Modeling UGAL on the Dragonfly Topology

Md Atiqul Mollah[1], Peyman Faizian[1], Md Shafayat Rahman[1], Xin Yuan[1], Scott Pakin[2], and Michael Lang[2]

[1] Florida State University, Tallahassee, FL
{mollah,faizian,rahman,xyuan}@cs.fsu.edu,
[2] Computer, Computational, and Stat. Sci. Div.
Los Alamos National Laboratory, Los Alamos, NM
{pakin,mlang}@lanl.gov}

Abstract. The Dragonfly topology has been proposed and deployed as the interconnection network topology for next-generation supercomputers. Practical routing algorithms developed for Dragonfly are based on a routing scheme called *Universal Globally Adaptive Load-balanced routing with Global information* (UGAL-G). While UGAL-G and UGAL-based practical routing schemes have been extensively studied, all existing results are based on simulation or measurement. There is no theoretical understanding of how the UGAL-based routing schemes achieve their performance on a particular network configuration as well as what the routing schemes optimize for. In this work, we develop and validate throughput models for UGAL-G on the Dragonfly topology and identify a robust model that is both accurate and efficient across many Dragonfly variations. Given a traffic pattern, the proposed models estimate the aggregate throughput for the pattern accurately and effectively. Our results not only provide a mechanism to predict the communication performance for large scale Dragonfly networks but also reveal the inner working of UGAL-G, which furthers our understanding of UGAL-based routing on Dragonfly.

1 Introduction

The Dragonfly topology features a cost-effective interconnect design. It is scalable and supports high aggregate throughput capacity at a lower cost in comparison to other alternatives such as fat-trees [1]. Dragonfly has been deployed in the Cray Cascade architecture [2] and in current supercomputers such as Cori [3] and Trinity [4].

To achieve high performance in the Dragonfly topology, different routing schemes must be used for different traffic patterns [1]. In particular, minimal routing (MIN) is better suited to uniform traffic while non-minimal Valiant Load-balanced routing (VLB) is essential for achieving good performance on adversarial traffic patterns. To unify the two routing schemes in one system, the Universal Globally Adaptive Load-balanced routing (UGAL) [1] was developed to adapt the routing decision for each packet between MIN and VLB paths based on the occupancy of packet queues [5]. The theoretical UGAL with perfect global link state information (UGAL-G) achieves high performance on Dragonfly [1], and performs similarly as MIN for uniform traffic and as VLB for adversarial traffic.

While UGAL-G is an ideal scheme that cannot be perfectly implemented, it is the foundation of practical routing schemes developed for Dragonfly [2, 6]. These practical adaptive routing schemes, including the one used in Cray Cascade [2], are based on UGAL and approximate the performance of UGAL-G. As such, the performance characteristics of UGAL-G is representative of all UGAL-based adaptive routing schemes.

Although UGAL-G and UGAL-based routing schemes have been extensively studied, all existing results are obtained through simulation and measurement. To the best of our knowledge, no theoretical model for UGAL-based routing has been developed. As such, the theoretical understanding of UGAL is lacking. For example, it is unclear how effectively these routing schemes can utilize the path diversity of a given network configuration and how sensitive the routing performances are to any change in local as well as in global network connectivity. An analysis of UGAL-G along this direction provides useful information to the problem of provisioning links and bandwidths on different Dragonfly designs.

In this work, we develop effective throughput models using linear programming (LP) for UGAL-G on the Dragonfly topology and identify a robust model for many Dragonfly variations that is both accurate and efficient. There are several theoretical as well as practical implications of our contribution. First, our proposed theoretical throughput models can accurately and efficiently predict the aggregate throughput for large scale Dragonfly networks. Second, the models reveal the implicit rate allocation in UGAL-G and thus, further our understanding of UGAL-based routing schemes. Third, the proposed models can be applied in many practical situations. For example, the models allow for efficiently exploring the design space of potential Dragonfly configurations and thus, enabling faster design prototyping before a detailed simulation on selected designs is performed. The models also give rate allocation that is competitive with UGAL-G. They can be applied to solve traffic engineering optimization problems in Software Defined Networking(SDN) architectures [7] to find rate allocation schemes that are competitive to adaptive routing in the SDN environment.

