Noname manuscript No.
(will be inserted by the editor)

Network Unfairness in Dragonfly Topologies

Pablo Fuentes · Enrique Vallejo · Cristóbal Camarero · Ramón Beivide · Mateo Valero

Received: date / Accepted: date

Abstract Dragonfly networks arrange network routers in a two-level hierarchy, providing a competitive cost-performance solution for large systems. Non-minimal adaptive routing (adaptive misrouting) is employed to fully exploit the path diversity and increase the performance under adversarial traffic patterns. Network fairness issues arise in the dragonfly for several combinations of traffic pattern, global misrouting and traffic prioritization policy. Such unfairness prevents a balanced use of the resources across the network nodes and degrades severely the performance of any application running on an affected node.

This paper reviews the main causes behind network unfairness in dragonflies, including a new adversarial traffic pattern which can easily occur in actual systems and congests all the global output links of a single router. A solution for the observed unfairness is evaluated using age-based arbitration. Results show that age-based arbitration mitigates fairness issues especially when using in-transit adaptive routing. However, when using source adaptive routing, the saturation of the new traffic pattern interferes with the mechanisms employed to detect remote congestion, and the problem grows with the network size. This makes source adaptive routing in dragonflies based on remote notifications prone to reduced performance, even when using age-based arbitration.

Keywords Dragonfly · fairness · networking

P. Fuentes, E. Vallejo, C. Camarero, R. Beivide
University of Cantabria
Tel.: +34-942-206772
E-mail: {pablo.fuentes, enrique.vallejo, cristobal.camarero, ramon.beivide}@unican.es

M. Valero
Barcelona Supercomputing Center and Universitat Politècnica de Catalunya
Tel.: +34-93-4137716
E-mail: mateo.valero@bsc.es

1 Introduction

Dragonfly networks are considered as one of the most promising network topologies for upcoming Exascale systems, and have been employed in the PERCS [6] and Cascade [5] system networks. Unfortunately, these networks easily suffer congestion under certain adversarial traffic patterns. To overcome network congestion and fully exploit path diversity, non-minimal adaptive routing mechanisms are required. These mechanisms employ an intermediate random node [24] to divert the traffic before sending minimally towards the destination, improving the utilization of the inter-group (*global*) links in the event of saturation in a link on the minimal path. A detailed view of the dragonfly topology and the routing mechanisms are presented in Section 2.

In a fair situation, all users receive the same service level regardless of their location. In this paper we will focus on per-node injection throughput unfairness, where source nodes receive a different service level from the network, being able to send different amounts of traffic. The causes of such unfairness are varied, including asymmetric topologies, non-uniform traffic patterns or even network faults. The impact of unfairness depends on the structure of the application; in the worst case of typical fork-join applications, the system can slow to the speed determined by the worst-serviced node.

The dragonfly network presents a balanced use of resources under uniform traffic. However, fairness problems appear under non-uniform traffic patterns, such as *adversarial* (ADV) and *adversarial-consecutive* (ADVc), which are detailed in Section 3. Under these traffic patterns, a single router in each group receives most of the ongoing traffic of the group if minimal routing is employed. Nodes connected to this router will have more difficulty to inject traffic and, therefore, receive a worse service from the network. The unfairness level experienced depends on the routing mechanism and, for adaptive strategies, the *global misrouting policy.* This policy defines the set of inter-group links which can be used to send non-minimal traffic to avoid a congested link. These adversarial traffic patterns and different global misrouting policies for both source or in-transit adaptive routing are detailed in Section 3.

Different global misrouting policies were first considered in [13]. Adversarial-consecutive traffic, which causes unfairness regardless the global misrouting policy, was first introduced in [10]. However, none of these works evaluated a network using an explicit mechanism to provide fairness among the nodes. Several explicit mechanisms to provide fairness in interconnection network have been introduced before [2,15]. This work extends the previous ones by reviewing the problem of unfairness in dragonflies and evaluates a solution based on age-based arbitration [2]. Such mechanism has been previously employed in system-level interconnection networks such as the Cray XT4 [1]. Interestingly, this solution is particularly effective for in-transit adaptive routing, but not so much for source adaptive routing. An analysis of the problem shows that the congestion notification mechanism employed for source routing fails to identify congested links under ADVc traffic.

In short summary, our main contributions are:

- We review the main aspects which cause unfairness in dragonfly networks for different routing mechanisms, in particular adversarial traffic patterns, global misrouting policies and in-transit traffic prioritization. Results conclude that explicit fairness mechanisms are required in these networks.
- We present an evaluation of dragonfly networks using age-based Arbitration for explicit fairness. Our results show that age-based arbitration is particularly effective providing fairness with in-transit adaptive routing, and provides fairly good results for source adaptive routing.
- We identify a limitation of the congestion-notification mechanisms employed in adaptive source routing, which become ineffective under ADVc traffic. The unfairness caused by this limitation grows with the network size, promoting the use of in-transit adaptive routing.

The rest of the paper is organized as follows. Section 2 presents some background on dragonfly networks and their routing mechanisms. Section 3 details the most relevant causes for network unfairness, and the age-based fairness mechanism evaluated in this work. Section 4 describes the simulation infrastructure and the evaluation methodology employed in this work. Sections 5 and 6 present the performance and fairness results with and without age-based arbitration. A discussion is presented in Section 7, including an assessment of the validity of the results for larger networks. Finally, Section 8 presents some related work and Section 9 concludes the paper.

2 Background: topology and routing in dragonfly networks

In this section we introduce a description of the dragonfly network and the different routing mechanisms proposed.

2.1 Dragonfly networks

The dragonfly [19] is a low-diameter network based on high radix routers. Routers in a dragonfly network are deployed in a two-level hierarchical layout, with fully-connected groups of routers conforming a virtual high-radix router. Such groups are connected on a second-level interconnection pattern. In this work, we focus on dragonfly networks with complete graphs in both hierarchical levels, denoted as *canonical dragonflies* in [7].

A canonical dragonfly network can be described using three parameters [19]:

- p is the number of nodes linked to every router.
- a is the number of routers per group in the first hierarchical level.
- h is the number of inter-group (*global*) links in each router, connecting with a router in a different group.

Additionally, the global link arrangement specifies the distribution of global links among the routers of each group; in this work we employ the *palmtree*

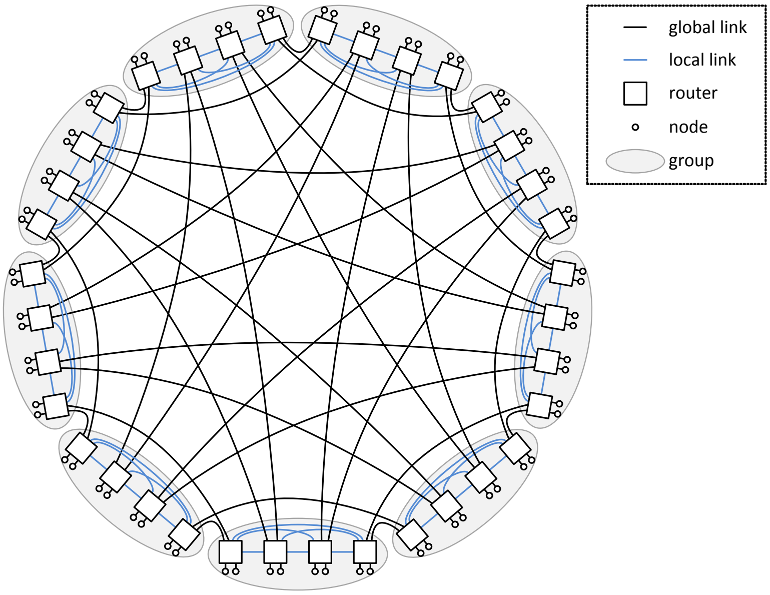


Fig. 1: Minimal-size dragonfly network with $h = p = 2$ (2 global links per router, 2 computing nodes per router) and $a = 4$ (4 routers per group).

arrangement [7], but the study and results for other arrangements are similar. An example dragonfly network is depicted in Figure 1 with $p = h = 2, a = 4$.

Performance in dragonflies is tightly connected to the pattern of communications and the routing mechanism. For random traffic patterns that stress uniformly the links in the network, the use of the shortest path between source and destination nodes provides sufficient performance in terms of throughput and latency. However, performance is severely affected under other traffic patterns with higher contention in the inter-group links, due to a poor use of the path diversity. For these cases, non-minimal routing mechanisms are required to achieve good performance.

2.2 Routing mechanisms

Several routing mechanisms have been proposed for the dragonfly network [19, 18, 14, 11, 7]. In this work we classify them in three categories: oblivious, source-based adaptive, and in-transit adaptive routing. Oblivious routing selects a path at injection which is independent of the current status of the network, whereas adaptive routing mechanisms react to congestion to improve network performance. Source-based adaptive routing selects between multiple paths at injection, depending on a decision which is typically based on a direct or indirect measure of the network congestion. By contrast, in-transit adaptive routing can switch between minimal and non-minimal paths at injection and along the route, what avoids the need for indirect congestion measures.

The routing mechanisms which have been selected to model these routing classes are described next. We highlight in bold the particular mechanisms which are implemented in the evaluation, as described in Section 4. In all cases a virtual-channel-based deadlock-prevention mechanism is employed, as detailed in Section 2.3.

2.2.1 Oblivious routing

Several oblivious routing mechanisms are employed as a reference, depending on the traffic pattern.

Minimal routing (***MIN***) is the reference for random uniform traffic. It delivers traffic through the shortest path, employing up to three hops (one *local* and one *global* link to reach the destination group, and one *local* link to arrive to the destination node, a *lgl* route).

For adversarial traffic patterns (ADV when all traffic from a source group is sent to the same destination group, and ADVc as introduced in Section 3.2.2), nonminimal routing is required to avoid the congested links. In this case, Valiant routing (*VAL*, [24]) can be used to send traffic non-minimally. It selects a random intermediate node between the source and the destination to divert the traffic through longer routes. These longer paths will be less congested than the minimal path under adversarial traffic patterns. Valiant requires up to six hops to complete the network traversal, three to the intermediate node (*lgl-*) and three from the intermediate node to the destination (*-lgl*).

