\sum_{i=transit\ ports} Last\ Fb_i}{i}$$

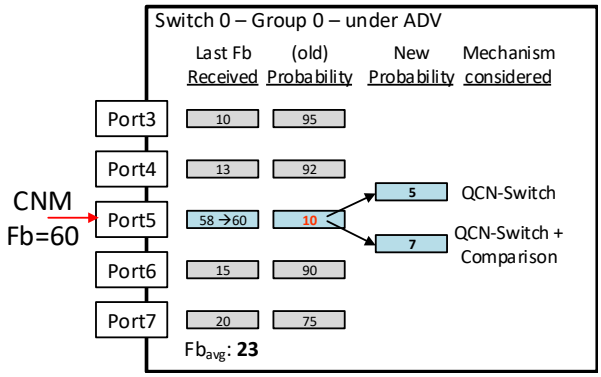
Upon reception of a QCN message, the switch recalculates Fb_{avg} and compares this average with the feedback received in the CNM. If the value just received is greater, the port's probability is reduced by the factor:

$$R = 1 - L_f \times (Fb - Fb_{avg})$$

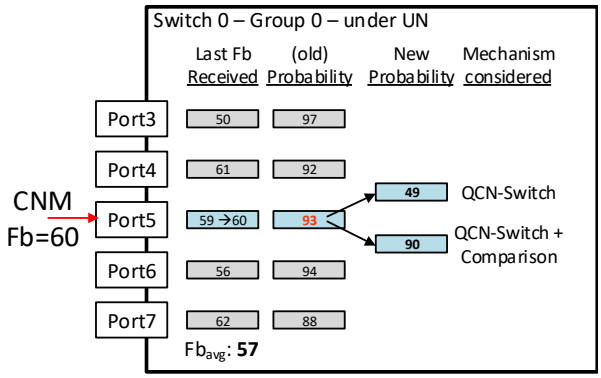
If Fb is lower, the probability associated to that port is increased by $A_i\%$, as in *QCN-Switch base*,

Figure 6 shows an example of the update of probabilities using both *QCN-Switch Base* or *QCN-Switch + feedback-comparison*, under uniform or adversarial traffic. The use of Fb_{avg} to calculate the reduction factor R in the latter reduces the impact of congestion notifications on uniform loads, when all ports experience similarly high congestion values.

[2]In our implementation, we model QCN CP sampling in the input ports, as detailed in subsection IV-A



(a) Adversarial traffic pattern.



(b) Uniform traffic pattern.

Fig. 6. Sample update of probability values when CNM with Fb equal to 60 arrives, under two different traffic scenarios for a switch in a ($p = 2, a = 4, h = 2$) Dragonfly network.

Obviously, a complete mechanism should include both QCN source-processing and feedback comparison, but in this work-in-progress paper we explore them separately to identify their individual impact.

IV. Evaluation

We have evaluated our proposal implementing it in a network simulator considering random uniform and adversarial traffic patterns for different injection loads.

A. Simulation tools

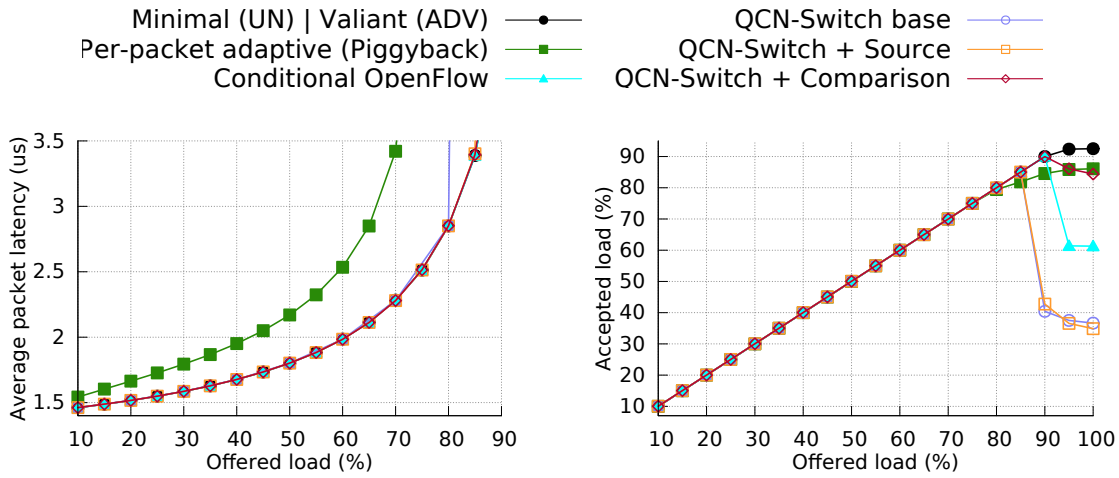
We employ the FOGSim network simulator [15] to evaluate the performance of non-minimal adaptive routing using the mechanisms proposed in Section III in a power-efficient Dragonfly network with $p = 4$ terminals per switch, $a = 8$ switches per group and $h = 4$ global links per switch. We have run a battery of simulations according to the explained network size and topology, leading to 1,056 terminals, which is representative of HPC systems. We select 1 KB for packet size as an intermediate value between minimum and maximum packet size for Ethernet technology. The cycle-accurate simulator models an input-output-buffered router. Four Ethernet CoS levels are considered, implemented as virtual channels, and used for deadlock avoidance and to prioritize QCN control messages. QCN detection points are implemented at the input ports of the switches, as suggested in [16]. Table I shows the network parameters, the QCN standard parameters and the parameters applied to the QCN-Switch as described in Subsection III-A. Note that the policies described in Section III are replacing QCN injection throttling at the NICs.