Given a traffic pattern and a Dragonfly topology, our models estimate the aggregate throughput for the pattern under the *maximum concurrent flow (MCF) model*, which is commonly used to model the throughput performance of interconnects [8–11]. The models are validated through simulations with a flit-level simulator, Booksim [12]. The results demonstrate that to accurately model UGAL-G, the LP formulations need only a small number of variables per flow. This enables the models to be used for large-scale systems with tens of thousands of flows. The study also reveals that even with the precise global network state information, UGAL-G does not have effective control over all the paths that are available and does not allocate rates to individual paths to maximize its performance. Instead, for the general cases when the numbers of MIN and VLB paths are sufficiently large, UGAL-G effectively allocates rates to groups of paths instead of individual paths.

The rest of the paper is structured as follows. Section 2 discusses the background of this work, describing the Dragonfly topology, its variation in Cray Cascade, UGAL-G routing, and the MCF throughput model. Section 3 introduces our performance models for UGAL-G on Dragonfly. Section 4 presents the results of a set of experiments used

to validate the models. Section 5 discusses related work. Finally, in Section 6 we draw some conclusions from our work.

2 Background

2.1 Dragonfly topology

We will briefly introduce the Dragonfly topology. More details about the topology can be found in Kim et al.'s original paper [1]. The Dragonfly topology has a 2-layer structure. A group of low-radix routers/switches are interconnected with an intra-group topology into a *group* that works as a single virtual router with a very high radix. In this paper, the terms router and switch will be used interchangeably. The groups are then connected with some inter-group topology. Figure 1 shows an example of the 2-layer Dragonfly topology. In this example, each group consists of 4 switches; there are a total of 9 groups in the system.

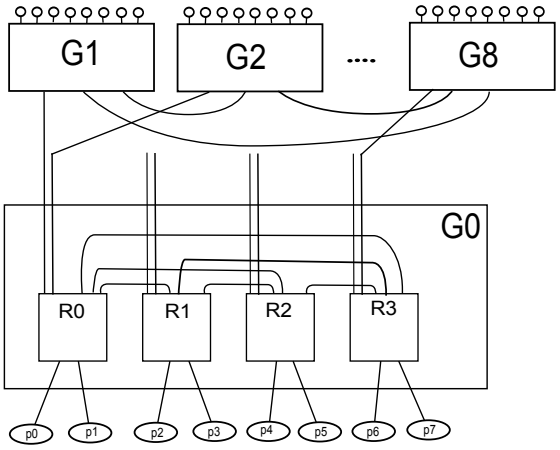


Fig. 1. Dragonfly Architecture(p=h=2,a=4,g=9)

Various topologies can be used to form the intra-group connectivity. A typical intra-group topology is a fully connected graph where all pairs of switches are directly connected [1]. An example of such an intra-group topology is shown in the G0 group in Figure 1. The groups in a Dragonfly are also fully connected where there is at least one *global* link connecting each pair of groups. Such a topology is uniquely defined by four parameters: the number of links per switch connecting to local compute nodes p, the number of switches in each group a, the number of global links per switch connecting to switches in other groups h, and the number of groups g. In a fully connected Dragonfly group, the number of links per switch connecting to local switches is $a-1$. We will use the 4-tuple notation $dfly(p,a,h,g)$ to denote such a topology and 3-tuple notation $group(p,a,h)$ to denote an individual Dragonfly group. By definition, the number

of ports in each switch in $dfly(p,a,h,g)$ is $p+a-1+h$; the number of global links from each group is $a \times h$, and the number of groups, g, is thus at most $a \times h+1$. The number of global links between each pair of groups is $a \times h/(g-1)$. The total number of switches and the total number of compute nodes in $dfly(p,a,h,g)$ is $a \times g$ and $p \times a \times g$ respectively. As discussed in [1], a load-balanced Dragonfly system should have $a = 2p = 2h$. Figure 1 illustrates a balanced system with the largest possible group count $dfly(p = 2, a = 4, h = 2, g = 9)$. In this case, each group has $a = 4$ switches and $a \times h = 8$ global links with $a \times h/(g-1) = 1$ global link connecting to each of other groups.

2.2 Cray Cascade topology

The Cray Cascade architecture employs Dragonfly as its topology [2]. It has a well-defined structure for each group, but allows a variable number of groups to form a system.

