Adaptive Routing in InfiniBand Hardware

1st José Rocher-González
Computing Systems Department
University of Castilla-La Mancha
Albacete, Spain
Jose.Rocher@uclm.es

2nd Ernst Gunnar Gran
HPC at Simula/
IIK at NTNU
Gjøvik, Norway
ernst.g.gran@ntnu.no

3rd Sven-Arne Reinemo
SimulaMet
Oslo, Norway
svenar@simula.no

4th Tor Skeie
HPC at Simula/
Ifi at UiO
Oslo, Norway
tskeie@ifi.uio.no

5th Jesús Escudero-Sahuquillo
Computing Systems Department
University of Castilla-La Mancha
Albacete, Spain
Jesus.Escudero@uclm.es

6th Pedro Javier García
Computing Systems Department
University of Castilla-La Mancha
Albacete, Spain
pedrojavier.garcia@uclm.es

7th Francisco J. Quiles Flor
Computing Systems Department
University of Castilla-La Mancha
Albacete, Spain
Francisco.Quiles@uclm.es

***Abstract*—Interconnection networks are the communication backbone of modern high-performance computing systems and an optimised interconnection network is crucial for the performance and utilisation of the system as a whole. One element of the interconnection network is the routing algorithm, which directly influences how we are able to utilise the physical network topology. InfiniBand is one of the most common network architectures used in high-performance computing and traditionally it only supported static routing. For multi-path networks such as Fat-trees, static routing is inefficient because it cannot balance traffic in real-time nor utilise multiple paths efficiently under adversarial traffic. This again potentially leads to unnecessary contention and an underutilised network, which has led to numerous proposals on how to avoid this by using adaptive routing. Adaptive routing has recently been introduced in InfiniBand and in this paper we evaluate to what extent the expected benefits of adaptive routing is true for InfiniBand. Through a set of experiments on HDR InfiniBand equipment we describe the basic behaviour of adaptive routing in InfiniBand, its benefits in Fat tree topologies and the unfortunate side effects related to unfairness that adaptive routing in general might introduce, including such phenomena as the reverse parking lot problem and congestion spreading.**

***Index Terms*—Adaptive routing, routing algorithms, latency, bandwidth, fairness, InfiniBand, high performance computing**

I. Introduction

Adaptive routing (AR) has been extensively discussed in the literature for at least two decades, where theory and algorithms have been proposed for various interconnection network technologies and network topologies [2], [7], [19], [26], [30]. A large body of work has shown that AR will offer superior performance compared to deterministic and oblivious routing for many non-uniform traffic scenarios. The evaluation work herein has mainly been conducted as simulation studies [1], [9], [14], [17], [18], [24], [25], [28]. On the contrary, there are also contributions revealing that deterministic routing (DR) for some scenarios and topologies, such as Fat trees, achieve a similar, or even a higher throughput than AR. This has been shown to be the case for hot-spot traffic, where AR might spread congestion, having a negative impact on traffic not destined to hot-spot receivers [23].

There are also network topologies, such as the Dragonfly versions that are dependent on non-minimal global AR between the network groups to achieve adequate performance [4], [8], [15], [27]. The reason for this is that the number of global links between the network groups is limited. If the shortest path global links are always taken, these links easily become susceptible to congestion under adversarial traffic. Therefore, packets have to be routed via an intermediate group first, to balance the load between the global links. A key point herein is not to select an intermediate group where the last hop global link is severely congested. Therefore indirect global AR has been introduced where the selection function uses information not directly available at the source router , as for instance progressive AR, piggyback routing, and reservation routing [13], [22].

It took, however, many years before AR was introduced into interconnect technologies. For instance, when InfiniBand was standardised in early 2000, AR was not part of the standard [12]. The main reason for this was the negative effect on application performance caused by out-of-order packet delivery, which was not well understood, and the inherent silicon complexity of AR compared to DR [19], [23]. Around ten years after InfiniBand was standardised, Mellanox provided support for group-adaptive routing [20] in their FDR InfiniBand products. More recently, their AR concept has been refined, as part of the EDR and HDR technologies, to include inter-switch notifications issued by switches experiencing congestion. The purpose is to inform other switches about congestion so that they may stop forwarding packets towards thronged areas [11]. This mechanism is for instance appropriate for both Fat trees and Dragonflies. Mellanox has also implemented a patented solution to handle out-of-order delivery packets [3]. Through these and several additional development steps, the AR offered by InfiniBand is reported to have become more powerful and useful, but evaluation is limited to some experiences from the early versions of AR [6] the pre-exascale system CORAL [29] that only focuses on bisection bandwidth without considering fairness.

In this paper we study the basic behaviour of InfiniBand AR. Our results show that AR is versatile and in many cases outperforms DR in terms of throughput. However, our study also reveals that AR potentially contributes to congestion spreading, and, just as important, that AR for some traffic scenarios introduces unfairness in a manner opposite to the well-known parking lot problem [10]. This new type of unfairness we label the *reverse parking lot problem*. The paper concludes with a fairness discussion explaining the fundamental characteristics causing the traditional and the reverse parking lot problems.

II. Background

In the following sections we introduce the parking lot problem and AR as implemented in InfiniBand as this is relevant background for the discussion in Section IV and V.

A. Fairness in Switch Arbitration

Fairness is a key property of any interconnection network, and a particular concern for the switch arbiters. In general, a fair arbiter provides equal service to all ingress ports requesting a particular egress port. Note however, that ingress ports and traffic flows are different concepts, and that a local arbiter at a switch, ignorant of the number of traffic flows sharing an ingress port, could be fair at the level of ingress ports (i.e., giving each ingress port its fair share of access to a given egress port) but still unfair at the traffic flow level. This challenge is referred to as the parking lot problem, illustrated in Figure 1.

In this example, the parking lot consists of four sections of cars, indicated by different colors, and a single exit lane at the top shared by cars from all four sections. At each section, cars that want to exit from the local section is interleaved with cars already present in the exit lane in a locally fair way, one by one, without consideration of the cars' colors. This is equivalent to different traffic flows from different sources sharing (parts of) the path towards a destination in a lossless network, where each entry point to the path is governed by a locally fair arbiter. Notice how this local fairness gives an obvious advantage for the cars closest to the exit, the green ones, while unfairness in treatment for cars of different colors increases with the distance from the exit (or more precisely, by the number of entry points adding new cars with a different color). In this example, cars from the green section is given half of the capacity of the exit road, 1/4 is given to cars from the yellow section, and finally 1/8 each to cars from the red and blue sections. The parking lot problem is known to be an issue when utilizing DR in Fat trees, just as in other topologies with DR and shared paths, and this problem produces unfairness to the switch arbitration, as we demonstrate in Section IV.