In the original definition of Valiant, the intermediate node is selected randomly between all nodes in the network at packet generation time. In our case, we have implemented two related nonminimal oblivious routing variants according to the global misrouting policies introduced in Section 3.1: ***Oblivious-RRG*** is similar to Valiant, since it selects the intermediate destination completely randomly. By contrast, ***Oblivious-CRG*** modifies the initial selection of the random intermediate node, restricting it to nodes in groups directly connected to the source router. This saves the (frequent) first local hop, but restricts the amount of random intermediate nodes.

2.2.2 Source-based adaptive routing

We employ PiggyBack (*PB*, [18]) as a source adaptive routing mechanism. It estimates the congestion of the network and selects between *VAL* and *MIN* routing at packet injection depending on the saturation status of the minimal link. A link is considered as saturated when its associated credit count exceeds a given threshold, relative to the status of the other nodes. The saturation status information is shared across the routers in the same group, in a sort of Explicit Congestion Notification (ECN).

As in the previous case, we have implemented two variants of source-based adaptive routing, depending on the use of Oblivious-CRG or Oblivious-RRG for the selection of the nonminimal path. We denote these two variants as ***Source-based-CRG*** and ***Source-based-RRG*** respectively.

2.2.3 In-transit adaptive routing:

Our implementation applies in-transit global and local misrouting. Global misrouting (sending traffic to a non-minimal group) can be selected at injection or after a first hop in the source group as in *PAR* [18]. Selection relies on the number of credits of the output ports in the current router. At intermediate or destination groups, local misrouting (sending traffic to a non-minimal router in the same group) is used if the links from the minimal path are considered saturated. This avoids pathological performance issues identified in [14, 25].

We have implemented the three variants of global misrouting policy introduced in Section 3.1, and denoted them ***in-transit-CRG***, ***in-transit-RRG*** and ***in-transit-MM*** respectively.

2.3 Deadlock-avoidance mechanisms

Our evaluation considers a lossless networks where all the packets are delivered to their destinations. To avoid packet drop and impede deadlock from stalling the network, a deadlock-avoidance mechanism is required. Oblivious and source-based adaptive routing mechanisms rely on the use of virtual channels (VCs) in an incremental fashion for every hop across the packet path, as employed in [19] and [6]. Aside from deadlock prevention, the use of multiple virtual channels mitigates Head-of-Line (HoL) blocking. It can be observed that only certain hops of the path will be conducted in each type of link (local or global), reducing the amount of VCs below the maximum path length, to the maximum number of hops per link type.

In our evaluations, the amount of VCs has been restricted to the lowest amount required to prevent deadlock. *MIN* routing requires only 2/1 VCs (2 VCs in local links, 1 in global links) to avoid deadlock, corresponding to the three hops in the longest path. Similarly, our oblivious and source adaptive implementations require 4/2 VCs.

In-transit adaptive routing employs Opportunistic Local Misrouting *OLM* [11] to minimize the cost of the implementation by reducing the required number of VCs for non-minimal local hops, lowering the total amount to 3/2 VCs.

3 Unfairness in dragonfly networks

In this section we review three aspects which have been identified to decrease fairness in dragonfly networks: global misrouting policy, adversarial traffic patterns and prioritization to in-transit traffic. We also present the fairness mechanism which has been evaluated in Section 6.

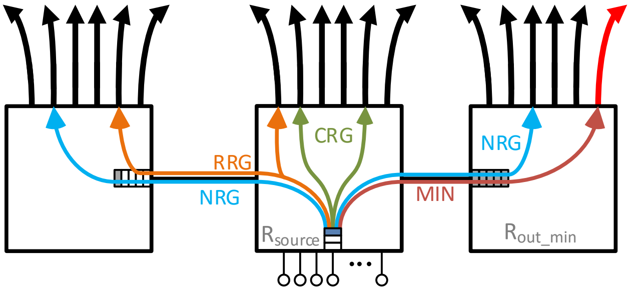


Fig. 2: Global misrouting policies for source routing.

3.1 Global misrouting policies

A remote group can be directly or indirectly connected to a given router in the dragonfly network. When a group is directly linked to the current router, only one global link needs to be traversed to reach such group. Arriving to an indirectly linked group implies traversing another router in the current group, requiring two hops: one local link from the current router to the neighbor router which is connected to the destination group, and one global link between the two groups (lg).

The *global misrouting policy* defines the intermediate group in non-minimal paths, depending whether it is a directly or indirectly connected group from the current router. In general, three different global misrouting policies can be considered for source-based adaptive routing:

- ***Random-router Global, (RRG)***: the intermediate group is selected randomly across the network, regardless of its distance from the current router.
- ***Current-router Global, (CRG)***: only those groups that are directly linked to the current (source) router are candidates for the non-minimal path. In this case, there is always a 1 hop distance towards the intermediate group.
- ***Neighbor-router Global, (NRG)***: in non-minimal paths, traffic is diverted to a group connected to a different router in the source group. Packets traverse 2 links (lg) before reaching the intermediate group.

These policies are depicted in Figure 2. *RRG* balances evenly the non-minimal traffic load between all the global links in the network, whereas *CRG* reduces the length of non-minimal paths. *NRG* has the longest average non-minimal path what reduces performance under a uniform pattern of communications; therefore, its use for source routing is not evaluated in this paper. However, path length is not the solely objective metric: under adversarial traffic patterns, these policies also impact the fairness of the network, as will be evaluated in Sections 5 and 6.

For in-transit adaptive routing, the three previous policies can be used on each hop. Alternatively, different policies can be applied for the cases of source

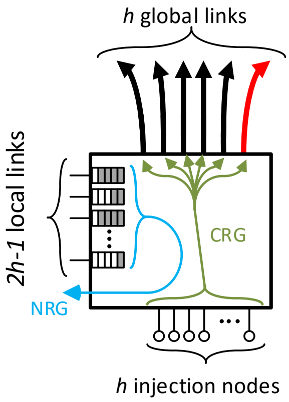


Fig. 3: MM global misrouting policy for in-transit adaptive routing.

(at injection) or in-transit (after one or more local hops) adaptive routing. The ***Mixed-mode (MM)*** mechanism implements such differentiated policy for in-transit adaptive routing:

- ***MM*** employs a *CRG* policy when attempting misrouting at the source router, and a *NRG* policy for traffic which is in-transit.

The *MM* policy is depicted in Figure 3. This *MM* policy tries to balance traffic at injection evenly across all the global links in the network; simultaneously, under adversarial traffic patterns it pretends to reduce the impact of non-minimal traffic over those global links which are heavily congested due to minimally routed traffic. However, as evaluated in Section 5, this is not enough to avoid unfairness in all cases, since results with *ADVc* traffic are still unfair.

3.2 Adversarial traffic patterns

Compared to uniform traffic, a non-uniform traffic pattern distribution introduces an uneven distribution of traffic on the network, which generates unfairness in the areas which process more traffic. In this section we present two traffic patterns which are known to introduce unfairness: *adversarial* and *adversarial-consecutive* traffic. In both cases, there exists a single router which concentrates the minimal outputs for all the traffic originated in the group, this is, the path which all traffic would follow if it were sent minimally to the destination. This router will be denoted R_{out}.

3.2.1 Adversarial traffic

Adversarial traffic (ADV, [19]) represents the worst pattern in terms of throughput, and also introduces significant problems related to fairness. Under ADV

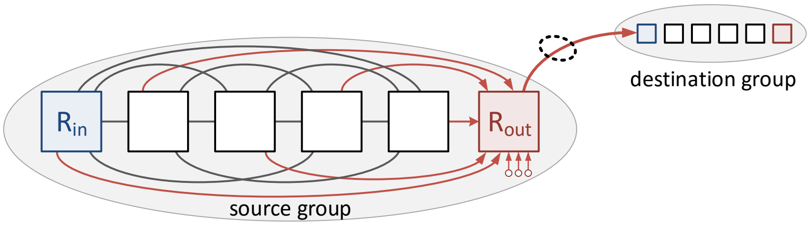


Fig. 4: Adversarial (ADV) traffic pattern in a dragonfly with $h = 3$. All the traffic from each source group i targets the group $i + 1$. The highlighted router R_{out} connects to the minimal global links towards those destination groups, entering from R_{in}.

traffic, all the traffic generated in a given group is sent towards a single destination group, with a fixed offset between source and destination groups. We will denote $ADV + i$ the pattern in which all the traffic from group n is sent to group $(n + i)\, mod\ N$, where N is the number of groups. R_{out} is the router in each group which contains the direct link to group $(n + i)\, mod\ N$. Similarly, we can define R_{in} as the router which receives the incoming traffic to a group when minimal routing is employed. Figure 4 depicts an example of ADV traffic in a dragonfly with $h = 3$, highlighting the routers R_{in} and R_{out}.

Under minimal routing, all traffic concentrates in R_{out} and is received in R_{in}. Since these two routers need to forward more traffic, their nodes can inject less and suffer a worse service level. Under adaptive nonminimal routing, the situation is not completely fair: the routing mechanism actually requires a certain level of congestion in R_{out} in order to select a nonminimal path.

3.2.2 Adversarial-consecutive traffic

In the *adversarial-consecutive (ADVc)* traffic pattern, messages are sent randomly to destinations in the h groups which are connected to the R_{out} router. With a typical *palmtree* arrangement of global links [7], these destinations are the h consecutive groups $(+1, +2, ..., +h)$ after the source group. Figure 5 illustrates this traffic pattern with a dragonfly network with $h = 3$. For other arrangements, the ADVc pattern is determined by selecting all the destination groups which are connected to a given router R_{out}.

ADVc traffic is not as adversarial in terms of throughput as ADV. Using *MIN* routing, throughput is limited to h/ap phits/node/cycle. This limitation is less severe than under ADV (which is $1/ap$) and is avoided by using nonminimal routing. However, $ADVc$ traffic constitutes a challenge for throughput fairness, since the bottleneck router of the group is likely to get all its global links congested due to the traffic routed minimally from other neighbours in the group. Furthermore, a CRG global misrouting policy (as defined in the next subsection) will aggravate this effect: in R_{out}, allowed non-minimal global paths coincide with minimal global links for traffic flows from other routers, which are probably congested. Finally, as studied in Sections 5 and 6, this

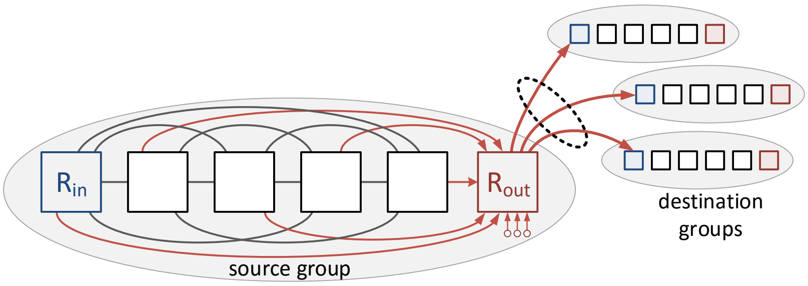


Fig. 5: Adversarial-consecutive (ADVc) traffic pattern in a dragonfly with $h = 3$. Traffic from each source group i targets the next $h = 3$ consecutive groups ($i + 1, i + 2, i + 3$). The highlighted router R_{out} connects to the minimal global link towards those destination groups.

traffic pattern saturates all the output links in R_{out}, what interferes with the congestion notification mechanism employed in PiggyBack, which is one of the best performing source adaptive routing mechanisms proposed for dragonflies.