Unlike $dfly(p,a,h,g)$, switches in a Cray Cascade group are not fully connected. Every group in Cascade is formed of a pair of cabinets. Each cabinet houses three chassis. Each chassis contains 16 blades. Each blade connects a single *Aries* router and four compute nodes. Each chassis backplane provides all-to-all connections among sixteen Aries routers. Each router is also connected to five other routers in the remaining five chassis within the same group. Each inter-chassis link is equivalent to three intra-chassis links in terms of bandwidth. Each Aries router has a total of 48 ports: 8 ports for local compute nodes, 15 ports connecting to 15 routers in the same chassis, 15 ports to 5 routers in the same slot but different chassis, and 10 ports to other groups. Figure 2 shows the interconnect topology of a single Cascade group. Logically, a cascade group consists of a 6×16 mesh with fully connected X and Y dimensions. Each pair in the same row is connected by one link while each pair in the same column is connected by three links.

In practice, the number of global links connecting a pair of groups in Cascade can be configured. For example, in the NERSC Edison supercomputer, there are 24 global links (spreading among multiple pairs of switches) connecting each pair of groups [13]. The details about how the global links are connected can be quite involved. The Cascade topology that we consider in this paper is a six-group system whose connectivity is directly read from the connectivity dump file for the first 6 groups of the Edison supercomputer [13].

2.3 Routing in Dragonfly and UGAL

The following terminology will be used to describe routing in Dragonfly. Packets are routed from a *source compute node* to a *destination compute node*. The switch that the source compute node connects to is called the *source switch*. The switch that the destination compute node connects to is called the *destination switch*. The group that the source compute node is in is called the *source group*; the group that the destination compute node is in is called the *destination group*. We will describe routing for a generic Dragonfly topology. The routing can also be applied to the Cascade Dragonfly variation.

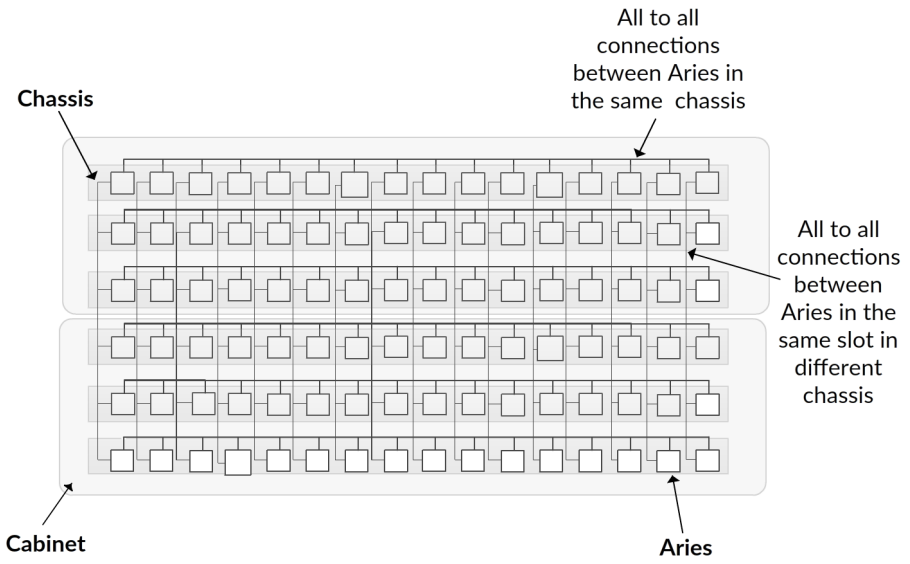


Fig. 2. Cray Cascade intra-group topology

In a Dragonfly topology, packets are routed along either a *minimal* or a *non-minimal path*. The minimal path is the shortest path from the source compute node to the destination compute node that contains at most one global link. The thick segmented line in Figure 3 shows a typical minimal path from s to d, where the path takes one local hop in the source group from the source switch to the switch that has a global link to the destination group, then the global link to the destination group, and finally a local link at the destination group to the destination switch. Depending on the positions of the source and the destination, the minimal path may have fewer hops. In $dfly(p,a,h,g)$, two routers belonging to different groups may be connected through one of the $(a \times h)/(g-1)$ global links between the two groups. Thus, there are $(a \times h)/(g-1)$ minimal paths between such router pairs.

The Minimal routing (MIN) scheme routes packets only with minimal paths. It minimizes the resource usage and works well for traffic patterns where MIN can evenly distribute the load such as the random uniform traffic. However, since the number of links between each pair of groups is typically small, for traffic patterns where many nodes in one group must communicate to many nodes in another group, the MIN routing will perform poorly since all of the traffic from one group to another must use the small number of links between the two groups. Such traffic patterns are considered adversarial.