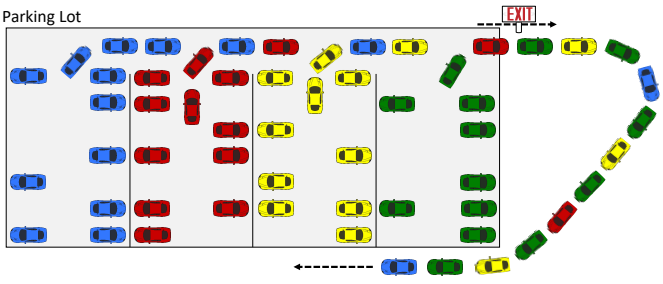


Fig. 1: The traditional parking lot problem

B. Adaptive Routing in InfiniBand

Adaptive Routing in InfiniBand, as provided by Mellanox, enables a switch to select the egress port based on port load [21]. To configure the AR mechanism, the Mellanox Subnet Manager (SM) loads the Adaptive Routing Manager (ARM) module during initialization. This module scans all the switches in the topology, identifies which ones support AR, and then configures the AR functionality on these switches. Basically, the ARM configures routing tables to allow switches to select one egress port from a set of ports belonging to the same AR *group*, given a specific destination LID. Not all topologies, however, are supported by automatic AR group configurations. The configuration of the AR groups relies on the selection of one of the currently supported algorithms [21]:

- **LAG**: An algorithm for topologies with multiple links between switches. All ports on a switch linked to the same remote switch are in the same AR group. Supported topologies include meshes and 3D torus and Hypercubes.
- **TREE**: An algorithm for Fat trees and quasi-Fat trees, including ones with parallel links between switches. All local ports with minimal hop paths to a destination are in the same AR group.
- **DFP**: An algorithm for the Dragonfly+ topology.

For more information on how to configure a given algorithm for a specific topology and routing engine, please refer to [21].

In addition to grouped adaptive routing, Mellanox InfiniBand also provides a mechanism called Adaptive Routing Notification (ARN) [11]. This mechanism is based on the idea that upstream switches need to know about faulty or congested downstream links or nodes in the network to most efficiently route around the affected areas. However, as the overall ARN mechanism is not yet fully implemented and supported [21], it has not been evaluated in this paper.

III. Experiment Configurations

To understand the performance and capabilities of AR in InfiniBand, we perform a series of gradually more complex experiments using two different cluster configurations (C1 and C2) and several different communications scenarios. The

Description	Software version
Mellanox Quantum QM8700 HDR switches	27.2008.2500
Mellanox ConnectX-6 HCAs	20.30.1004
Gigabyte R272-Z30 servers	5.14
- 1 Epyc Rome 7302P, 128GB RAM	
Mellanox OpenFabric drivers (OFED)	v5.3-1.0.0.1
OpenMPI	4.0.5
OSU Micro benchmarks	5.6.3

TABLE I: Hardware and software of the test bed.

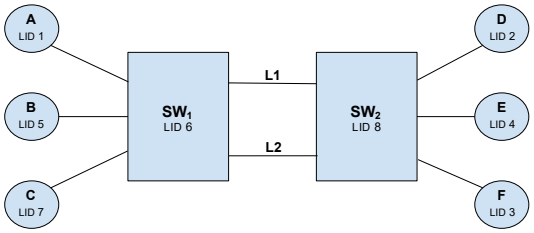


Fig. 2: Configuration 1 (C1) consists of two switches, SW_1 and SW_2, connected by two links to allow for adaptivity, and three end nodes connected to each switch.

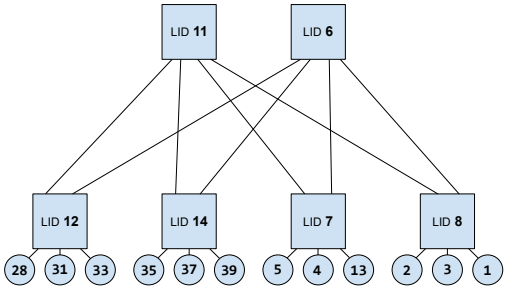


Fig. 3: Configuration 2 (C2) consists of a 2 level Fat tree topology with 6 switches. Each leaf switch is connected with three end nodes and all nodes are identified by their LID.

two configurations are described in more detail below while the individual communication scenarios are presented as we discuss the results of individual experiments in section IV. The hardware and software relevant for our experiments are listed in more detail in Table I.

A. Configuration 1

C1 consists of six end nodes interconnected using two switches as shown in Figure 2. The switches, SW_1 and SW_2, are interconnected using two links in order to allow for AR (i.e., each of the two links represents an alternative path between the switches). The end nodes A, B and C are connected to switch SW_1, while the end nodes D, E and F are connected to switch SW_2. Node F is running the subnet manager. The routing tables for DR make sure that the traffic between a given source-destination pair always follows the same path. For AR, links L1 and L2 are grouped using the LAG algorithm, thus all traffic between the switches are distributed across these two links.

B. Configuration 2

In C2, twelve end nodes are interconnected using six switches in a Fat tree topology as shown in Figure 3. Each leaf switch has individual connections to three different end nodes and two up-links. This arrangement results in a 3:2 over-subscription, chosen to showcase the potential benefits of AR versus DR. For AR, links between leaf switches and root switches are grouped using the TREE algorithm, and all end node traffic forwarded from leaf switches to root switches are distributed across the grouped links (note that in this topology there is a single shortest path available from a given root switch to a given end node). Node 28 is running the subnet manager. Again, in the case of DR, traffic between a given source-destination pair always follows the same path.

IV. Experiment Results

In this section we present and discuss our findings. We start with a series of experiments using C1 where we evaluate the basic behaviour of AR. Then we move on to the two-level Fat tree of C2 to evaluate AR performance in a larger topology with respect to throughput, fairness and latency.

A. Throughput and fairness results from Configuration 1

Our experiments using C1 consists of three scenarios with different communication patterns. In scenario 1 (S1), packets flow from node A to node D. The purpose of this scenario is to investigate the behaviour of AR in the most simple scenario: two alternative paths, represented by the links L1 and L2, and no interfering traffic. In scenario 2 (S2), two flows are added to S1, one flow from B to E and one flow from C to F. In this scenario, the three flows are competing for (routing-dependant) access to the links L1 and L2, though still with different destination nodes per flow and no egress port contention at SW_2. Finally, in scenario 3 (S3), A sends to D, while B and C both send traffic to E. In addition to all flows still sharing L1 and L2, the two flows from B and C also share the last link towards E, potentially creating congestion at the corresponding egress port of SW_2.

Figure 4 shows the average throughput of the flow from node A to node D in S1 for both DR and AR. With only one active flow there is no bottleneck in the network and there is virtually no difference between the achieved throughput for DR and AR. However, by inspecting the performance counters on the switches, we do observe a difference in how the switch to switch links are utilised. DR only uses one switch to switch link as predefined by the routing table of the switch (based on the destination LID of the packets), while AR distributes traffic between the two switch to switch links. Note that AR distributes traffic across these two links, L1 and L2, in a round-robin fashion for all injection rates even if none of the two links are saturated. If we add a second flow to S1, going from B or C to E or F, AR evenly distributes traffic belonging to the two flows onto L1 and L2, and as the inter-switch capacity is still sufficient to handle both flows the AR results per flow remain as in Figure 4. For DR, however, the result depends on the routing table of SW_1 and if the two flows are forwarded to SW_2 on separate links or on the same link (L1 *or* L2 only). In the former case, the throughput per flow remains as in Figure 4, while in the latter case the throughput is halved, as expected.