This traffic pattern occurs in practice when an application is spread over not the whole system but $(h + 1)$ groups. A consecutive allocation of groups is the simplest approach for the job scheduler. In such case, even uniform traffic between the application processes translates into $ADVc$ traffic in the network (at least in one of the groups). Alternative allocation schemes which avoid consecutive group allocation can also inadvertently generate this traffic pattern, with a different bottleneck router in one of the groups of the system, especially for large h or for different global link arrangements.

3.3 Prioritization of in-transit traffic

Prioritization of in-transit traffic is a switch arbitration policy which always selects an in-transit packet rather than one in the injection queues when both compete for an output port. Such policy has been implemented in several systems, such as the whole line of BlueGene supercomputers [3,8]. This prioritization favours draining the network rather than injecting more traffic. Therefore, in-transit traffic prioritization reduces network congestion and can obtain higher throughput. However, as presented in Section 5.1, this policy aggravates unfairness when the traffic pattern is not uniform.

3.4 Explicit global fairness mechanism based on age-based arbitration

In this section we introduce age-based arbitration [2]. Alternative mechanisms which explicitly handle network fairness are presented in Section 8. Age-based arbitration is a variant of the switch arbitration mechanism which takes packet age into account. When two packets contend for the same output port, the arbiter compares their age (the elapsed time since they were generated) and

Parameter	**Value**
Router size	23 ports (h=6 global, p=6 injection, 11 local)
Router latency	5 cycles
Frequency speedup	2×
Group size	12 routers, 72 computing nodes
System size	73 groups, 5,256 computing nodes
Global link arrangement	Palmtree [7]
Link latency	10 (local), 100 (global) cycles
Virtual Channels	2 (global ports), 3 (local and injection ports), 4 (local ports in oblivious and source-adaptive mechanisms)
Switching	Virtual Cut-Through
Buffer size (phits)	32 (output buffer, local input buffer per VC), 256 (global input buffer per VC)
Packet size	8 phits
Congestion thresholds	55% (Adaptive in-transit), $T = 5$ (Adaptive source, local links - PB [18]), $T = 3$ (Adaptive source, global links - PB [18]),

Table 1: Simulation parameters.

always selects the oldest. This favors latency fairness by equalizing the delay of competing flows, and also provides throughput fairness.

The complexity of this mechanism relies on tracking the age of these network packets. A perfect globally synchronized network clock is not feasible, so actual implementations rely on increasing the packet age (which is a field in the packet header) by the amount of time travelling through network links and waiting in buffers.

4 Evaluation methodology

In this section we introduce the environment for our evaluations, detailing the simulation tool and the parameters we have selected. Then we describe the performance metrics that will be reproduced in Sections 5 and 6.

4.1 Simulation infrastructure

We employ the in-house designed FOGSim network simulator [12] for our evaluations. We model a dragonfly network with $h = 6$, 5256 nodes and 876 input-output-buffered routers of radix 23. An evaluation of the evolution of performance and unfairness metrics with bigger network sizes is portrayed in Section 7.1. Each router employs multiple virtual channels as a deadlock avoidance mechanism, as presented in Section 2.3. A fine-grain model of a high-radix router as described in [20] cannot be implemented for a network of this size. Thus, we employ a simpler model of a router with a 5-cycle pipeline and an iterative separable batch allocator. Routers commute traffic at 2× the link speed to reduce the performance limitations from HoL blocking and suboptimal allocator decisions. We also evaluate the impact of prioritizing in-transit traffic from injection traffic, similar to Blue Gene systems [3]. Table 1 reflects the parameters employed in our simulations.

Modelled routers employ a traditional round-robin allocator, or an age-based arbitration mechanism which has been integrated into the simulator. Our age model is ideal: at injection each packet is marked with a globally synchronized timer, which is employed by network routers to select the oldest packet. In practice, aging mechanisms [2] are required to emulate this ideal behavior.

The link latency of 10 and 100 cycles for the local and global links models the use of 2 and 20 meters wires delivering data at a 10GB/s pace, with routers transmitting 10 bytes per cycle and operating at 1 GHz. A more detailed justification for this selection can be found in [18].

All our evaluations have been conducted employing three different types of synthetic traffic: Uniform Random (*UN*), Adversarial (*ADV+1*) and Adversarial consecutive (*ADVc*). *UN* traffic selects a random destination node across all the network for every packet injected. In *ADV+1* traffic all the nodes in a given group address their traffic towards the nodes in the next group (+1); results for other destination groups are similar. Under *ADVc* traffic the nodes send their packets randomly to the nodes in the $h = 6$ next immediately consecutive groups, as detailed in Section 3.2.2. Nodes generate the packets following a Bernoulli process with an adjustable injection probability expressed in phits/(node·cycle).

In all our experiments we first warm-up the network for an adequate amount of time before tracking the average latency and throughput statistics during 15,000 cycles of execution. Curves in Sections 5 and 6 present the average of 3 different simulations. Results comprise the different metrics explained below.

4.2 Performance and throughput metrics

We have measured performance and fairness results. Performance metrics measure the capacity of the network to absorb properly a traffic load for a given traffic pattern, whereas the fairness metrics give a quantitative measure of the unbalance in the allocation of network resources between computing nodes.

We consider two performance metrics:

- Throughput: the average amount of traffic (in phits/(node·cycle)) that can be delivered to the destinations.
- Latency: the average delay between the moment a phit is inserted into the injection queue at the source router and the time it is delivered at the destination, measured in cycles. This value can be broken down into its different components, namely the waiting time at the injection, local transit and global transit queues, the delay associated to the traversal of the links in the minimal path, and the traversal of the links in the non-minimal path.

As for the fairness metrics, multiple indicators are frequently used to quantify the presence of throughput unfairness:

- Number of injected packets (or traffic load): we compute the number of injected packets at each router of a given group. This allows to determine the difference in network resources allocation to the nodes at each different router, and detect the existence of a router whose nodes suffer starvation.
- Minimal injected load (*Min inj. load*): the lowest number of packets (or traffic load) injected per router in the network. This allows to detect a case of unfairness across the whole network. This value represents a combined metric of performance and fairness, since it can be constructed as the product of average throughput (which is a performance metric) and the quotient between lowest injected throughput and average throughput (which constitutes a metric of fairness):

  $$min\ inj. = \frac{thput_{min}}{thput_{avg}} \times thput_{avg}$$

  However, it fails to determine if the existence of unfairness is an isolated anomaly or a common behavior for multiple routers in the network. For this reason, we contemplate the next two metrics.
- Max-to-min ratio (*Max/Min*): quotient between the highest and lowest number of injections per router in the network. This highlights both the cases in which a router receives an excessively high or low amount of resources compared with the rest of the network.
- Coefficient of variation (CoV): the quotient between the variance and the average number of injections per router:

  $$COV = \frac{\sigma}{\mu}$$

  With this metric we are able to discriminate between a case in which one router has an isolated situation of starvation and another router is given an abnormally high number of resources, and a case in which half of the routers starve and the other half benefit from an unfairly high number of allocated resources. Obviously, from the point of view of the applications both situations are undesirable, but it can be argued that the latter would have a more negative impact on the application performance.

5 Fairness and performance results without explicit fairness mechanism

This section presents performance and fairness results of a dragonfly network without employing any explicit mechanism to improve fairness. Section 5.1 first considers a network with prioritization of in-transit traffic, as defined in Section 3.3. Results with prioritization of in-transit traffic are particularly unfair when using in-transit adaptive routing mechanisms; with source adaptive routing, results are more fair but performance is poor. Section 5.2 removes such priority to allocate resources evenly between injection and in-transit traffic flows.

Results display a significant mitigation in throughput unfairness, specially for in-transit routing mechanisms, at the expense of a drop in performance. In both sections, results are split into performance and fairness results.

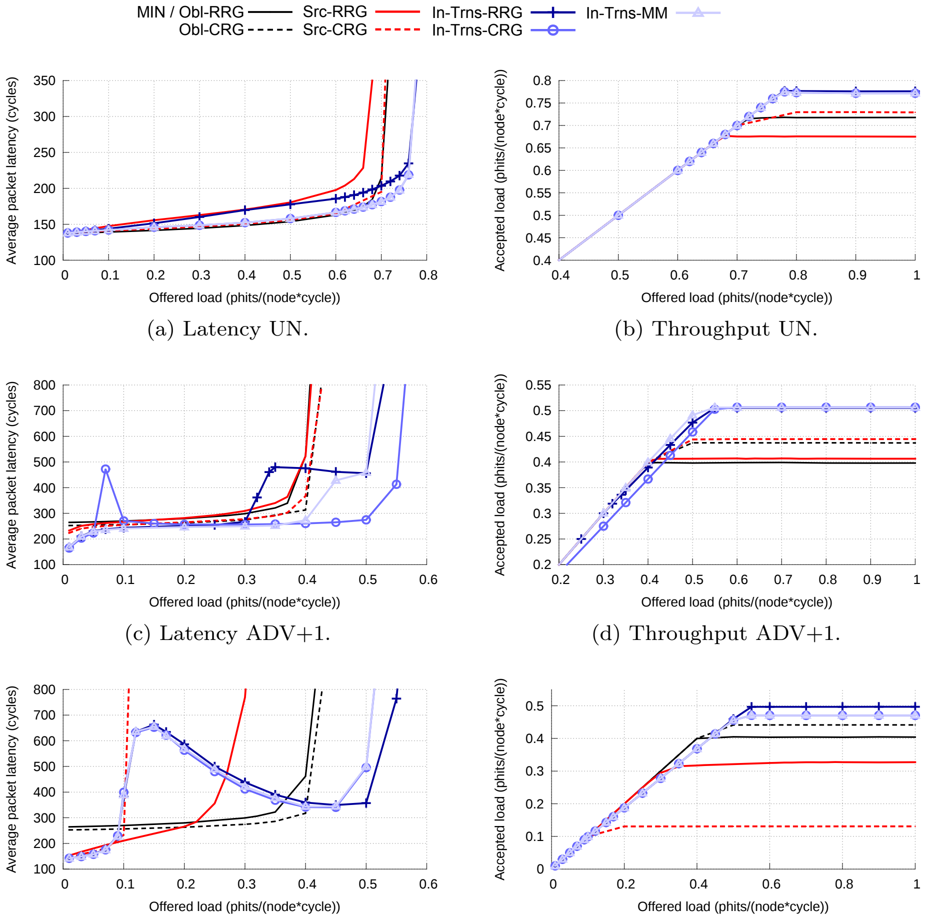
5.1 Performance results with in-transit-over-injection priority