To avoid congestion on global links for an adversarial traffic pattern, Valiant Load-balanced routing (VLB) [14] can be used to spread non-uniform traffic evenly over the set of available links. A VLB path can be considered as using MIN to find a path from the source to a randomly selected intermediate switch that is not in the source and destination groups, and then, from the intermediate switch to the destination. A VLB

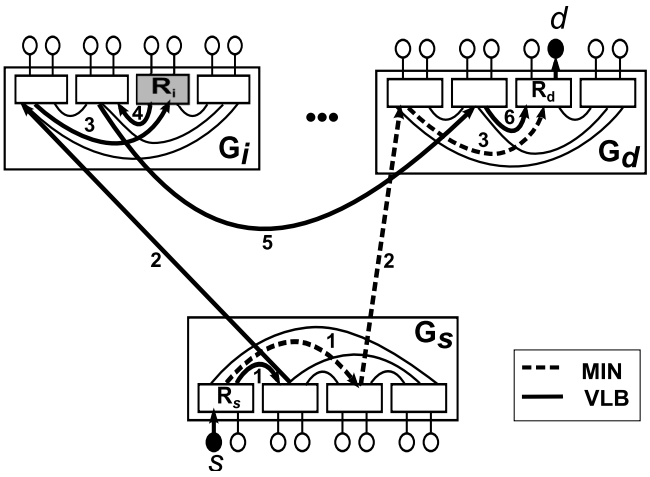


Fig. 3. MIN and VLB routing on Dragonfly

path is thus non-minimal. Figure 3 shows a 6-hop VLB path in solid thick lines. With a VLB route, a packet is first sent to an intermediate router (R_i in this example) and then to the destination. We note that the initial works on Dragonfly routing [1, 6] consider randomly selecting an intermediate group to obtain VLB paths. However, it is shown by Garcia et al. [15] that the randomly choosing a group leads to local link congestion at the intermediate group and instead, random selection of an intermediate switch is preferred. In $dfly(p,a,h,g)$, there are a total of $a \times (g-2)$ intermediate switches, $(a \times h)/(g-1)$ minimal paths from the source to each intermediate switch and again, $(a \times h)/(g-1)$ minimal paths from intermediate switch to destination. Therefore, the total number of VLB paths between two nodes of a $dfly(p,a,h,g)$ that are not in the same group is given by

$$\frac{a^3 \times h^2 \times (g-2)}{(g-1)^2} \tag{1}$$

The Universal Globally Adaptive Load-balanced routing (UGAL) selects among MIN and VLB paths for each packet based on the traffic condition. The traffic condition is inferred from the occupancy of packet queues of the network sensed at the source switch. For each packet, UGAL first randomly selects a small number of candidate MIN and VLB paths from all possible MIN and VLB paths for further consideration. In the original UGAL proposal and its Dragonfly adaptation, the number of MIN paths is 1 and the number of VLB paths is 1 [1,5]; in Cascade, 2 MIN paths and 2 VLB paths are chosen as candidates [2]. Then, UGAL selects a path from among the candidate paths for routing that would achieve the smallest packet delay. In contrast, UGAL-G assumes that the precise global network state information is available, and uses the total queue length on all links along the path to estimate the packet delay. Let TQ_{MIN} be the smallest path queue length for all MIN paths considered, and TQ_{VLB} be the smallest path queue length for all VLB paths considered. UGAL-G selects the MIN path

if $TQ_{MIN} \leq TQ_{VLB}$, and the VLB path otherwise. Other UGAL-based schemes [2, 6] rely on some practically measurable quantities such as credit-round-trip latency and piggybacked link-state information broadcast on source group to estimate the actual packet delay and approximate UGAL-G.

2.4 Maximum Concurrent Flow

Given a traffic pattern, there are various models to quantify the aggregate throughput performance. Among the throughput models, the maximum concurrent flow model is one of the commonly used models [8–11]. The Maximum Concurrent Flow (MCF) can informally be described as the maximum attainable throughput by all flows for a traffic pattern in a given network. In other words, MCF is the single largest rate that can be assigned to all flows without violating any capacity constraints. It is therefore the lower bound of the flow rates for all flows in the traffic pattern.

Without the routing constraint, the MCF rate for a given pattern on a given topology can be computed using the linear programming (LP) formulation given by Shahrokhi and Matula [8]. LP is an approach to minimize an objective function subject to a set of linear inequalities. Their linear-programming formulation considers all possible paths to route each flow. The models proposed in this work not only consider the specific UGAL routing on Dragonfly, which is constrained, but also how the paths are selected in UGAL. This allows us to develop more accurate and efficient models for UGAL on Dragonfly.