Moving on to S2, three flows are present in the C1 topology: A to D, B to E, and C to F. The results from this scenario are presented in Figure 5a and 5b and show the throughput for the three different flows using DR and AR, respectively. All flows try to send at the maximum link rate of 200 Gbps, but are limited by the aggregated inter-switch bandwidth of 400 Gbps.

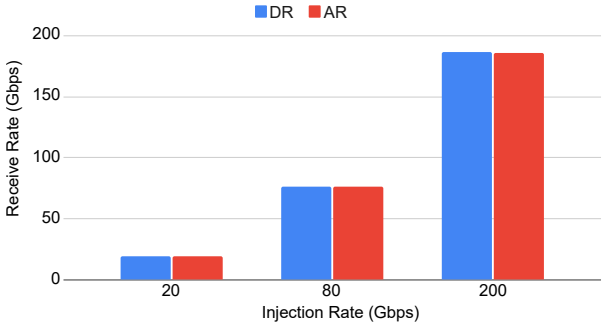
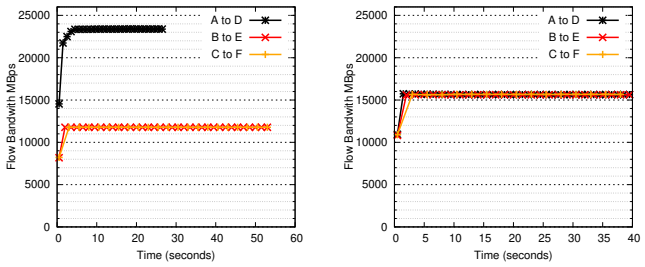


Fig. 4: Average throughput of the flow in S1 for injection rates of 20, 80, and 200 Gbps.

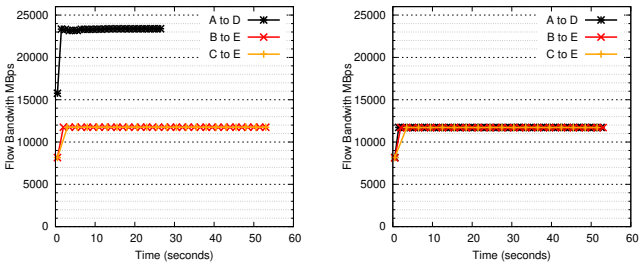


(a) Deterministic. (b) Adaptive.

Fig. 5: Throughput per flow for Scenario 2 (C1).

With DR, the flow from A to D is routed across link L1, while the flows from B to E and C to F are sharing link L2. Using DR in this scenario, two of the flows inevitably have to statically share one of the two links, here the flows from B and C, which yields an unfair distribution of bandwidth. The flow from A achieves a throughput of 188 Gbps, while the flows from B and C achieve about 94 Gbps each. When AR is used, all three flows are load balanced across the two inter-switch links and the distribution of bandwidth between the three flows is fair with each flow achieving a throughput of 125 Gbps. In other words, the total achievable throughput is the same for DR and AR in S2, but using DR leads to an unfair distribution between the three flows, while AR maintains fairness. Furthermore, note that in the case of DR, the flows from B and C would share the link L2 even if the flow from A was not present. This would lead to an unfortunate underutilization of the aggregated link capacity between the switches. On the other hand, AR would also in this case load balance the traffic using both L1 and L2, increasing the overall performance accordingly.

Finally, Figure 6a and 6b show the throughput for the three flows present in S3 for DR and AR, respectively. Considering AR first (Figure 6b), we note that AR maintains fairness between the three flows, but at a cost: AR is only able to utilize 75% of the inter-switch bandwidth. This is due to the fact that B and C combined are sending more traffic towards E than the link between SW_2 and E can handle. Consequently, congestion builds up at the corresponding egress port at SW_2 and spreads to both L1 and L2. The congestion spreading then leads to head of line blocking of the flow from A to



(a) Deterministic (b) Adaptive

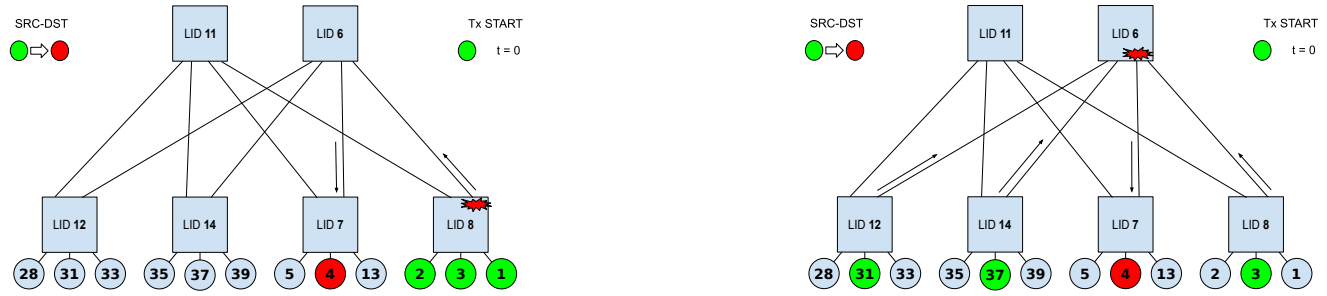
Fig. 6: Throughput per flow, Scenario 3 (C1).

D, which again hinders this flow from progressing any faster than the two flows headed towards E. Indeed, C1 combined with AR illustrates a very basic example of a general AR challenge: AR potentially distributes traffic more widely in the network compared to DR, which during periods of congestion could widen congestion trees present in a lossless network and lead to an increase in the overall amount of head of line blocking. On the other hand, using DR for S3, one out of three scenes will play out, depending on the configuration of the routing table of SW_1: 1) DR leads to the optimal solution if the two flows headed for E share the same inter-switch link while the flow towards D uses the other link. This is the situation shown in Figure 6a, resulting in 100% of the aggregated inter-switch bandwidth being used, maximizing the combined throughput of all three flows. 2) DR leads to the same inter-switch bandwidth utilization as AR (75%) if one of the flows headed for E is sharing an inter-switch link with the flow headed for D, while the other flow headed for E is the sole user of the other inter-switch link. Finally, 3) DR leads to the worst case solution if all three flows end up sharing the same link between the switches, resulting in only 50% of the inter-switch bandwidth being used. In other words, for S3, DR may lead to 50%, 75%, or 100% utilization of the inter-switch bandwidth, depending on the configuration of the routing table of SW_1. The immediate conclusion is that AR is the most *predictable* solution because it guarantees 75% utilisation in any similar scenario. However, as we will see in the next section, more complex topologies may lead to less predictable and less fair performance for AR.

B. Throughput and fairness results from Configuration 2

For Configuration 2 (C2) we use six different communication scenarios of increasing complexity to study the performance and behaviour of AR, and compare it to DR. The six scenarios build on different hot-spot situations, shown in Figures 7a-12a, to show how AR is able to cope with the individual cases. In the following, both switches and end nodes will be referred to by their LID numbers.