TABLE I
Simulation parameters.

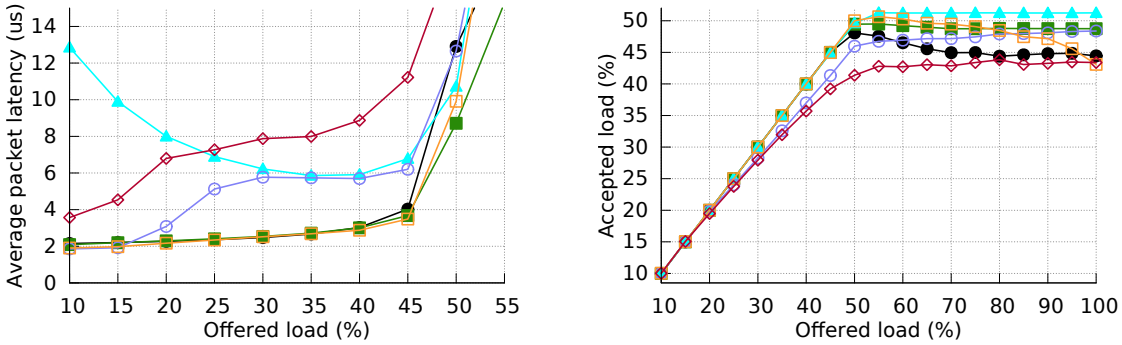
	Parameter	Value
Network configuration	Total end terminals	1,056 hosts
	Topology	Dragonfly
	Groups	33 groups
	Switches per group	8 switches
	Switch degree	15 ports
	Link speed	40 Gbps
	Packet size	1,000 bytes
	Switch frequency	1 GHz
	Switch latency	200 ns
	Local/Global link latency	40/400 ns (8/80 m)
	CoS levels	4
	Injection queues size	200 KBytes
	Transit queue size	100 KBytes
QCN	Queue's reference point (Q_{eq})	20% of queue size
	Weight value (w)	0
	Congestion point cycle	100 pkts.
QCN-Switch	Reduction limiting factor (L_f)	1 / 128
	Packet counter limit (PC_l)	100 pkts.
	% Probability increase (A_i) in *QCN-Switch base* and *QCN-Switch + Source*	1 %
	% Probability increase (A_i) in *QCN-Switch + Comparison*	10 %

We feed the network with synthetic traffic. Each node injects frames according to a Bernoulli process with a variable load, similarly to other network simulation experiments [17]. Two traffic patterns have been considered: Uniform (UN), in which the destination of each frame is any terminal in the network, and Adversarial (ADV), in which the destination of each frame is selected randomly between all nodes in the consecutive group. ADV traffic patterns concentrates the traffic on a single global link between two groups, so non-minimal routing is required to obtain a good performance.