3 Performance Models for UGAL-G on Dragonfly

3.1 Notation

Let A be a set and $|A|$ be the size of the set. Let a Dragonfly network be represented as a graph $G = (V, E)$, where V is the set of nodes and E is the set of links in the network. $V = PE \cup S$ contains two types of nodes. PE is the set of compute nodes; and S is the set of switches. The nodes are numbered from 0 to $|V| - 1$. For each link $e \in E$, C_e is the link capacity.

Let $s \in PE$ and $d \in PE$. A flow from s to d is denoted as (s, d). A traffic pattern F is a set of flows. The traffic in a flow is carried over a set of paths for the flow. Each path p is represented as a set of links. For each flow, UGAL-G considers all MIN paths and all VLB paths. For a flow (s, d), $P_{s,d}^{MIN,L}$ is the set of MIN paths with path length L; $P_{s,d}^{VLB,L}$ is the set of VLB paths with path length L; $P_{s,d}^{MIN}$ is the set of all MIN paths for the flow; $P_{s,d}^{VLB}$ is the set of all VLB paths and $P_{s,d}$ is the set of all considered paths. Clearly, $P_{s,d}^{MIN} = \cup_L P_{s,d}^{MIN,L}$; $P_{s,d}^{VLB} = \cup_L P_{s,d}^{VLB,L}$; and $P_{s,d} = P_{s,d}^{MIN} \cup P_{s,d}^{VLB}$. Let $e \in E$ be a link. If a path p uses a link e, we say that $e \in p$. Given a set of paths P, $P(e)$ returns a subset of P only containing paths that use link e. Table 1 summarizes the notations.

3.2 Performance models

We use linear programming (LP) to model UGAL-G performance as an optimization problem. For accuracy, our models consider the following UGAL-G features.

$G = (V, E)$	the topology with node set V and edge set E
$C_e, e \in E$	link capacity
(s, d)	a flow from s to d
$P_{s,d}$	the set of all MIN and VLB paths for (s, d)
$P_{s,d}^{MIN}$	the set of MIN paths for (s, d)
$P_{s,d}^{MIN,L}$	the set of MIN paths of length L for (s, d)
$P_{s,d}^{VLB}$	the set of VLB paths for (s, d)
$P_{s,d}^{VLB,L}$	the set of VLB paths of length L for (s, d)
$P(e)$	$\{p \mid e \in p \text{ and } p \in P\}$

Table 1. Notation used in the models

- **Feature 1**: UGAL-G considers all MIN and VLB paths.
- **Feature 2**: UGAL-G randomly selects a small number of MIN and VLB paths as candidate paths for each packet.
- **Feature 3**: UGAL-G implicitly differentiates paths of different lengths. UGAL-G selects paths based on the path latency. As a result, it biases towards using shorter paths: if the queue length is the same for all links, shorter paths will have smaller aggregate queue length and are more likely to be selected by UGAL-G.

The challenge to develop accurate performance models is to capture the dominating factors in the UGAL-G routing process. UGAL-G uses an identical process to select between MIN and VLB paths. Thus, the spectrum of UGAL-G's control over MIN and VLB paths is the same. Next, we will use VLB paths to describe the potential control that UGAL-G has on paths. Consider the spectrum of UGAL-G's control over VLB paths. At one end, since UGAL-G considers all VLB paths (Feature 1), if it may have a fine-grain control at the path level, it could allocate rates for individual paths so as to maximize the aggregate throughput for a pattern. This level of control will be referred to as **individual** control. On the other end, UGAL-G randomly selects a small number of VLB paths as candidate paths for each packet (Feature 2). If the random selection dominates the performance, the routing essentially treats all VLB paths the same as a group and uniformly distribute the load to each of the paths. This level of control will be referred to as **all random** control. In general, the level of control falls in between the two extremes. Feature 3 states that UGAL-G differentiates paths of different lengths. This gives another potential level of control in between the two extremes, which we call **path-length-based random** control. In this control, the VLB paths are grouped based on their lengths. The routing scheme may allocate rates differently for different groups, but will treat paths in the same group the same. Further refinement of the levels of control is possible. However, it will be shown later that the combination of these three levels of control already yields accurate modeling.