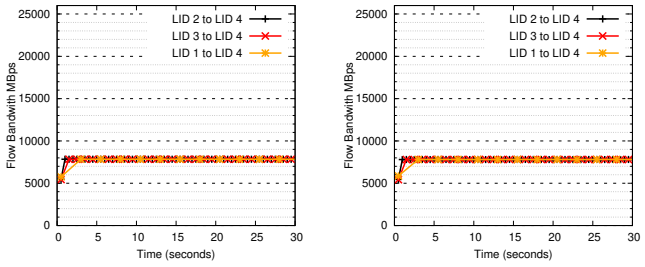
The first two scenarios, shown in Figure 7, both represent basic hot-spot scenarios where three nodes are sending traffic to a single destination connected to a remote switch (node 4). In scenario 1, the three senders (node 1, 2 and 3) are connected



(a) Scenario 1 (C2)

(b) Scenario 2 (C2)

Fig. 7: Configuration 2 - Scenario 1 and 2



(a) Deterministic

(b) Adaptive

Fig. 8: Results for Scenario 1 (C2). The results for Scenario 2 (C2) is virtually the same.

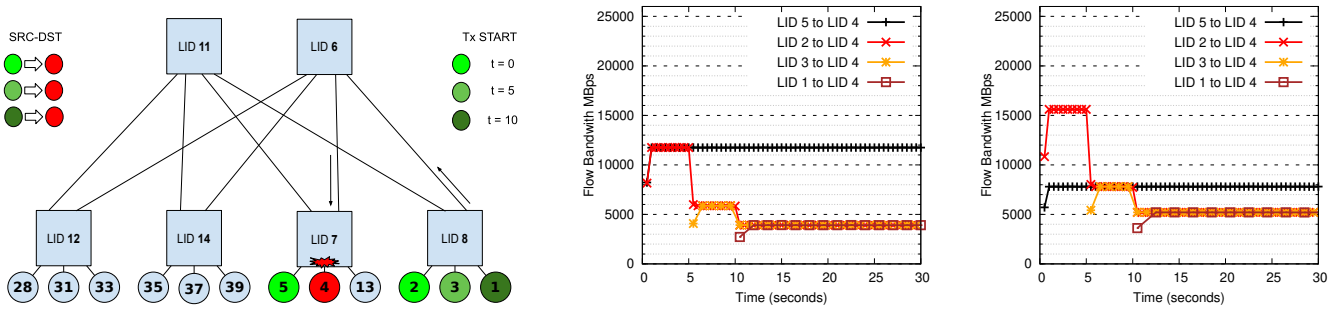
to the same leaf switch, while in scenario 2 the three senders (node 31, 37, and 3) are connected to different leaf switches.

Figure 8 shows the achieved throughput for Scenario 1 and 2. The injection rate of the three flows (200 Gbps desired per flow) is limited by the bandwidth of the hot receiver (node 4) in both scenarios, both for AR and DR. Consequently, the achieved throughput is basically identical in all four cases, but the location of the root of congestion is different. In Scenario 1, for DR the root of congestion stays fixed at the egress port of an uplink of the switch local to the senders (LID 8 in Figure 7a), indicated by the spiky red cloud. For AR, two roots of congestion immediately form at the egress ports of the uplinks at LID 8 as the combined injected traffic from the three source nodes exceeds the capacity of the links connecting LID 8 to the root switches. However, as the final link towards the hot receiver (node 4) is the main bottleneck, the corresponding egress port at LID 7 will swiftly become the permanent root of congestion, leaving the old roots of congestion as part of the branches in the main congestion tree. In the second scenario, for DR the root of congestion is located at the egress port of the root switch corresponding to the path chosen by the routing algorithm, in our case LID 6 (Figure 7b). In case of AR, temporary roots of congestion will form at both root switches prior to the permanent root of congestion being created at the node 4 egress port of LID 7 – as in the first scenario.

To conclude, even though the throughput is the same in all four cases (scenarios 1 and 2, DR and AR), the location of the root of congestion is important as it affects how congestion spreads in the network. For DR the congestion is confined to a reduced number of links. On the one hand, Scenario 1 contains the congested traffic as the congestion root is in the same switch as the sources. On the other hand, when the congestion root is placed in a remote switch, as in Scenario 2, it affects more links due to the backpressure. However, the effects of congestion are worse when we use AR. In these scenarios, congestion affects all uplinks from the source remote switch/es and the destination switch downlinks.

Scenarios 3 and 4 both represent hot-spot scenarios where one or two of the senders are closer, in the number of hops, to the receiver than the rest of the senders. In scenario 3 (Figure 9a), we start with one local sender connected to the same leaf switch as the receiver and one remote sender connected to a distant leaf switch. Then we increase the number of remote senders over time, one by one, up to a total of three remote senders, all connected to the same remote leaf switch. Scenario 4 is similar, but with two local senders and two remote senders on two different leaf switches (Figure 10a).

Figure 9 shows the achieved throughput for Scenario 3. Starting with DR (Figure 9b), during the first 5 seconds, when we have one remote sender (node 2) and one local sender (node 5) for destination node 4, throughput is equally divided between the two flows. However, as we add more remote flows, these flows end up sharing half the bandwidth of the last link towards the destination node while the local sender hogs the other half. With two remote flows they get a quarter of the total bandwidth each, and with three remote flows they get one sixth of the bandwidth each. This is an example of the well-known parking lot problem [5]: At LID 7, the remote flows share the same ingress port as they all arrive from LID 6, while the local sender is the sole user of its own ingress port. Thus, a fair arbiter at LID 7, alternating between the two given ingress ports when forwarding traffic to node 4, will give half of the bandwidth to the local sender and the other half to the remote flows. When using AR (Figure 9c), the parking lot problem is still an issue, but only after 10 seconds as the number of remote flows surpasses the number of downlinks towards node 4 through the leaf switch with LID 7. When this happens, i.e., the number of remote flows are higher than the number of ingress ports used by the remote flows at LID 7, the flow from node 5 again gets an advantage when it



(a) Scenario 3 (C2) (b) Deterministic (c) Adaptive

Fig. 9: Results for Scenario 3 (C2).

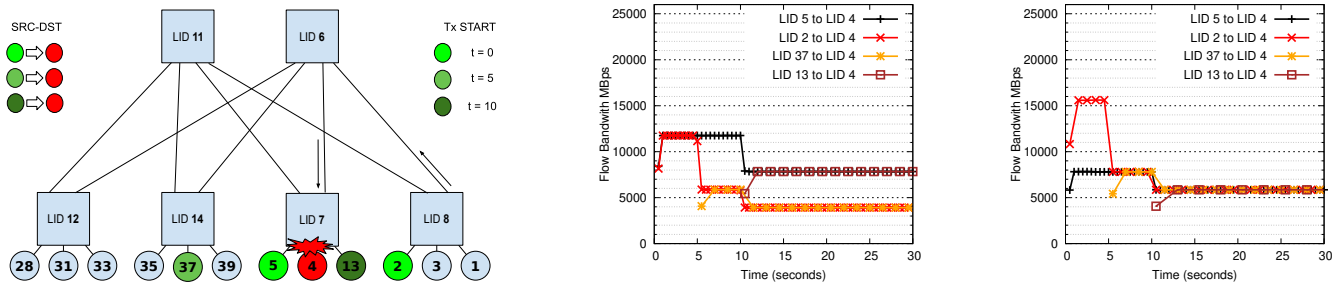
comes to arbitration and receives more than its fair share of the resources. On the contrary, during the first 5 seconds, where only one remote flow is present, this flow achieves twice the bandwidth of the local flow. The LID 7 arbiter is still fair from a local perspective, but during this time period the remote flow arrives on two ingress ports while the local flow uses a single one. The result is that the remote flow is given 2/3 of the bandwidth towards node 4, while the rest is given to the local flow. However, as already detailed, as more remote flows are added the throughput for each remote flow decreases as is the case with DR. The total throughput for all flows are the same for DR and AR, but for AR the variance is significantly lower and therefore better fairness is achieved in this case.