We evaluate our non-minimal conditional routing based on QCN snooping with the three different mechanisms for updating minimal probability which we denote *QCN-Switch base*, *QCN-Switch + Source* and *QCN-Switch + Comparison*. Moreover, we employ the *Conditional OpenFlow* presented in our previous work [11] to compare to this new proposal. We use *Piggyback* routing (PB [9]) as an adaptive reference; this implements per-packet adaptive routing relying on state information for every global channel in its group distributed among switches. Note that this routing algorithm is *unfeasible* in Ethernet technology. Additionally, *Minimal* (MIN) and *Valiant* (VAL, [13]) routing are references for best response under UN and ADV traffic respectively.



(a) Average Latency and Throughput under uniform (UN) traffic



(b) Average Latency and Throughput under adversarial (ADV) traffic.

Fig. 7. Average latency and throughput of the three proposed mechanisms under uniform (a) and adversarial (b) traffic patterns.

B. Latency and throughput results

This section presents performance results for the three proposed mechanisms. Figure 7 shows average latency and throughput results under uniform and adversarial traffic patterns. These plots show results for the oblivious routing reference (minimal or Valiant), the adaptive Piggyback routing (which is not feasible using Ethernet technology), our previous work based on Ethernet Pauses (conditional-OpenFlow) and the three adaptive routing proposals based on QCN introduced in Section III.

Under uniform traffic, *Piggyback* sends part of the traffic non-minimally, which increases latency and reduces maximum throughput. By contrast, all the Ethernet-based mechanisms obtain optimal latency, similar to the reference *MIN*. Regarding throughput, the "positive control loop" that makes *Conditional OpenFlow* collapse at high loads remains and is even aggravated when using *QCN-Switch base* and *QCN-Switch + Source*. This occurs because the original mechanism based on Pauses sends traffic minimally as much as possible, whereas *QCN-Switch base* and *QCN-Switch + Source* detect CNMs and start to forward most traffic non-minimally. However, *QCN-Switch + Comparison* corrects this problem and presents good throughput at saturation; while there is a slight drop, it is relatively small and obtained throughput is competitive with the adaptive reference *Piggyback*.

In the context of ADV traffic, the previous approach *Conditional OpenFlow* presents an excessive average latency at low loads which has been already discussed in Section II-C. Our base proposal *QCN-Switch base* reduces this latency at low loads considerably, since it forwards a significant portion of traffic non-minimally thanks to the reception of QCN messages from the minimal path. At intermediate loads (20-25%), this latency increases because a larger ratio of traffic is forwarded minimally. *QCN-Switch + Source* improves this latency at intermediate loads, since it allows all switches in a group to detect congestion. This effect is studied in Subsection IV-C by focusing on throughput fairness. Unfortunately, the latency of the *QCN-Switch + Comparison* mechanism, which is the only one obtaining competitive throughput in Figure 7a, is worse than *QCN-Switch base*. The comparison with the average feedback value Fb_{avg} makes minimal routing more frequent, which increases average latency under adversarial traffic.

Throughput results under adversarial traffic are consistent with the latency results analyzed above. While *QCN-Switch + Comparison* gets a lower saturation throughput, the three

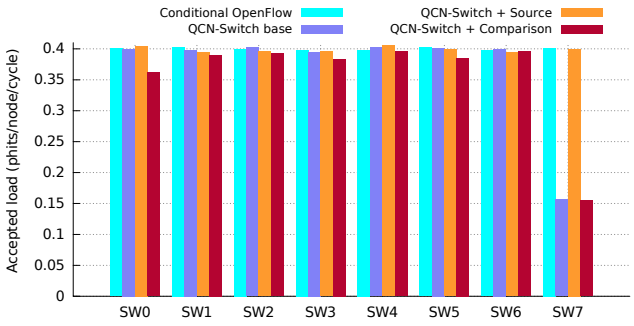


Fig. 8. Throughput injected in each switch of group 0 for different routing mechanisms under ADV traffic with load 0.4 phits/node/cycle.

alternatives are competitive with the previous proposals. *QCN-Switch + Source* obtains the maximum saturation throughput, and despite suffering some congestion at high loads, its throughput is competitive with the oblivious *Valiant*. The throughput before the saturation point is indicative of unfairness issues, which will be explored next.

C. Fairness

The throughput under ADV traffic in Figure 7b presents pathological effects caused by unfairness. Before reaching the saturation load, both *QCN-Switch base* and *QCN-Switch + Comparison* present a throughput value which is *lower* than the offered load of the simulation. This is observed easily in the slope of their throughput curves before saturation, lower than the expected 45º. This does not occur in the reference mechanisms, *PB, Valiant* and *Conditional OpenFlow*.