5.1.1 Latency and throughput

Figure 6 shows average throughput and latency for the described oblivious, source adaptive and in-transit adaptive routing mechanisms under *UN*, *ADV+1* and *ADVc* traffic patterns, using in-transit-over-injection priority. Performance under *UN* traffic in Figures 6a and 6b is good for all the routing mechanisms; in these plots, the black reference corresponds to *MIN*. The latency of the *CRG* and *MM* policies (which use global links for misrouting at injection) is close to the minimal marked by the *MIN* routing. Source adaptive routing mechanisms perform misrouting only at injection, and for less than 20% of the packets. In-transit adaptive routing employs non-minimal paths for up to the 30% of the traffic, and most of that misrouting is performed in-transit. In this case, the use of *RRG* is detrimental compared to the other policies, as it increases latency and has a negligible to negative effect in throughput (especially with source routing).

The impact of the global misrouting policy gains interest under adversarial traffic patterns *ADV+1* and *ADVc* (Figures 6c, 6d, 6e and 6f). In these cases, the reference black lines represent nonminimal oblivious routing. Under *ADV+1* traffic, *CRG* again performs better (higher throughput and lower latency) than *RRG* for all the routing mechanisms; the spike in average latency for *in-transit-CRG* is discussed later. *RRG* employs in average longer paths than *CRG* (because of the extra local hop in the source group) what increases latency and reduces throughput. Best performance is achieved by the in-transit adaptive routing with the *MM* global misrouting policy, as a consequence of utilizing the most beneficial selection at injection (*CRG*) and during network traversal (*NRG*). Interestingly, all misrouting policies under both source and in-transit adaptive routing perform misrouting for a similar 97% of the total delivered traffic.

The effect of unfairness with in-transit adaptive routing under *ADV+1* is obvious in Figure 6c. Average latency presents a peak when the bottleneck router starts to suffer starvation, because in-transit traffic received from neighbour routers is given precedence in the arbitration. With *CRG*, this occurs at an extremely low load. After this point, the accepted load of this starved router remains low. Its high latency is hidden when averaging with the remaining routers in the group, which are not saturated and inject a higher load. *CRG* and *RRG* experiment this behavior at a higher traffic load of 0.3 and 0.4 phits/(node·cycle), respectively, and the reduction of the average latency never occurs. Instead, there is a flat region where the general increase in



(a) Latency UN.

(b) Throughput UN.

(c) Latency ADV+1.

(d) Throughput ADV+1.

(e) Latency ADVc.

(f) Throughput ADVc.

Fig. 6: Latency and throughput under uniform (UN) and adversarial traffic (ADV+1, ADVc), prioritizing transit over injection. Round-robin arbitration.

latency is compensated by a lower presence of high-latency packets from the starving routers.

When starvation occurs in R_{out}, the average accepted throughput is lower than the offered load, even before reaching the saturation point. The most prominent case in Figure 6d is in-transit adaptive routing with *CRG*.

Under *ADVc* traffic in Figures 6e and 6f, all the routing mechanisms fail to perform well in both metrics. The oblivious and source adaptive routing mechanisms have lower latency and do not present peaks due to throughput unfairness below the saturation point, but their throughput is relatively low. In the case of source adaptive routing, the Piggyback implementation we employ fails to properly identify global links as saturated . This enforces more than 15% of the traffic to be sent minimally with *RRG*, and more than 40% in

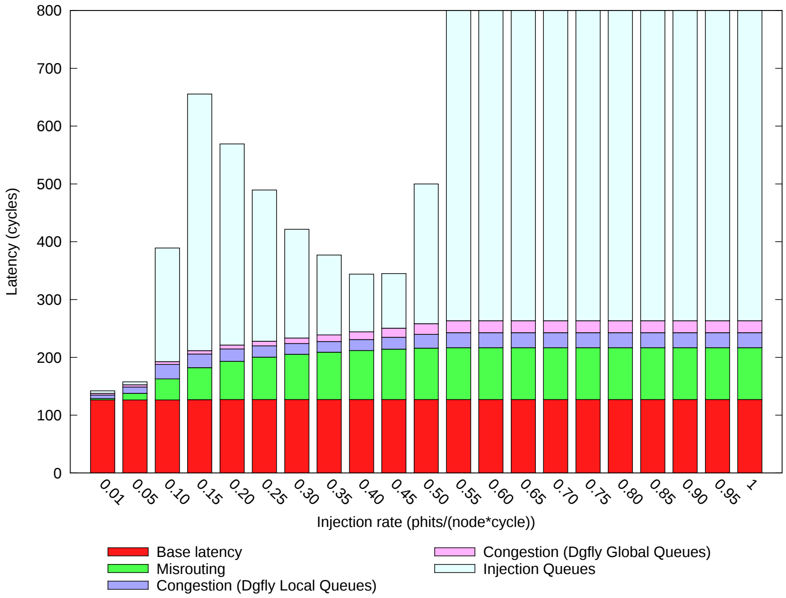


Fig. 7: Breakdown of the latency components for the in-transit adaptive routing with MM policy under ADVc traffic.

the case of *CRG*. This problem is evaluated with more detail in Section 7.3. In-transit adaptive routing performs best in throughput but clearly suffers from throughput unfairness. This can be appreciated in the throughput curves before saturation which are below those of oblivious routing, and in the peak and subsequent fall in latency at 0.15 phits/(node·cycle). The advantage in throughput over source adaptive routing is due to the ability of in-transit adaptive routing to route non-minimally in presence of congestion across the packet path, enforcing a much higher 90% of total misrouted traffic.

It is remarkable that *CRG* is the most suitable global misrouting policy for oblivious nonminimal routing under *ADV+1* and *ADVc* traffic, whereas the source adaptive routing benefits from the *RRG* policy under the *ADVc* traffic pattern. This conduct arises because the granularity for the congestion threshold in the local queues is much lower than for the global queues, forcing an excessive amount of minimally-routed traffic through the bottleneck router.

Figure 7 displays a latency breakdown for the in-transit adaptive routing with *MM* policy under *ADVc* traffic. Five different components are considered: link traversal through the minimal and non-minimal paths, waiting time in local and global link queues, and waiting time at injection. Misrouting latency, caused by the traversal of the non-minimal links, increases with the injection rate until the saturation point, at 0.5 phits/(node·cycle). Congestion, both in local and global links, has a relatively low impact on the total latency under all traffic loads. The average waiting time at injection queues shows a

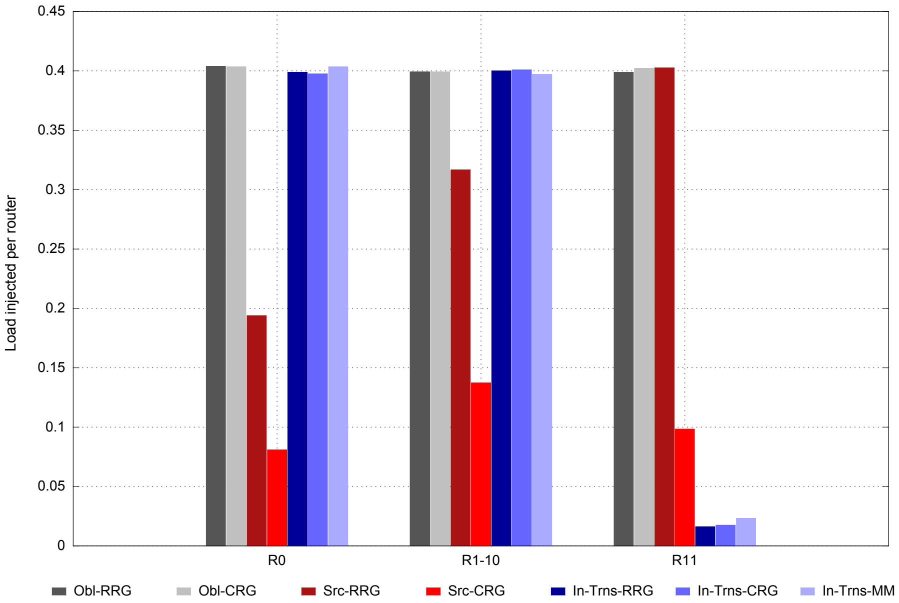


Fig. 8: Injected load of the nodes of each router in a group, under ADVc traffic with a traffic load of 0.4 phits/(node·cycle). Results for routers 1-10 are averaged in one set of columns. In-transit traffic is given priority over injection.

remarkable behavior: it grows before reaching a peak at 0.15 phits/(node·cycle) and then steadily diminishes until reaching saturation. Again, this behavior reflects an unfairness effect in which the bottleneck router saturates at low loads and suffers high latency, but its impact is hidden as more packets from other routers are averaged when the offered load increases.

5.1.2 Throughput fairness with in-transit-over-injection priority

Figure 8 portrays the injected load in every router of one group under *ADVc* traffic with a traffic load of 0.4 phits/(node·cycle), for the different combinations of routing and global misrouting policy. With this traffic pattern, R_0 and R_{11} behave respectively as R_{in} and R_{out} as depicted in Figure 5. The values for routers 1-10 have been averaged for the sake of clarity, as they present the same behavior.