The results for scenario 4 with two local and two remote senders are shown in Figure 10. As in scenario 3, for DR we observe that the throughput for remote flows are affected when more remote flows are added (Figure 10b, from 5 to 10 sec.), but we also see that the local senders are naturally affected when more local senders are added (Figure 10b, after 10 sec.). Again, unfairness is created between the remote and local flows due to the parking lot problem. With AR this unfairness is completely eliminated as can be observed in Figure 10c. However, if the ratio between the local and remote flows changes or the ratio between the links in the AR group and the remote flows changes, unfairness will be reintroduced as further elaborated on in Section V. Still, the variance will remain lower for the per flow throughput for AR compared to DR and thus, better fairness is achieved in this particular case.

Finally, scenarios 5 and 6 represent hot-spot situations where we include non-congested flows to see how AR and DR affect network performance in such cases. Scenario 5 has three flows, two remote senders sending to the hot receiver (1 and 37 to 4) and one remote sender sending to a non-congested receiver (39 to 28), as shown in Figure 11a. This last sender is connected to the same leaf switch as one of the senders sending to the hot receiver, thus they are potentially sharing uplinks from LID 14 to one or both of the root switches. Scenario 6, shown in Figure 12a, is similar but with more flows of both types to better understand how an increasing number of flows are affected by AR.

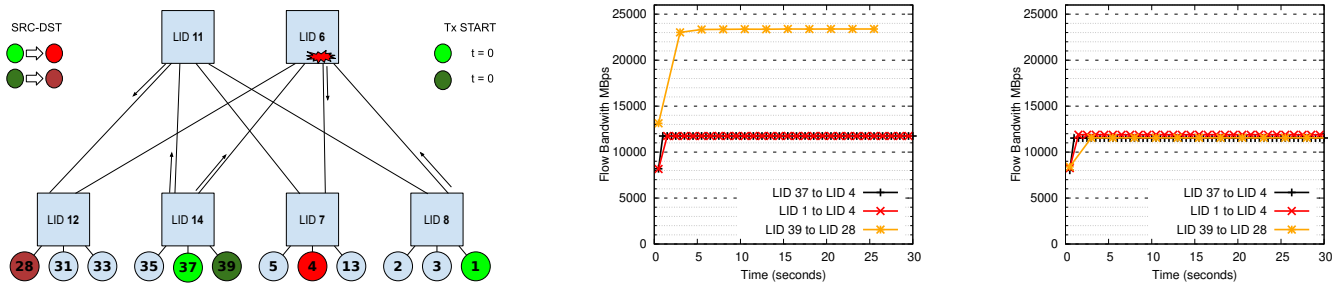
In the results for scenario 5, shown in Figure 11, we see what happens when we add a first potential victim flow to the topology, from node 39 to 28, i.e. a flow not destined for a hot receiver. With DR the two flows towards 4 create a root of congestion in root switch LID 6, but the flow from 39 to 28 is unaffected by this congestion as it follows a different path towards 28. The result is that all three flows are able to realize the full potential of the topology, with the flow from 39 to 28 achieving maximum performance, while the two flows destined for node 4 evenly share the bandwidth of the last link towards this hot receiver. With AR, however, a root of congestion is created at the node 4 egress port of LID 7, with branches of the corresponding congestion tree growing along all AR paths from node 1 and 37 towards node 4. In particular, the congestion tree will cover both uplinks from LID 14 to the root switches, and consequently cause head of line blocking for the flow from 39 to 28 which then becomes a victim of congestion. In other words, while DR limits congestion by using path diversity to route different destinations across different links, AR spreads congestion across all the links in the AR group. In our particular case, this congestion spreading leads to a significant drop in performance for the flow going from 39 to 28, and while the variance for AR is significantly lower than for DR, this fairness is achieved at the cost of losing one quarter of the total throughput.

Scenario 6 is an extension of scenario 5. This scenario starts from a situation of congestion and then additional flows are added. From time 5 to 10, the situation is the same as in scenario 5. Then at time 10 a new flow is added from 13 to 3, where the source node shares the leaf switch with the congested node and the destination node shares the leaf switch with a contributor to congestion. As this new flow is using inter-switch links in the opposite direction of the existing flows the maximum bandwidth is achieved independent of DR or AR – as expected. Finally, a third flow is added at time 15 (5 to 31) where the source node shares the switch with the congested destination and the destination node shares the switch with the destination affected by HoL blocking (congestion spreading) when AR is used. However, as this new flow is using uplinks in the opposite direction of the branches of the congestion tree, and subsequently downlinks shard what a victim of congestion, the flow achieves full bandwidth. This confirms that the victim



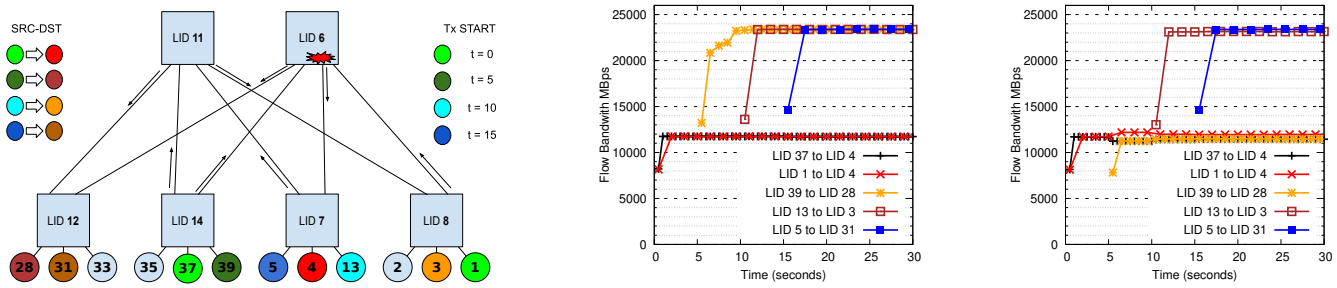
(a) Scenario 4 (C2) (b) Deterministic (c) Adaptive

Fig. 10: Results for Scenario 4 (C2).



(a) Scenario 5 (C2) (b) Deterministic (c) Adaptive

Fig. 11: Results for Scenario 5 (C2).