Figure 8 explores this problem in more detail, comparing the throughput obtained by each switch of Group 0 at load 0.4 phits/node/cycle. Switch SW7 corresponds to the switch S_{out} in Figure 1. Using *QCN-Switch base* or *QCN-Switch + Comparison*, SW7 injects significantly less traffic than the other switches of the group. As discussed in Section III, SW7 does not receive a significant amount of congestion notification messages from the global link, because all traffic received in the destination group is quickly dispatched to its destination switch, so the queues do not get full; recall that our QCN model implements the detection point at the input buffers.

As expected by the throughput slope, *QCN-Switch + Source* is the only mechanism that solves this unfairness, and all the switches inject a similar amount of traffic.

V. Related work

Explicit congestion notification has been used in commodity networks for a long time. IP has ECN bits for explicit congestion notification [14]. These bits are set when switch queues exceed a given threshold, possibly following a *marking* policy such as RED [18]. Datacenter TCP [19] estimates not only the presence, but the *amount* of congestion, based on the count of ECN flags measured during a given interval (typically, a TCP RTT). The differences between these mechanisms and QCN [12] are that QCN generates congestion notification messages at layer 2, and that these notifications include an integer feedback value; such feedback is computed from both the current queue occupancy and from the increase or decrease rate.

Minkenberg *et al.* [20] suggested for the first time the use of ECN congestion notifications to adapt traffic in Datacenters. Their proposal differs from our approach in two fundamental aspects. First, they do not consider non-minimal routing, with the increased load introduced by non-minimal paths and the associated positive feedback loop. Second, they do not consider a probability for each available path, nor a recovery mechanism to restore minimal routing when congestion dissapears. Instead, they consider fixed time intervals, and routing information is reset on each interval, discarding all the received congestion information and reverting to minimal traffic.

UGAL [21] selects dynamically between minimal routing or Valiant randomization, but it requires global information. More elaborate mechanisms improve this non-minimal routing decision, both at the source or in-transit [9], [10]. By contrast, our reference mechanism [11] implements source-adaptive routing and it employs a pre-calculated non-minimal intermediate destination per source node.

Unfairness occurs naturally in any network under non-uniform traffic patterns. Fuentes *et al.* present a study of unfairness issues in Dragonflies in [22]. They identify a novel traffic pattern, denoted ADVc, which causes the maximum unfairness. Employing such traffic pattern to evaluate our proposal will be considered in future work.

VI. Conclusions

This paper has identified the main limitations of a source-adaptive non-minimal routing mechanisms for commodity Ethernet networks, in particular an excessive amount of minimal traffic under adversarial traffic, some minor unfairness issues, and a "positive control loop" which causes a throughput collapse under uniform traffic.

These problems were caused by the naïve implementation of the base design, which relies on Ethernet Pauses. QCN 802.1Qau introduces explicit congestion notification messages in Datacenter Ethernet. However, as analyzed in the paper, leveraging this congestion control information to build a non-minimal adaptive routing mechanism is not trivial: the *base* approach explored suffers from unfairness, throughput drop and high latency. Two alternatives partially solve these problems in isolation, by comparing the feedback status of different ports (throughput drop), and by snooping congestion notification messages generated by the local switch (unfairness).

The current paper presents work in progress. Obviously, a complete proposal should avoid both pathological issues simultaneously, while providing competitive latency results. Considering the high interval between congestion notifications in Ethernet and the finer granularity of the congestion notification, different policies for the increase and decrease of each output port probability can be devised and analyzed. All these aspects are left for future work that completes the preliminary results presented in this paper.

ACKNOWLEDGMENT

This work has been supported by the Spanish Ministry of Education, Culture and Sports under grant FPU14/02253, by the Spanish Science and Technology Commission (CICYT) under contracts TIN2013-46957-C2-2-P and TIN2016-76635-C2-2-R and by the European HiPEAC Network of Excellence. The Mont-Blanc project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 671697. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

REFERENCES

[1] "Top500 supercomputer ranking," 2015. [Online]. Available: http://www.top500.org/

[2] B. Goglin, "High-performance message-passing over generic ethernet hardware with Open-MX," *Parallel Computing*, vol. 37, no. 2, pp. 85–100, 2011.

[3] Mellanox, "RoCE in the data center," Mellanox, Tech. Rep., 2014.