Oblivious non-minimal routing (*Obl-RRG* and *Obl-CRG*) does not suffer from throughput unfairness, injecting a similar amount traffic in all the routers of the group with both global misrouting policies. However, adaptive routing mechanisms present a completely different conduct. Source adaptive routing *Src* tends to favor some routers in detriment of others: with a *RRG* global misrouting policy, router R_0 injects a significantly lower amount of packets than the rest, whereas router R_{11} injects a higher amount of traffic; with the *CRG* policy both R_0 and R_{11} inject a lower amount of traffic than the others.

	Avg sat. load	Offered load	Min inj. load	Max/Min	COV
MIN	0.07	0.05	0.0432 (86.46%)	1.336	0.0425
		0.40	0.0249 (6.23%)	9.3792	0.1882
Obl-RRG	0.40	0.35	0.3347 (95.6%)	1.106	0.0154
		0.45	0.3363 (74.7%)	1.367	0.0472
Obl-CRG	0.43	0.40	0.3828 (95.7%)	1.095	0.0145
		0.45	0.4236 (94.1%)	1.102	0.0148
Src-RRG	0.32	0.30	0.1908 (63.6%)	1.673	0.0501
		0.40	0.1897 (47.4%)	2.196	0.1217
Src-CRG	0.20	0.10	0.0904 (90.4%)	1.218	0.0292
		0.40	0.0753 (18.8%)	2.735	0.1029
In-Trns-RRG	0.52	0.40	0.0033 (0.82%)	585.69	0.2866
		0.55	0.0026 (0.47%)	231.93	0.2865
In-Trns-CRG	0.50	0.40	0.0028 (0.70%)	185.60	0.2861
		0.55	0.0014 (0.25%)	622.70	0.2907
In-Trns-MM	0.50	0.40	0.0062 (1.55%)	72.576	0.2858
		0.60	0.0019 (0.32%)	333.51	0.2900

Table 2: Fairness metrics for the different routing mechanisms and global misrouting policies, under ADVc traffic. Values are specified for two different traffic loads per combination: one below and one above the average saturation point. Traffic in the transit queues is being prioritized over traffic in the injection queues.

With in-transit adaptive routing (which obtained the best throughput and latency results in almost all cases presented in Figure 6), the injected traffic at the bottleneck router is several orders of magnitude lower than in the other routers of the group for the three global misrouting policies.

We quantify the unfairness through the metrics described in Section 4.2. Table 2 refers the minimum injection, max/min ratio, and coefficient of variation for all the routers in the network for the simulation in Figure 8. Since the level of unfairness typically increases after saturation, for each routing mechanism we indicate its average saturation load, and we present results for two load values, one slightly before and another after saturation.

For the sake of reference, results with *MIN* routing are included. *MIN* achieves extremely low throughput values under all adversarial traffic patterns, and thus saturates at a traffic load of only 0.07 phits/(node·cycle). However, it achieves reasonably good fairness metrics before reaching saturation, with a lower Max/Min than all in-transit adaptive routing mechanisms. It presents higher unfairness than non-minimal oblivious routing, specially for traffic loads above the saturation point, because the severity of the congestion is much higher and thus limits the amount of injection that can be achieved at the router directly connected to the destination group.

Fair mechanisms such as *Obl-RRG* before saturation present a minimum injected load which corresponds roughly to 95% of the offered load. After saturation, this percentage is obviously reduced, but the Max/Min ratio typically also increases, indicating that some nodes inject more than others.

All the in-transit adaptive configurations and *Src-CRG* perform significantly worse than oblivious and *Src-RRG*, with a significantly lower injected traffic per router. The *Max/Min* metric adds further information, with all the routing mechanisms achieving the same order of magnitude before saturation for the different global misrouting policies: around 1.1 for oblivious, around 1.2-1.6 for source adaptive, and around 70-500 for in-transit adaptive. The problem of in-transit adaptive routing relies on the starvation in the congested router in the group, as observed in Figure 8.

The conclusion of this subsection is that prioritization of in-transit traffic for adaptive routing is disadvantageous in general: source adaptive routing presents relative low throughput, and in-transit adaptive routing suffers severe starvation under adversarial traffic patterns.

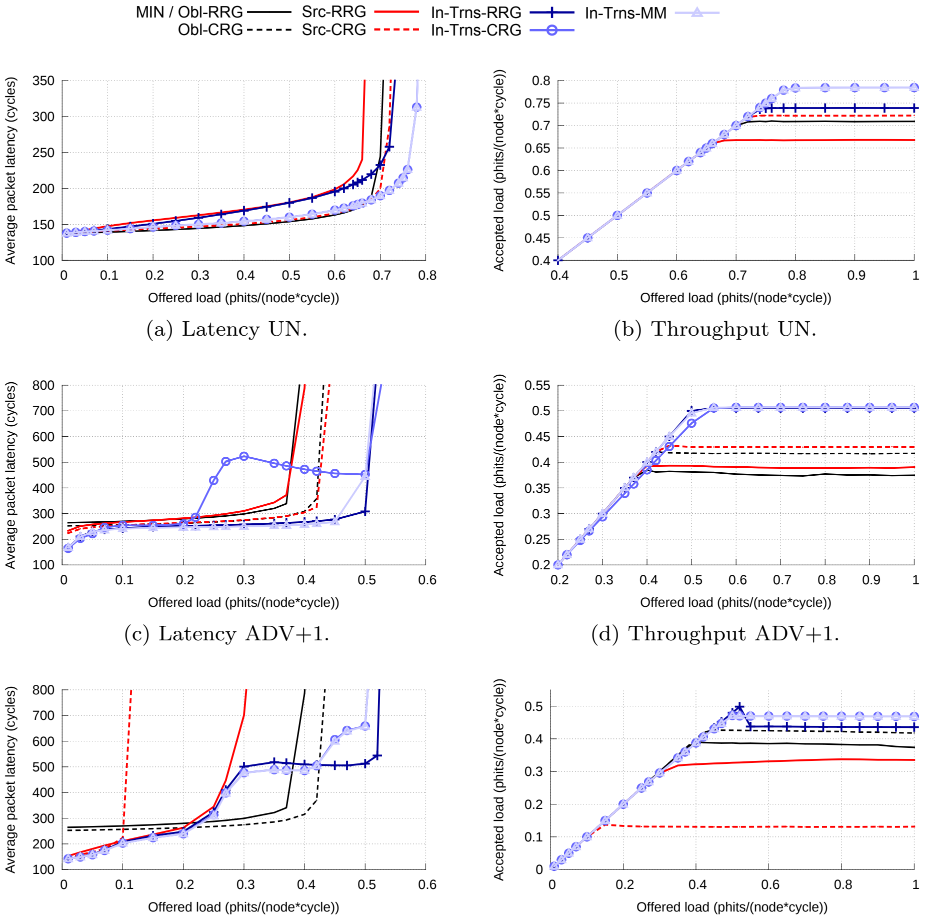
5.2 Performance and fairness without in-transit-over-injection priority

This subsection evaluates dragonflies without priority to in-transit traffic. Figure 9 presents performance results. Removing this priority increments the congestion level in the network, what can reduce throughput. However, the reduction is minimal: under *UN* traffic, throughput for *MIN* decreases around a 1.2%. For source adaptive routing, the behavior is similar to the one with priority in Figure 6. However, for in-transit adaptive routing this change improves latency significantly under *ADV* traffic. With *CRG* or *MM*, latency peaks caused by starvation do not appear; with *RRG*, the peak appears but at a much higher load.

Nevertheless, *ADVc* traffic still exhibits a latency response that can be undoubtedly attributed to throughput unfairness. The improvement over the results with transit priority in Figures 6e and 6f is noteworthy, but unable to effectively eliminate it.

Figure 10 presents the injected load of the nodes of each router in a group under *ADVc* traffic with a load of 0.4 phits/(node·cycle), without in-transit-over-injection priority. Compared to Figure 8, oblivious routing mechanisms maintain their behaviour, without any significant throughput unfairness between the routers. Source adaptive routing displays a difference with the *CRG* policy in the bottleneck router R_{11}, showing a significatively higher load. This load is not only higher than the case with priority, but also higher (more than 2×) the load in other routers in the group. Such variation can be easily explained by the absence of transition-over-injection priority, which was preventing a higher injection at the bottleneck router. Since the selection between minimal and nonminimal paths is based on the saturation of the links, the bottleneck router becomes itself aware of the status of the minimal global links faster than any other network. Hence, it is capable of exploiting the global links as soon as they stop being saturated, and makes an unfairly high use of said resources.

Disabling in-transit-over-injection priority significantly improves the injected load of R_{out} with in-transit adaptive routing under all three global misrouting policies (*RRG*, *CRG*, *MM*), with an similar improvement for all of them.



(a) Latency UN.

(b) Throughput UN.

(c) Latency ADV+1.

(d) Throughput ADV+1.

(e) Latency ADVc.

(f) Throughput ADVc.

Fig. 9: Latency and throughput under uniform (UN) and adversarial traffic (ADV+1, ADVc), without prioritizing transit over injection. Round-robin arbitration.

Values in Table 3 quantify the unfairness level with this configuration. Interestingly, before reaching saturation *MIN* routing achieves the same results as with priority, but for higher traffic loads the unfairness aggravates when the priority is removed. This occurs because, by removing the priority of in-transit over injection, each node at the router directly linked to the destination groups has the same share of the global output links as all the nodes in a different router combined. Nonetheless, *MIN* achieves such low performance under adversarial traffic patterns that the impact of removing in-transit-over-injection priority in this case shall not be considered relevant for the evaluation.