(a) Scenario 6 (C2) (b) Deterministic (c) Adaptive

Fig. 12: Results for Scenario 6 (C2).

of congestion (flow from 39 to 28) is head of line blocked in the upward direction only.

C. Latency results from Configuration 2

Figures 13, 14 and 15 show the average latency results in microseconds for the all to all, all reduce and all gather benchmarks in the OSU Micro benchmarks suite running in C2. The benchmark measures the average latency of the MPI_Bcast collective operation across N processes, for various message lengths, over a large number of iterations. Figure 13 shows a 20% reduction in latency between DR and AR for large message sizes. DR shows longer latency times than AR when we run the benchmark using C2 (2L in Figure 13). Modifying the C2 configuration by adding two more parallel links between the leaf and root switches makes the Fat tree oversubscribed and the differences between AR and DR disappear (4L in Figure 13) and we can use them as a baseline to compare against AR 2L. The remaining scenarios show no significant differences between DR and AR.

V. Fairness Analysis

As it has been described in Section II-A, the parking lot problem produces unfairness in the switch arbiters. For instance, this issue is clearly visible in the previous section, configuration 2, scenarios 3 and 4 (Figure 9b and 10b). In both cases, flows coming from remote senders are sharing an ingress port at switch LID 7, and as a consequence they achieve a lower throughput than the senders directly connected to switch LID 7.

On the contrary, the introduction of AR has the potential of reversing the (un)fairness situation of the parking lot problem, where sending traffic from a distance could be an advantage

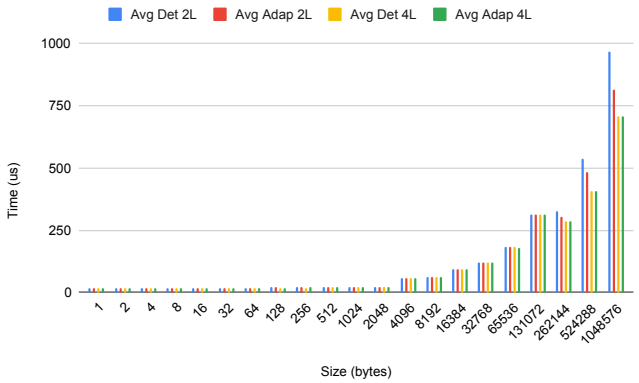


Fig. 13: C2 - OSU MPI All-to-All latency test

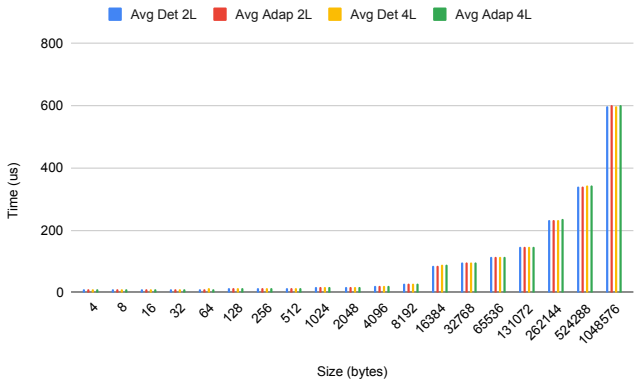


Fig. 14: C2 - OSU MPI AllReduce latency test

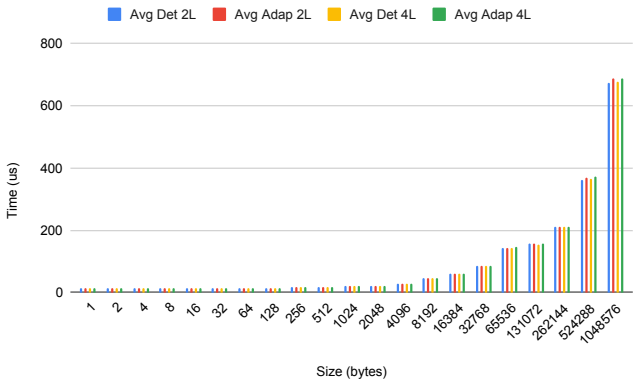


Fig. 15: C2 - OSU MPI AllGather latency test

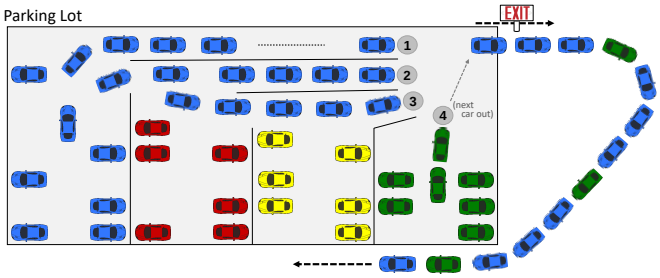


Fig. 16: The reverse parking lot problem – Case 1

if multiple paths exist. The reason is that AR will distribute flows onto available paths for a given source-destination pair. As a consequence, packets belonging to the same traffic flow can concurrently be present at several ingress ports at a given downstream switch, resulting in an advantage over flows present at fewer ingress ports. To illustrate this situation, we return once more to the parking lot, but this time with some new lanes added, as shown in Figure 16. Now, each section of the parking lot has its own lane (lanes 1–4), but with the additional rule that cars from a section to the left is allowed to use lanes added to the right of the section (but not the other way around). This is comparable to a network scenario where distant nodes potentially have more alternative paths towards a given destination than nodes closer to the destination, however possibly then sharing parts of the paths with traffic from other sources. Firstly, consider the situation where only cars from the blue and green sections are leaving the parking lot, i.e., they are the only ones headed for the exit (Figure 16). As the blue cars will be present in lanes 1, 2 and 3, while the green cars only will be present in lane 4, the resulting unfairness at the exit point is obvious. Cars from the blue section will get 3/4 of the capacity of the exit road, while the remaining 1/4 is given to the cars from the green section. Thus, with multiple lanes toward the exit of the parking lot, the fairness issues of the traditional parking lot problem is turned upside-down. We refer to this new challenge as the *reverse* parking lot problem. Returning to configuration 2, scenarios 3 and 4, the effect of the reverse parking lot problem is clearly visible in the AR results during the first 5 seconds (Figure 9c and 10c). In both scenarios, traffic from the remote sender uses two ingress ports at switch LID 7, while the local sender uses only one.

Finally, consider the reverse parking lot problem in a situation where cars from all sections want to leave the parking lot. This case is depicted in Figure 17. At each section where a new lane is added, the arbitration scheme is still locally fair, interleaving local cars and cars from sections to the left. In this specific example, on average the cars from the blue area will get access to the exit road 7/16 of the time, ie., they will get almost half of the capacity of the exit road. On the other hand, cars from the yellow section is the most unfortunate ones, left with only 1/8 of the exit capacity. In between, we find the cars from the red and green sections, left with 3/16 and 1/4 of the exit road capacity, respectively. Note that in this scenario, cars from the section farthest way from the exit are the most fortunate ones, while the ones closest to the exit, the green ones, are the runner-ups. Indeed, the exact end result at the point of exit depends on the number of lanes available and the number of sections with leaving cars at any point in time.