The level of control that UGAL-G has would depend on the number of MIN and VLB paths, which is determined by the Dragonfly topology. When the number of MIN (VLB) paths is small, each MIN (VLB) path is likely considered as a candidate path for each packet; and UGAL-G can have a high level of control over the rate allocation over the MIN (VLB) paths. On the other hand, when the number of MIN (VLB) paths is very large, the chance for each MIN (VLB) path to be selected as the candidate path

Model	MIN	VLB
No. 0	individual	individual
No. 1	individual	path-length-based random
No. 2	individual	all random
No. 3	path-length-based random	path-length-based random
No. 4	path-length-based random	all random
No. 5	all random	all random

Table 2. Summary of models (Model No. 3 is a robust and efficient model for different topologies)

by UGAL-G is very small. As a result, UGAL-G will have a low level of control of the rate allocation over such paths. In between these two extremes, path-length-based random control may be more appropriate.

Given a Dragonfly topology, it is unclear which level of control UGAL-G has for the MIN and VLB paths. In general, the number of MIN paths is significantly smaller than the number of VLB paths. As such, UGAL-G will have more control over MIN paths than over VLB paths. To find a robust model that is both accurate and efficient, we develop a set of six models that applies each of the three levels of control on the two types of paths (MIN and VLB) with the assumption that UGAL-G will have an equal or higher level of control over MIN paths than over VLB paths. The models are summarized in Table 2. Our experiments indicate that Model No. 3 with path-length-based random control for both MIN and VLB paths is a robust and efficient model across many variations of Dragonfly including the Cascade topology, achieving accurate modeling results and low modeling complexity.

Model No. 0 (the upper bound, individual control on both MIN and VLB paths)

For each flow, UGAL-G considers all MIN and VLB paths. Model No. 0 assumes that UGAL-G has individual control over both MIN and VLB paths so that it can allocate the rate for each path to maximize the throughput. To model the individual control over each MIN and VLB path, each MIN or VLB path can have a different rate, which is represented as one variable in the LP formulation. Our linear programming formulation uses the edge-path formulation assuming that each path considered by UGAL-G can be assigned a different rate to maximize the MCF rate.

The LP formulation is shown in Figure 4. In this model, one variable $x_{s,d}^p$ is assigned to each path p considered by UGAL-G for a flow (s,d) in the pattern. The variable $x_{s,d}^p$ represents the rate allocated for the path. Hence, for flow (s,d), the sum of the rates allocated to all of its paths, $\sum_{p \in P_{s,d}} x_{s,d}^p$, is the flow rate. The variable α is the MCF rate for the pattern. By MCF definition, the rates for all flows must be no less than the MCF rate. The constraints in (1) ensure that the rates for all flows are no less than the MCF rate. Constraints (2) are link capacity constraints that state that for each link, the total rates for all paths that use the link, $\sum_{e \in p, p \in P_{s,d}, (s,d) \in F} x_{s,d}^p$, do not exceed the link capacity.

The formulation in Figure 4 assumes that the rate for each path can be tuned to maximize the MCF throughput, which provides an upper bound for all UGAL-based algorithms. This formulation, however, has two issues. First, solving the problem on

Maximize α

Subject to:

$$\alpha - \sum_{p \in P_{s,d}} x_{s,d}^{p} \leq 0, \ \forall (s,d) \in F \quad (1)$$

$$\sum_{e \in p, p \in P_{s,d}, (s,d) \in F} x_{s,d}^{p} \leq C_e, \forall e \in E \quad (2)$$

Fig. 4. Model No. 0: the upper bound MCF rate for all UGAL-based schemes (individual control over MIN paths and individual control over VLB paths)

reasonably sized networks becomes computationally infeasible due to the use of a large number of variables. In practical Dragonfly networks, the number of minimal paths is usually not very large, while the number of VLB paths can easily approach tens of thousands to millions. See Table 3 for Dragonfly examples with the numbers of MIN and VLB paths. This formulation can easily introduce more than one million variables for some topology. Solving LP problems of such sizes is computationally infeasible with today's technology. The second issue is that this formulation does not consider the inner working of UGAL-G such as Features 2 and 3. Thus, it may not yield accurate estimation results for UGAL-G.