The most significant change with respect to the numbers in Table 2 occurs for in-transit adaptive mechanisms. Their starvation problem in R_{out} is avoided, so their Max/Min ratio is reduced to reasonable values around 1.8

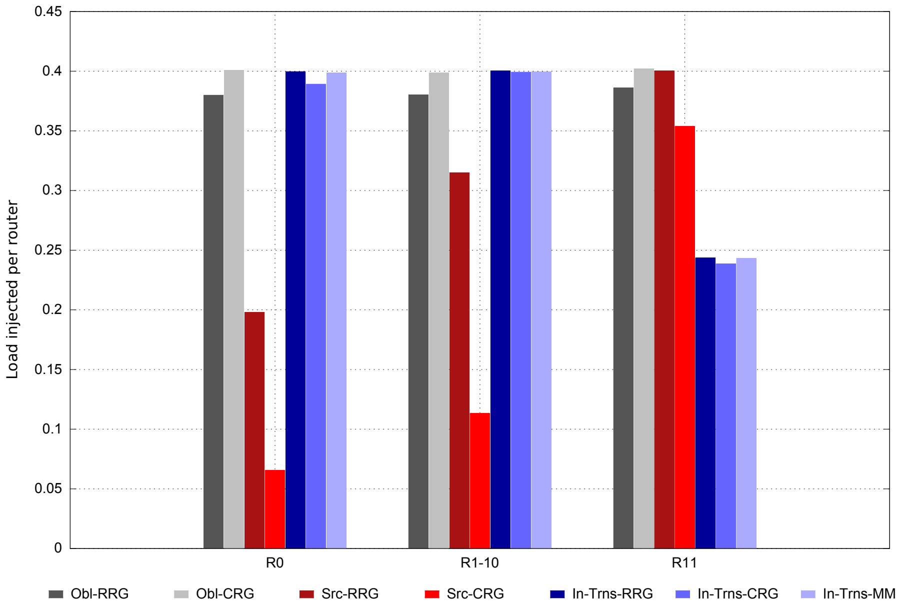


Fig. 10: Injected load of the nodes of each router in a group, under ADVc traffic with a traffic load of 0.4 phits/(node·cycle), without in-transit-over-injection priority. Results for routers 1-10 are averaged in one set of columns.

	Avg sat. load	Offered load	Min inj. load	Max/Min	COV
MIN	0.07	0.05	0.0432 (86.46%)	1.336	0.0425
		0.40	0.0119 (2.98%)	34.266	1.0790
Obl-RRG	0.40	0.35	0.3334 (95.2%)	1.105	0.0155
		0.40	0.3500 (87.5%)	1.190	0.0173
Obl-CRG	0.42	0.40	0.3835 (95.8%)	1.093	0.0144
		0.45	0.3913 (86.9%)	1.191	0.0230
Src-RRG	0.32	0.30	0.1974 (65.8%)	1.608	0.0472
		0.40	0.1998 (49.9%)	2.086	0.1194
Src-CRG	0.15	0.10	0.0895 (89.5%)	1.219	0.0293
		0.40	0.0614 (15.3%)	6.673	0.5562
In-Trns-RRG	0.52	0.40	0.2270 (56.7%)	1.850	0.1106
		0.55	0.2240 (40.7%)	2.488	0.1418
In-Trns-CRG	0.50	0.40	0.2266 (56.6%)	1.852	0.1111
		0.55	0.2071 (37.6%)	2.707	0.1633
In-Trns-MM	0.50	0.40	0.2271 (56.7%)	1.843	0.1101
		0.55	0.2134 (38.8%)	2.622	0.1634

Table 3: Fairness metrics for the different routing mechanisms and global misrouting policies, under ADVc traffic, without in-transit-over-injection priority. Two load values are employed, below and above the average saturation point of every combination.

before saturation. Interestingly, while in-transit adaptive mechanisms present worse fairness than source adaptive in terms of fairness and COV, their absolute minimum injected load is higher because their average throughput is significantly better. In any case, all the adaptive routing mechanisms are significantly unfair when compared to oblivious ones, with Max/Min ratio around 1.1 and much lower COV results.

6 Results with age-based arbitration

This section presents the results obtained when age-based arbitration is employed instead of the default Round-robin arbitration. Our implementation of age-based arbitration has been described in Section 4.

We have evaluated the use of priority for in-transit traffic in combination with age-based arbitration, selecting the oldest of the available in-transit packets, and attending the oldest injection traffic otherwise. However, this interferes with the nature of the age-based arbitration mechanism and reduces fairness, presenting similar pathologies to those observed in Section 5.1.2. We omit these results for simplicity.

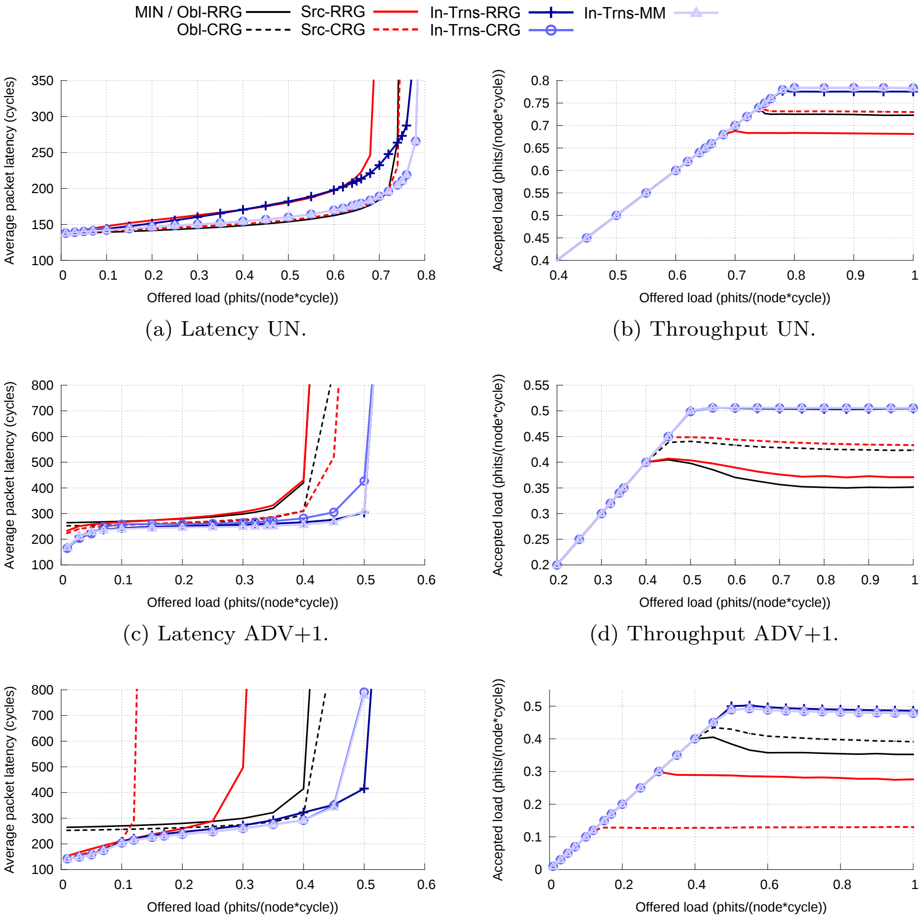
Figure 11 shows the latency and throughput results for all the routing mechanisms and the three considered traffic patterns. We first observe that none of the curves presents clear fairness pathologies such as latency spikes, as occurred with the previous configurations. Under uniform traffic, the use of the *RRG* policy is detrimental for latency because it employs longer paths. *CRG* saves the first local hop in most cases, what also increases average throughput. The *MM* policy performs similarly to *CRG* with in-transit adaptive routing, or even slightly better for adversarial traffic.

The poor throughput of Source-based adaptive mechanisms under *ADVc* traffic is remarkable. This result resembles the one in Figure 9f, but it is interesting that age-based arbitration does not solve it. As discussed before, the problem comes from an improper detection of congestion in remote global links. This problem is detailed later in Section 7.3.

For both adversarial traffics, the use of age-based arbitration introduces a problem of congestion, which slightly reduces throughput after saturation. With in-transit adaptive routing this effect is much lower, barely noticeable in *ADVc* traffic. With both oblivious and source adaptive the problem is clear. Interestingly, under *ADVc* traffic the throughput results for source adaptive are inverted, and *RRG* clearly presents the best result. We suspect that in all cases this comes from alleviating the pressure in the congested router.

The injection per router is presented in Figure 12. Again, router R_0 receives the traffic from other groups while router R_{11} receives the outgoing traffic from the group, all under minimal routing. The rest of the routers present the same behaviour and are collapsed into a single set of bars for the sake of simplicity.

There are two very significant changes from the results without age-based arbitration. First, all the in-transit adaptive routing mechanisms obtain a fair result. This is expected, since age-based arbitration provides global fairness between all competing flows. The interesting part is that our source adaptive mechanism fails to obtain fairness. In fact, when a nonminimal routing decision is taken at injection, the traffic does not compete with other flows in the



(a) Latency UN.

(b) Throughput UN.

(c) Latency ADV+1.

(d) Throughput ADV+1.

(e) Latency ADVc.

(f) Throughput ADVc.

Fig. 11: Latency and throughput under uniform (UN) and adversarial traffic (ADV+1, ADVc), no transit over injection priority, age-based arbitration.

congested router R_{out}, so the arbitration mechanism employed in said router does not really impact. For *Src-RRG* there is a significant variation between traffic in R_0, R_{11} and the rest of the routers, with R_{11} (the congested router) receiving the best injection rate. With *Src-CRG* the variation is reduced, but the throughput is so low that the mechanism is not competitive.

Finally, Table 4 quantifies the fairness results of each configuration. Even with the explicit fairness mechanism of age-based arbitration, all the configurations present a given level of unfairness after saturation, specially *Src-CRG*. Before saturation, in-transit adaptive mechanisms perform as fair as the oblivious ones using any of the global misrouting policies, as observed in Figure 12. The results with source adaptive routing, again, are constrained by their poor average throughput, particularly with *CRG*. The reference *MIN* routing per-

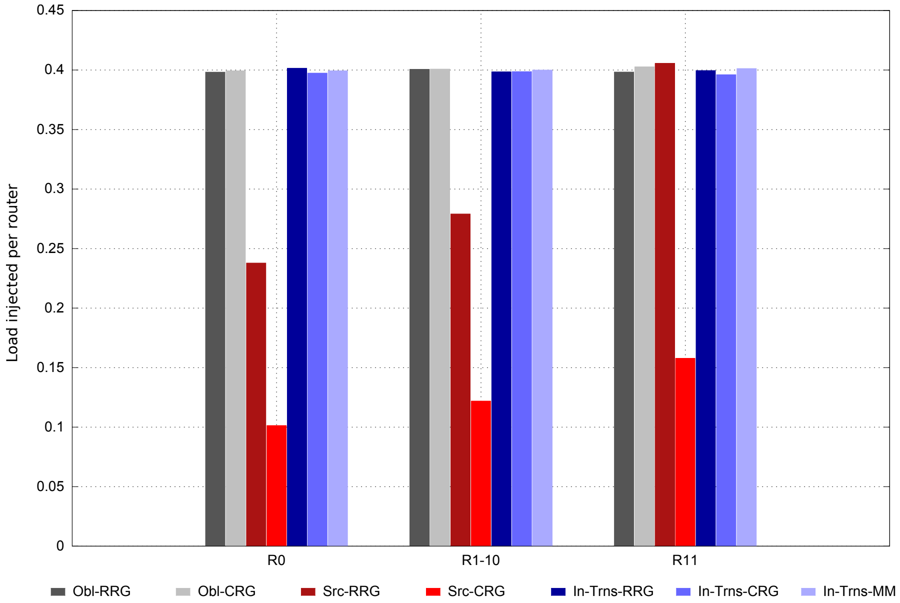


Fig. 12: Number of injected packets per router in a group of the dragonfly network, under ADVc traffic with a traffic load of 0.4 phits/(node·cycle), employing Age-based arbitration without transit over injection priority. Results for routers 1-10 are averaged in one set of columns.