[4] B. Casemore, L. Rosenberg, R. Brothers, R. Costello, R. Mehra, P. Jirovsky, and N. Greene, "Worldwide enterprise communications and datacenter networks 2014: Top 10 predictions," IDC report, 2014.

[5] Open Compute Project Community, "Open compute project network specifications and designs," http://www.opencompute.org/wiki/Networking/SpecsAndDesigns, 2015.

[6] *IEEE Standard for Local and metropolitan area networks–Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks - Amendment 17: Priority-based Flow Control, 802.1Qbb*, IEEE Std., 2011.

[7] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-driven, highly-scalable dragonfly topology," in *International Symposium on Computer Architecture (ISCA)*, 2008, pp. 77–88.

[8] B. Arimilli, R. Arimilli, V. Chung, S. Clark, W. Denzel, B. Drerup, T. Hoefler, J. Joyner, J. Lewis, J. Li *et al.*, "The PERCS high-performance interconnect," in *18th Symposium on High Performance Interconnects*. IEEE, 2010, pp. 75–82.

[9] N. Jiang, J. Kim, and W. J. Dally, "Indirect adaptive routing on large scale interconnection networks," in *Intl. Symp. on Computer Architecture (ISCA)*, 2009, pp. 220–231.

[10] M. García, E. Vallejo, R. Beivide, M. Odriozola, and M. Valero, "Efficient routing mechanisms for dragonfly networks," in *The 42nd International Conference on Parallel Processing (ICPP-42)*, 2013.

[11] M. Benito, E. Vallejo, and R. Beivide, "On the use of commodity ethernet technology in exascale hpc systems," in *Proceedings of the 2015 IEEE 22Nd International Conference on High Performance Computing (HiPC)*, ser. HIPC '15. Washington, DC, USA: IEEE Computer Society, 2015, pp. 254–263. [Online]. Available: http://dx.doi.org/10.1109/HiPC.2015.32

[12] *IEEE Standard for Local and Metropolitan Area Networks–Virtual Bridged Local Area Networks - Amendment: 10: Congestion Notification, 802.1Qau*, IEEE Std., 2010. [Online]. Available: http://www.ieee802.org/1/pages/802.1au.html,

[13] L. Valiant, "A scheme for fast parallel communication," *SIAM journal on computing*, vol. 11, p. 350, 1982.

[14] K. Ramakrishnan, S. Floyd, and D. Black, "RFC3168: The addition of explicit congestion notification (ECN) to IP," 2001.

[15] M. García, P. Fuentes, M. Odriozola, E. Vallejo, and R. Beivide. (2014) FOGSim interconnection network simulator. [Online]. Available: http://fuentesp.github.io/fogsim/

[16] F. D. Neeser, N. I. Chrysos, R. Clauberg, D. Crisan, M. Gusat, C. Minkenberg, K. M. Valk, and C. Basso, "Occupancy sampling for terabit cee switches," in *2012 IEEE 20th Annual Symposium on High-Performance Interconnects*, Aug 2012, pp. 64–71.

[17] J. García-Haro, R. Marín-Sillué, and J. L. Melús-Moreno, *ATMSWSIM An efficient, portable and expandable ATM SWitch SIMulator tool*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 193–212. [Online]. Available: http://dx.doi.org/10.1007/3-540-58021-2_11

[18] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Netw.*, vol. 1, no. 4, pp. 397–413, Aug. 1993. [Online]. Available: http://dx.doi.org/10.1109/90.251892

[19] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, "Data center TCP (DCTCP)," in *Proceedings of the ACM SIGCOMM 2010 Conference*, ser. SIGCOMM '10. New York, NY, USA: ACM, 2010, pp. 63–74. [Online]. Available: http://doi.acm.org/10.1145/1851182.1851192

[20] C. Minkenberg, M. Gusat, and G. Rodriguez, "Adaptive routing in data center bridges," in *17th IEEE Symposium on High Performance Interconnects (HOTI)*, 2009, pp. 33–41.

[21] A. Singh, "Load-balanced routing in interconnection networks," Ph.D. dissertation, Stanford University, 2005.

[22] P. Fuentes, E. Vallejo, C. Camarero, R. Beivide, and M. Valero, "Network unfairness in dragonfly topologies," *The Journal of Supercomputing*, vol. 72, no. 12, pp. 4468–4496, 2016. [Online]. Available: http://dx.doi.org/10.1007/s11227-016-1758-z