Model No. 1 (individual control on MIN paths and path-length-based random control on VLB paths)

Model No. 0 would yield an accurate modeling result only if UGAL-G were capable of tuning the rate for each available MIN and VLB path in the most effective manner. In the Dragonfly topology, the number of MIN paths for each flow is usually small while the number of VLB paths can be much larger. For example, in $dfly(3,6,3,10)$, the number of VLB paths between two nodes that are not in the same group is 192 as calculated from Formula 1, while the number of MIN paths for each flow is 2. In such a situation, considering a small number of (1 or 2) VLB paths for each packet is not likely to result in effective use of VLB paths while the routing may have individual control over MIN paths since the MIN path is considered for every packet. Model No. 1 assumes individual control over MIN paths and path-length-based random control over VLB paths and targets Dragonfly networks with a small number of MIN paths and a reasonably large number of VLB paths per flow.

The LP formulation for Model No. 1 is shown in Figure 5. In this model, for each flow (s,d), a variable $x_{s,d}^{p}$ is assigned to each MIN path $p \in P_{s,d}^{MIN}$. In addition, another variable $x_{s,d}^{VLB,L}$ is assigned for all VLB paths of length L ($P_{s,d}^{VLB,L} \neq \emptyset$) of a given flow (s,d): each of the VLB paths of length L will have the same rate, $x_{s,d}^{VLB,L}$, while VLB paths of different lengths may have different rates. The LP formulation of Model No. 1 is basically the same as that of Model No. 0 except that all VLB paths of the same length L for each flow is assumed to have the same rate. $\sum_{p \in P_{s,d}^{MIN}} x_{s,d}^{p} + \sum_{P_{s,d}^{VLB,L} \neq \emptyset} |P_{s,d}^{VLB,L}| \times x_{s,d}^{VLB,L}$ is the rate allocated for flow (s,d); and Constraints (1) ensure that the rates for all flows are no less than the MCF rate. $\sum_{p \in P_{s,d}^{MIN}(e), (s,d) \in F} x_{s,d}^{p} + \sum_{P_{s,d}^{VLB,L}(e) \neq \emptyset, (s,d) \in F} |P_{s,d}^{VLB,L}(e)| \times x_{s,d}^{VLB,L}$ is the total rate allocated over link e; and Con-

straints (2) are link capacity constraints that ensure that the rate allocated over each link is no more than its capacity.

1 Maximize α
2 Subject to:
3 $$\alpha - (\sum_{p \in P_{s,d}^{MIN}} x_{s,d}^{p} + \sum_{P_{s,d}^{VLB,L} \neq \emptyset} |P_{s,d}^{VLB,L}| \times x_{s,d}^{VLB,L}) \leq 0, \ \forall (s,d) \in F \quad (1)$$
4 $$\sum_{p \in P_{s,d}^{MIN}(e),(s,d) \in F} x_{s,d}^{p} + \sum_{P_{s,d}^{VLB,L}(e) \neq \emptyset,(s,d) \in F} |P_{s,d}^{VLB,L}(e)| \times x_{s,d}^{VLB,L} \leq C_e, \forall e \in E \quad (2)$$

Fig. 5. Model No. 1: Maximize the MCF rate with the assumption that VLB paths of the same length for a flow have the same rate (individual control over MIN paths and path-length-based random control over VLB paths)

The Model No. 1 in Figure 5 will be accurate when the random selection of VLB paths (Feature 2) and the path length preferences (Feature 3) have impacts on the throughput performance. Since VLB paths have similar path lengths in Dragonfly, Model No. 1 only needs a small number of variables for VLB paths, which significantly reduces the number of variables over Model No. 0. For example, the longest VLB path in $dfly(p,a,h,g)$ is 6 hops, as shown in Figure 3. Therefore, there could be at most 6 different path lengths for all VLB paths and thus, only up to 6 variables corresponding to VLB routing is required per flow in the model LP formulation. This reduction in the number of variables enables Model 1 to be used to solve much larger problems in much larger systems.

Model No. 2 (individual control on MIN paths and all random control on VLB paths)

Model No. 1 considers the three features of UGAL-G: (1) the routing considers all MIN and VLB paths, (2) the large number of VLB paths is randomly selected for consideration for each packet, and (3) UGAL-G inherently differentiates between paths of different lengths. When the number of VLB paths is very large, the random selection of VLB paths to be considered for each packet may be the dominating factor. In this case, UGAL-G may only have the all random control over VLB paths. Model No. 2 that assumes individual control of MIN paths and all random control of VLB paths is designed for such cases.