Nonetheless, AR can improve fairness under certain conditions. In the following we study the AR fairness behaviour and compare it to DR for a canonical K-ary 2-level Real Life Fat Tree (RLFT), where each leaf switch has K links in the

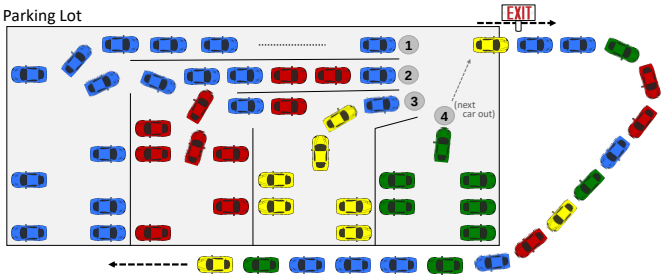


Fig. 17: The reverse parking lot problem – Case 2

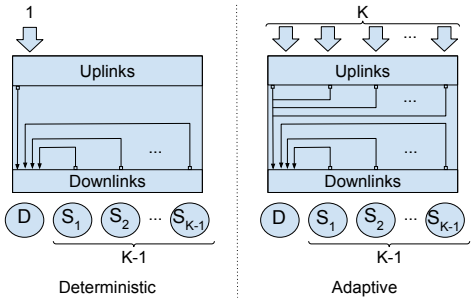


Fig. 18: First stage downstream phase (S1D) - DR vs AR

TABLE II: Number of ingress ports arbitrated per egress port in a 2-stage RLFT, using different routing algorithms, only shortest paths considered. SX identifies a switches at stage X. U denotes the upward phase and D the downward phase.

Routing	S1U	S2D	S1D
Deterministic - Total	K	$2K-1$	K
Deterministic - Uplinks	0	0	1
Deterministic - Downlinks	K	$2K-1$	$K-1$
Aaptive - Total	K	$2K-1$	$2K-1$
Adaptive - Uplinks	0	0	K
Adaptive - Downlinks	K	$2K-1$	$K-1$

downward direction and K links in the upward direction, while each root switch has 2K links in the downward direction only.

Table II shows the number of ingress ports arbitrated per egress port using DR and AR. The columns S1U and S1D give the number of ingress ports arbitrated per egress port at a leaf switch when forwarding traffic in the upwards and downwards direction, respectively, while S2D gives the number of ingress ports arbitrated per egress port at a root switch when forwarding traffic downwards (downwards is the only option at the root switches). The 'Total' row is the sum of the 'Uplinks' and 'Downlinks' rows for each routing algorithm. Overall, the table shows the differences between DR and AR when it comes to arbitration during the upward and downward routing phases of the RLFT stages.

The upward phase S1U and downward phase S2D maintain the same number of ingress ports arbitrated for both DR and AR. However, the downstream phase S1D shows differences as further detailed in Figure 18. In the deterministic case, the number of ingress ports to arbitrate for a given egress port counts the K-1 Downlinks (traffic from below) plus a single Uplink (traffic from above) i.e., the single path used by DR to forward traffic from remote subtrees to the destination in question. On the other hand, AR balances the flows between more uplinks during the upstream phase S1U. This leads to differences in the corresponding downstream phase S1D of AR, increasing the number of ingress ports from root switches potentially requesting the same egress port as S1D from 1 to K. This is the feature that potentially solves the parking lot problem, but at the same time possibly makes the reverse parking lot problem materialize. That is, the arbitration could still become unfair considering traffic flows from both upward and downward ports. Note that the flows coming from nodes connected to the same leaf switch always have the same fraction of access to the requested egress port (1/N), where N represents the number of ingress ports requesting the egress port, limited to 2K-1 ports. The fraction of access to the requested egress port of flows coming from other subtrees is K/N. We recognize three different situations depending on the number of flows coming from other subtrees, i.e., remote flows (F_{remote}):

- $F_{remote} < K$: Remote flows will have an advantage over local flows (the reverse parking lot problem is present)
- $F_{remote} == K$: Remote flows and local flows will share the egress port evenly.
- $F_{remote} > K$: Local flows will have an advantage over remote flows (the parking lot problem is present)

Scenario 3 shown in Figure 9 illustrates the three different situations described. Specifically, the results in Figure 9c show that during the first 5 seconds the external flow obtains two times the bandwidth of the internal flow (2/3 vs 1/3 of the egress link capacity) because the number of incoming remote flows is less than the number of ingress ports from root switches ($F_{remote} < K$). During the next 5 seconds, all flows have the same bandwidth (1/3) since the number of external flows is 2 (i.e., $F_{remote} == K$). During the last five seconds, the local flows maintain the bandwidth while the remote flows reduce the bandwidth as the number of remote flows is 3 ($F_{remote} > K$). On the contrary, DR results, shown in Figure 9b, maintains the bandwidth of the local flow while the remote flows are suffering from the parking lot problem.

VI. Conclusions

Adaptive routing has recently been introduced in InfiniBand and in this paper we evaluated AR in two different topologies using Infiniband HDR switches. Our main findings are the following:

- AR improves network load balancing. When there are more than one path between source and destination nodes, AR balances the use of the egress ports reducing contention.
- DR in Fat trees promotes unfairness by favouring sources closer to the destination, also known as the parking lot problem. Under certain condition AR can eliminate this

unfairness, but it can also introduce unfairness by favouring sources further away from the destination, which we label the reverse parking lot problem.
- In congestion scenarios AR spreads congestion to a larger part of the network than what is the case with DR routing.

In future work we will evaluate the use of ARN as an extension to the grouped adaptive routing we have evaluated in this paper. We will also evaluate the benefits of using AR/ARN in combination with InfiniBand congestion control.

Acknowledgements

The experiments in this paper were performed using the *Experimental Infrastructure for Exploration of Exascale Computing* (eX^3) [16], funded by the Research Council of Norway (contract 270053).

References

[1] B. Abali, C. B. Stunkel, J. Herring, M. Banikazemi, D. K. Panda, C. Aykanat, and Y. Aydogan, "Adaptive routing on the new switch chip for IBM SP systems," *J. Parallel Distributed Comput.*, vol. 61, no. 9, pp. 1148–1179, 2001. [Online]. Available: https://doi.org/10.1006/jpdc.2001.1747

[2] A. Bhatele, N. Jain, W. D. Gropp, and L. V. Kalé, "Avoiding hot-spots on two-level direct networks," in *Conference on High Performance Computing Networking, Storage and Analysis, SC 2011, Seattle, WA, USA, November 12-18, 2011*, S. A. Lathrop, J. Costa, and W. Kramer, Eds. ACM, 2011, pp. 76:1–76:11. [Online]. Available: https://doi.org/10.1145/2063384.2063486

[3] I. Borshteen, M. Kagan, N. Bloch, A. Shachar, H. Chapman, D. Bohrer, and D. Crupnicoff, "Handling transport layer operations received out of order," US Patent US20 150 172 226A1, Jun 18, 2015.

[4] C. Camarero, E. Vallejo, and R. Beivide, "Topological Characterization of Hamming and Dragonfly Networks and Its Implications on Routing," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 11, no. 4, pp. 39:1–39:25, Dec. 2014.