	Avg sat. load	Offered load	Min inj. load	Max/Min	COV
MIN	0.07	0.05	0.0432 (86.46%)	1.336	0.0425
		0.40	0.0453 (11.33%)	4.629	0.1402
Obl-RRG	0.40	0.35	0.3322 (94.9%)	1.108	0.0157
		0.50	0.3181 (63.6%)	1.576	0.0183
Obl-CRG	0.45	0.40	0.3822 (95.5%)	1.101	0.0145
		0.50	0.3741 (74.8%)	1.366	0.0606
Src-RRG	0.30	0.25	0.2357 (94.3%)	1.121	0.0186
		0.40	0.2270 (56.7%)	1.813	0.1412
Src-CRG	0.15	0.10	0.0912 (91.2%)	1.203	0.0292
		0.40	0.0982 (24.5%)	3.195	0.1587
In-Trns-RRG	0.50	0.40	0.3798 (94.9%)	1.107	0.0147
		0.55	0.4215 (76.6%)	1.352	0.0504
In-Trns-CRG	0.50	0.40	0.3798 (94.9%)	1.104	0.0148
		0.55	0.3732 (67.8%)	1.518	0.0693
In-Trns-MM	0.50	0.40	0.3829 (95.7%)	1.096	0.0146
		0.55	0.3767 (68.5%)	1.501	0.0683

Table 4: Fairness metrics for the different routing mechanisms and global misrouting policies under ADVc traffic with two traffic loads; one is below the average saturation point, and the other is above. Age arbitration is employed in all cases.

forms notably worse than the other mechanisms since the amount of congestion is extremely high due to a poor balance of the link usage.

7 Discussion

In this section we discuss some of the aspects evaluated in the paper in more detail. We first consider different network sizes and observe the evolution of performance and unfairness under ADVc traffic for different routing and allocation mechanisms. Next, we consider alternatives to avoid the appearance of ADVc traffic, which has been proven the most problematic with respect to fairness or performance. Finally, we provide more detail on the problem of our Source adaptive routing implementation and discuss alternative fairness mechanisms for dragonflies and their limitations.

7.1 Fairness evolution with network size

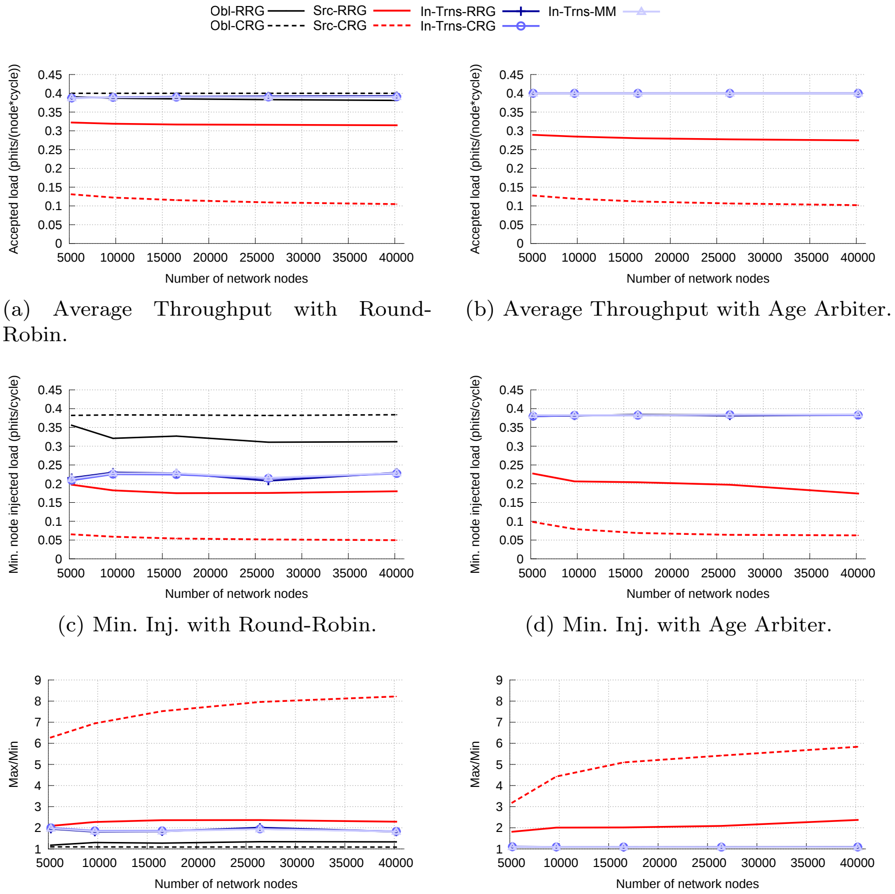
We study the evolution of the performance and fairness results in Sections 5 and 6 when different network sizes are considered. For this purpose, we have collected average throughput and fairness results for four additional network sizes, ranging up to more than 40000 nodes and using routers up to 40 ports ($h = 10$). Figure 13 illustrates those results, where the smallest network size corresponds to the one employed in previous sections.

In general, the problem of unfairness remains similar or becomes even more critical as the network size grows. The average throughput in Figures 13a and 13b remains similar as the network grows with oblivious and in-transit adaptive routing, whereas it decreases slightly when using source adaptive routing. This confirms that source adaptive routing mechanisms become less able to accurately determine the presence of congestion under bigger network sizes, reducing achieved performance.

Figures 13c, 13d, 13e and 13f demonstrate that the severity of the unfairness effect remains equally unchanged with the network size for both oblivious and in-transit adaptive routing with both arbitration policies. Furthermore, the effect exacerbates with source adaptive routing, specially in the case of *Src-CRG* routing. This validates both the inability of round-robin arbitration to prevent unfairness and the efficacy of age-based arbitration to eradicate it, as well as the fairness level achieved by in-transit adaptive routing.

7.2 Avoiding the appearance of adversarial-consecutive traffic

As presented in Section 3.2.2, *adversarial-consecutive* traffic occurs naturally with applications which are spread through multiple $(h + 1)$ groups, and one of them connects to the others from a single output router R_{out}. In such case, uniform traffic at the application level translates into *adversarial-consecutive* traffic in that particular group. While in our evaluations we considered that all groups adhere to the same pattern, in multiple cases only one group would suffer this pattern; in other groups the minimal output links can fall on multiple routers. *ADVc* traffic might occur naturally within a single large application, but we consider this case quite rare.



(a) Average Throughput with Round-Robin.

(b) Average Throughput with Age Arbiter.

(c) Min. Inj. with Round-Robin.

(d) Min. Inj. with Age Arbiter.

(e) Max/Min with Round-Robin.

(f) Max/Min with Age Arbiter.

Fig. 13: Throughput and unfairness evolution with network size under ADVc traffic with a traffic load of 0.4 phits/(node·cycle), employing round-robin and age-based arbitration. In-transit traffic is not given priority over injection.

The system batch scheduler could consider the topology when selecting nodes for a new job in order to avoid $ADVc$ traffic. To do this, it should check all selected groups to validate that remote ones are not connected via a single output router. However, since allocation is a dynamic process which occurs as new jobs arrive, a practical policy which completely avoids this traffic would be complex and probably would reduce the throughput of the system.

Modifying the topology is another option to try to avoid $ADVc$ traffic. Arranging the global links randomly has been considered before in [7]. However, such mechanism would randomize output nodes for a given subset of the network, but would not guarantee the absence of $ADVc$ or similar traffic patterns. An alternative is global trunking, which employs two or more global links between pairs of groups in small networks to provide full bisection bandwidth.

Networks with global trunking can employ disjoint pairs of routers for each parallel link between groups, avoiding the concentration of traffic in a single output router. Such design would divide the output load between two or more output routers; to minimize the concentration of traffic, parallel global links to different destination groups should connect to different routers. As far as we know, avoiding the collision of sets of parallel links in the same routers has not been considered before for the design of a dragonfly topology.

7.3 Unfairness of source-based adaptive routing under $ADVc$ traffic

In Sections 5.1 and 5.2 unfairness and degraded average performance has been observed for source adaptive routing mechanisms under the $ADVc$ traffic pattern, with both RRG and CRG global misrouting policies. Interestingly, Section 6 has proved that the unfairness remains even when age-based arbitration is employed, as opposed to the other routing mechanisms. This is specially concerning as source adaptive routing is considered one of the most suitable mechanisms considering the acceptable achieved performance and easiness of implementation.

In our evaluations we employ the PiggyBack (PB) mechanism [18] for source adaptive routing. With PiggyBack routing, packets are routed through minimal or nonminimal paths at injection depending on the saturation status of the global link that would be used in the minimal path. The definition of a global link as saturated depends on the average congestion of the different global links in the router, marking as saturated those links that double the average. This saturation status is notified to the other routers in the same group through Explicit-Congestion Notification (ECN) messages.

Under $ADVc$ traffic, the use of global links is similar for all the links connected to the same router, although it significantly varies between routers. Since the occupancy is compared to the average of global outputs only in the current router, the bottleneck router R_{out} cannot detect and communicate the saturation of all its global output links. In particular, the R_{out} router cannot discern a high-load case from this case in which all of its output links are congested. Consequently, the remaining routers in the network revert to employing the credits in their own ports, and misroute an excessively low amount of traffic. Since the amount of contending injection ports in the R_{out} router is higher than the amount of incoming traffic from any other single router, and age-based arbitration ensures that injection packets receive a fair amount of local resources, injection in R_{out} is higher compared to other routers in the group.