[5] W. J. Dally and B. Towles, *Principles and practices of interconnection networks*. Morgan Kaufmann, 2004.

[6] A. Daryin and A. Korzh, "Early evaluation of direct large-scale infiniband networks with adaptive routing," *Supercomput. Front. Innov.: Int. J.*, vol. 1, no. 3, p. 56–69, oct 2014. [Online]. Available: https://doi.org/10.14529/jsfi140303

[7] G. Faanes, A. Bataineh, D. Roweth, T. Court, E. Froese, R. Alverson, T. Johnson, J. Kopnick, M. Higgins, and J. Reinhard, "Cray cascade: a scalable HPC system based on a dragonfly network," in *SC Conference on High Performance Computing Networking, Storage and Analysis, SC '12, Salt Lake City, UT, USA - November 11 - 15, 2012*, J. K. Hollingsworth, Ed. IEEE/ACM, 2012, p. 103. [Online]. Available: https://doi.org/10.1109/SC.2012.39

[8] M. García, E. Vallejo, R. Beivide, M. Valero, and G. Rodríguez, "OFAR-CM: efficient dragonfly networks with simple congestion management," in *IEEE 21st Annual Symposium on High-Performance Interconnects, HOTI 2013, Santa Clara, CA, USA, August 21-23, 2013*. IEEE Computer Society, 2013, pp. 55–62. [Online]. Available: https://doi.org/10.1109/HOTI.2013.16

[9] P. Gaughan and S. Yalamanchili, "Adaptive routing protocols for hypercube interconnection networks," *Computer*, vol. 26, no. 5, 1993.

[10] E. G. Gran, E. Zahavi, S.-A. Reinemo, T. Skeie, G. Shainer, and O. Lysne, "On the relation between congestion control, switch arbitration and fairness," in *2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 2011, pp. 342–351.

[11] Z. Haramaty, E. Zahavi, F. Gabbay, D. Crupnicoff, A. Marelli, and G. Bloch, "Adaptive routing using inter-switch notifications," US Patent US20 140 211 631A1, Apr 21, 2015.

[12] InfiniBand Trade Association., "InfiniBandTM Architecture Specification - Release 1.0," Oct. 2000.

[13] N. Jiang, J. Kim, and W. J. Dally, "Indirect adaptive routing on large scale interconnection networks," in *36th International Symposium on Computer Architecture (ISCA 2009), June 20-24, 2009, Austin, TX, USA*, 2009, pp. 220–231.

[14] J. Kim, W. J. Dally, J. Dally, and D. Abts, "Adaptive routing in high-radix clos network," in *SC '06: Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*, 2006, pp. 7–7.

[15] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-Driven, Highly-Scalable Dragonfly Topology," in *35th International Symposium on Computer Architecture (ISCA) 2008, June 21-25, 2008, Beijing, China*, 2008, pp. 77–88.

[16] S. R. Laboratory. (2021) The eX^3 homepage. [Online]. Available: https://www.ex3.simula.no/

[17] D. Linder and J. Harden, "An adaptive and fault tolerant wormhole routing strategy for k-ary n-cubes," *IEEE Transactions on Computers*, vol. 40, no. 1, pp. 2–12, 1991.

[18] O. Lysne, J. M. Montanana, J. Flich, J. Duato, T. M. Pinkston, and T. Skeie, "An efficient and deadlock-free network reconfiguration protocol," *IEEE Transactions on Computers*, vol. 57, no. 6, pp. 762–779, 2008.

[19] J. Martínez, J. Flich, A. Robles, P. López, and J. Duato, "Supporting adaptive routing in iba switches," *Journal of Systems Architecture*, vol. 49, pp. 441–456, 11 2003.

[20] M. D. May, P. W. Thompson, and P. H. Welch, *Networks, Routers and Transputers*. IOS Press, 1994.

[21] Mellanox. (2021) How To Configure Adaptive Routing and SHIELD (New). [Online]. Available: https://community.mellanox.com/s/article/How-To-Configure-Adaptive-Routing-and-SHIELD-New

[22] M. N. Newaz, M. A. Mollah, P. Faizian, and Z. Tong, "Improving adaptive routing performance on large scale megafly topology," in *The 21st IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing, CCGrid, May 10-13, 2021, Melbourne, Victoria, Australia*. IEEE/ACM, 2021, p. 1.

[23] C. G. Requena, F. G. Villamón, M. E. Gómez, P. López, and J. Duato, "Deterministic versus Adaptive Routing in Fat-Trees," in *21th International Parallel and Distributed Processing Symposium (IPDPS) 2007, Proceedings, 26-30 March 2007, Long Beach, California, USA*, 2007, pp. 1–8.

[24] J. Rocher-Gonzalez, J. Escudero-Sahuquillo, P. J. García, and F. J. Quiles, "On the impact of routing algorithms in the effectiveness of queuing schemes in high-performance interconnection networks," in *25th IEEE Annual Symposium on High-Performance Interconnects, HOTI2017, Santa Clara, CA, USA, August 28-30, 2017*, 2017, pp. 65–72. [Online]. Available: https://doi.org/10.1109/HOTI.2017.16

[25] S. L. Scott and G. Thorson, "The cray t3e network: Adaptive routing in a high performance 3d torus," in *Hot Interconnects, August 1996*, 1996, pp. 147–156.

[26] D. D. Sensi, S. D. Girolamo, K. H. McMahon, D. Roweth, and T. Hoefler, "An in-depth analysis of the slingshot interconnect," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2020, Virtual Event / Atlanta, Georgia, USA, November 9-19, 2020*, C. Cuicchi, I. Qualters, and W. T. Kramer, Eds. IEEE/ACM, 2020, p. 35. [Online]. Available: https://doi.org/10.1109/SC41405.2020.00039

[27] A. Shpiner, Z. Haramaty, S. Eliad, V. Zdornov, B. Gafni, and E. Zahavi, "Dragonfly+: Low cost topology for scaling datacenters," in *3rd IEEE International Workshop on High-Performance Interconnection Networks in the Exascale and Big-Data Era, HiPINEB@HPCA 2017, Austin, TX, USA, February 5, 2017*, J. Escudero-Sahuquillo and P. J. García, Eds. IEEE Computer Society, 2017, pp. 1–8. [Online]. Available: https://doi.org/10.1109/HiPINEB.2017.11

[28] A. Singh, W. Dally, B. Towles, and A. Gupta, "Globally adaptive load-balanced routing on tori," *IEEE Computer Architecture Letters*, vol. 3, no. 1, pp. 2–2, 2004.

[29] S. S. Vazhkudai, B. R. de Supinski, and et al, "The design, deployment, and evaluation of the coral pre-exascale systems," in *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis*, 2018, pp. 661–672.

[30] E. Zahavi, I. Keslassy, and A. Kolodny, "Distributed adaptive routing for big-data applications running on data center networks," in *Symposium on Architecture for Networking and Communications Systems, ANCS '12, Austin, TX, USA - October 29 - 30, 2012*, T. Wolf, A. W. Moore, and V. K. Prasanna, Eds. ACM, 2012, pp. 99–110. [Online]. Available: https://doi.org/10.1145/2396556.2396578