Age-based arbitration is unable to alleviate said problem, since the imbalance is originated because of an excessively low amount of misrouting, and the packets can not be diverted to other nonminimal paths as it occurs with in-transit adaptive routing. To adequately address this issue, a different saturation decision is required. We can employ absolute rather than relative saturation level as congestion metric, but this would trigger an excessive amount of misrouting under high loads of UN traffic, significantly reducing its performance.

8 Related work

An early analysis of unfairness issues in dragonflies has been previously presented in [13, 10]. Such analysis did not evaluate any effective solution to the unfairness problem.

End-to-end congestion control mechanisms such as TCP [4] typically also deal with fairness, in particular with an Additive-Increase, Multiplicative Decrease (AIMD) policy. However, fairness is provided only between flows which compete for some given link in their paths. Adaptive nonminimal routing mechanisms typically employed in dragonflies are not suited for such congestion control policies, since packets from each flow follow different paths.

In-network congestion in dragonflies is typically employed to drive adaptive routing, as implemented in all the adaptive mechanisms in this paper, [19, 18, 11]. Nonminimal routing is employed in such case to avoid congested areas, and throughput is typically reduced in half due to the use of Valiant routing. The limitations of the selected source-based adaptive routing employed, PiggyBack [18], have been analyzed in Section 7.3. The same work which studies PiggyBack [18] proposes two additional source-based adaptive routing mechanisms, Credit Round Trip (CRT) and Reservation (RES) routing. These two routing proposals are outperformed by PiggyBack on steady-state latency evaluations, so they have not been considered in this work.

End-point congestion requires different handling. In [16] and [17] the authors propose several reservation mechanism for dragonflies, which avoid congestion by pre-reserving bandwidth for each flow. Alternative proposals include the use of dynamically allocated side-buffers in the network switches [9].

Explicit fairness mechanisms include age-based arbitration [2] and SAT [15]. Age-based arbitration, which has been evaluated in this paper, employs a modified allocator which considers the age of the packets for arbitration. Tracking packet age is quite costly, and multiple implementations in the Network-on-Chip environment try to mimic its performance with lower cost, [21, 22, 23]. SAT restricts injection when some nodes are starving and cannot inject at their desired rate. To do so, SAT relies on a circulating signal. When some node starves, it holds the SAT signal, what eventually slows down other nodes which are waiting for the periodic message. As far as we know, SAT has not been applied before in dragonfly networks.

9 Conclusions

In this work we have evaluated throughput unfairness in a dragonfly network with different routing mechanisms, under synthetic random uniform and adversarial traffic workloads. This includes the *adversarial-consecutive* traffic pattern, which is particularly delicate for throughput unfairness.

Prioritization of in-transit traffic provides minimal benefits in terms of average throughput, but presents a high amount of throughput unfairness. Under *adversarial-consecutive* traffic this unfairness grows into starvation when in-transit adaptive routing is used. Without prioritizing in-transit traffic, in-transit adaptive routing provides the best results, particularly with RRG or MM policies which avoid concentrating traffic in the congested minimal router. However, priority removal has a negative impact on throughput fairness with one of the source adaptive routing alternatives (because of an increase of throughput in one particular router, without a significant increase in average throughput), whereas it proves insufficient to completely remove unfairness in the other cases.

Age-based arbitration is employed as an explicit fairness mechanism to avoid throughput unfairness. With age-based arbitration, in-transit adaptive routing provides the best performance among all the routing mechanisms and achieves complete fairness. By contrast, source adaptive routing, while being relatively fair, provides relatively poor performance.

Source adaptive mechanisms based on PiggyBack fail to properly detect congested links and are not competitive under *adversarial-consecutive* traffic pattern, regardless of the arbitration policy. An alternative mechanism is required to provide fairness injecting minimally and non-minimally, potentially achieving a significant increase in throughput.

Acknowledgements This work has been supported by the Spanish Ministry of Education, FPU grant FPU13/00337, the Spanish Science and Technology Commission (CICYT) under contracts TIN2012-34557 and TIN2013-46957-C2-2-P, and the European HiPEAC Network of Excellence.

References

1. Abts, D.: Cray xt4 and seastar 3-d torus interconnect. In: Encyclopedia of Parallel Computing, pp. 470–477. Springer (2011)
2. Abts, D., Weisser, D.: Age-based packet arbitration in large-radix k-ary n-cubes. In: Supercomputing, 2007. SC '07. Proceedings of the 2007 ACM/IEEE Conference on, pp. 1–11 (2007). DOI 10.1145/1362622.1362630
3. Adiga, N.R., Blumrich, M.A., Chen, D., Coteus, P., Gara, A., Giampapa, M.E., Heidelberger, P., Singh, S., Steinmacher-Burow, B.D., Takken, T., Tsao, M., Vranas, P.: Blue Gene/L torus interconnection network. IBM Journal of Research and Development **49**(2.3), 265 –276 (2005). DOI 10.1147/rd.492.0265
4. Allman, M., Paxson, V., Blanton, E.: TCP congestion control. RFC 5681 (2009)
5. Alverson, R.: Cray high speed networking. In: IEEE Hot Interconnects (2012)
6. Arimilli, B., Arimilli, R., Chung, V., Clark, S., Denzel, W., Drerup, B., Hoefler, T., Joyner, J., Lewis, J., Li, J., et al.: The PERCS high-performance interconnect. In: 18th Symposium on High Performance Interconnects, pp. 75–82. IEEE (2010)
7. Camarero, C., Vallejo, E., Beivide, R.: Topological characterization of hamming and dragonfly networks and its implications on routing. ACM Trans. Archit. Code Optim. **11**(4), 39:1–39:25 (2014)
8. Chen, D., Eisley, N., Heidelberger, P., Senger, R., Sugawara, Y., Kumar, S., Salapura, V., Satterfield, D., Steinmacher-Burow, B., Parker, J.: The IBM Blue Gene/Q interconnection network and message unit. In: SC: Intl. Conf. for High Performance Computing, Networking, Storage and Analysis, pp. 1–10 (2011)

9. Duato, J., Johnson, I., Flich, J., Naven, F., Garcia, P., Nachiondo, T.: A new scalable and cost-effective congestion management strategy for lossless multistage interconnection networks. In: HPCA-11: Intl. Symp. on High-Performance Computer Architecture., pp. 108–119 (2005). DOI 10.1109/HPCA.2005.1
10. Fuentes, P., Vallejo, E., Camarero, C., Beivide, R., Valero, M.: Throughput unfairness in dragonfly networks under realistic traffic patterns. In: 1st IEEE International Workshop on High-Performance Interconnection Networks Towards the Exascale and Big-Data Era (HiPINEB), pp. 801–808 (2015). DOI 10.1109/CLUSTER.2015.136
11. García, M., Vallejo, E., Beivide, R., Odriozola, M., Valero, M.: Efficient routing mechanisms for dragonfly networks. In: The 42nd International Conference on Parallel Processing (ICPP-42) (2013)
12. García, M., Fuentes, P., Odriozola, M., Vallejo, E., Beivide, R.: FOGSim Interconnection Network Simulator. University of Cantabria (2014). URL `http://fuentesp.github.io/fogsim/`
13. García, M., Vallejo, E., Beivide, R., Odriozola, M., Camarero, C., Valero, M., Labarta, J., Rodríguez, G.: Global misrouting policies in two-level hierarchical networks. In: INA-OCMC: Workshop on Interconnection Network Architecture: On-Chip, Multi-Chip, pp. 13–16 (2013). DOI 10.1145/2482759.2482763
14. Garcia, M., Vallejo, E., Beivide, R., Odriozola, M., Camarero, C., Valero, M., Rodriguez, G., Labarta, J., Minkenberg, C.: On-the-fly adaptive routing in high-radix hierarchical networks. In: 41st International Conference on Parallel Processing (ICPP), pp. 279–288 (2012). DOI 10.1109/ICPP.2012.46
15. Izu, C., Vallejo, E.: Throughput fairness in indirect interconnection networks. In: 13th International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT '12, pp. 233–238. IEEE Computer Society (2012). DOI 10.1109/PDCAT.2012.129
16. Jiang, N., Becker, D., Michelogiannakis, G., Dally, W.: Network congestion avoidance through speculative reservation. In: High Performance Computer Architecture (HPCA), 2012 IEEE 18th International Symposium on, pp. 1–12 (2012). DOI 10.1109/HPCA.2012.6169047
17. Jiang, N., Dennison, L., Dally, W.J.: Network endpoint congestion control for fine-grained communication. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '15, pp. 35:1–35:12. ACM, New York, NY, USA (2015). DOI 10.1145/2807591.2807600. URL `http://doi.acm.org/10.1145/2807591.2807600`
18. Jiang, N., Kim, J., Dally, W.J.: Indirect adaptive routing on large scale interconnection networks. In: Intl. Symp. on Computer Architecture (ISCA), pp. 220–231 (2009)
19. Kim, J., Dally, W., Scott, S., Abts, D.: Technology-driven, highly-scalable dragonfly topology. In: ISCA'08: 35th International Symposium on Computer Architecture, pp. 77–88. IEEE Computer Society (2008)
20. Kim, J., Dally, W., Towles, B., Gupta, A.: Microarchitecture of a high-radix router. In: ACM SIGARCH Computer Architecture News, vol. 33, pp. 420–431. IEEE Computer Society (2005)
21. Lee, J.W., Ng, M.C., Asanovic, K.: Globally-synchronized frames for guaranteed quality-of-service in on-chip networks. In: 35th International Symposium on Computer Architecture, pp. 89–100. IEEE (2008)
22. Lee, M., Kim, J., Abts, D., Marty, M., Lee, J.: Probabilistic distance-based arbitration: Providing equality of service for many-core CMPs. In: Microarchitecture (MICRO), 2010 43rd Annual IEEE/ACM International Symposium on, pp. 509 –519 (2010). DOI 10.1109/MICRO.2010.18
23. Miao, S.J., Hsu, Y.: Group allocation: A novel fairness mechanism for on-chip network. In: Networked Embedded Systems for Enterprise Applications (NESEA), 2011 IEEE 2nd International Conference on, pp. 1–7 (2011). DOI 10.1109/NESEA.2011.6144932
24. Valiant, L.: A scheme for fast parallel communication. SIAM journal on computing **11**, 350 (1982)
25. Won, J., Kim, G., Kim, J., Jiang, T., Parker, M., Scott, S.: Overcoming far-end congestion in large-scale networks. In: Intl. Symp. on High Performance Computer Architecture (HPCA), pp. 415–427 (2015). DOI 10.1109/HPCA.2015.7056051