The LP formulation for Model No. 2 is shown in Figure 6. In this model, for each flow (s,d), a variable $x_{s,d}^{p}$ is assigned to each MIN path $p \in P_{s,d}^{MIN}$. In addition, another variable $x_{s,d}^{VLB}$ is assigned for all VLB paths, that is, each of the VLB paths is assumed to have the same rate $x_{s,d}^{VLB}$. Model No. 2 is basically the same as Model No. 1 except that all VLB paths for each flow are assumed to have the same rate. Constraints (1) ensure that the rates for all flows are no less than the MCF rate. $\sum_{p \in P_{s,d}^{MIN}(e),(s,d) \in F} x_{s,d}^{p} + \sum_{P_{s,d}^{VLB}(e) \neq \emptyset,(s,d) \in F} |P_{s,d}^{VLB}(e)| \times x_{s,d}^{VLB}$ is the total rate allocated over link e which must not exceed the link capacity. Such capacity constraints are summarized in Constraints (2).

1 Maximize α

2 Subject to:

3 $$\alpha - (\sum_{p \in P_{s,d}^{MIN}} x_{s,d}^{p} + |P_{s,d}^{VLB}| \times x_{s,d}^{VLB}) \leq 0, \ \forall (s,d) \in F \quad (1)$$

4 $$\sum_{p \in P_{s,d}^{MIN}(e),(s,d) \in F} x_{s,d}^{p} + \sum_{P_{s,d}^{VLB}(e) \neq \emptyset,(s,d) \in F} |P_{s,d}^{VLB}(e)| \times x_{s,d}^{VLB} \leq C_e, \forall e \in E \quad (2)$$

Fig. 6. Model No. 2: Maximize the MCF rate with the assumption that all VLB paths for a flow have the same rate (individual control for MIN paths and all random control for VLB paths)

The Model No. 2 in Figure 6 will be accurate when the random selection of VLB paths dominates the performance. It further reduces the number of variables for each flow in comparison to Model No. 1.

Model No. 3 (path-length-based random control on MIN paths and path-length-based random control on VLB paths)

Although the number of VLB paths is always significantly larger than the number of MIN paths for each flow in a Dragonfly topology, some Dragonfly topologies can have a significant number of MIN paths. Variants of Dragonfly such as the Cascade topology that do not have a fully connected intra-group network and have high number of global links between all group pairs, fall into this category. For such topologies, UGAL-G may not have the individual control over each MIN path. Model No. 3 assumes that the control over MIN paths as well as VLB paths is path-length-based random.

The LP formulation for Model No. 3 is shown in Figure 7. In this model, for each flow (s,d), a variable $x_{s,d}^{MIN,L}$ is assigned to each group of MIN paths of length L ($P_{s,d}^{MIN,L} \neq \emptyset$). For VLB paths, a variable $x_{s,d}^{VLB,L}$ is assigned for each group of VLB paths of length L ($P_{s,d}^{VLB,L} \neq \emptyset$). $\sum_{P_{s,d}^{MIN,L} \neq \emptyset} |P_{s,d}^{MIN,L}| \times x_{s,d}^{MIN,L} + \sum_{P_{s,d}^{VLB,L} \neq \emptyset} |P_{s,d}^{VLB,L}| \times x_{s,d}^{VLB,L}$ is the rate allocated for flow (s,d). Constraints (1) describe the MCF rate constraints. $\sum_{P_{s,d}^{MIN,L}(e) \neq \emptyset,(s,d) \in F} |P_{s,d}^{MIN,L}(e)| \times x_{s,d}^{MIN,L} + \sum_{P_{s,d}^{VLB,L}(e) \neq \emptyset,(s,d) \in F} |P_{s,d}^{VLB,L}(e)| \times x_{s,d}^{VLB,L}$ is the total rate allocated over link e; and the same expression is used in Constraints (2) to summarize capacity constraints on all links.

1 Maximize α

2 Subject to:

3 $$\alpha - (\sum_{P_{s,d}^{MIN,L} \neq \emptyset} |P_{s,d}^{MIN,L}| \times x_{s,d}^{MIN,L} + \sum_{P_{s,d}^{VLB,L} \neq \emptyset} |P_{s,d}^{VLB,L}| \times x_{s,d}^{VLB,L}) \leq 0, \forall (s,d) \in F \quad (1)$$

4 $$\sum_{P_{s,d}^{MIN,L}(e) \neq \emptyset,(s,d) \in F} |P_{s,d}^{MIN,L}(e)| \times x_{s,d}^{MIN,L} + \sum_{P_{s,d}^{VLB,L}(e) \neq \emptyset,(s,d) \in F} |P_{s,d}^{VLB,L}(e)| \times x_{s,d}^{VLB,L} \leq C_e, \forall e \in E \quad (2)$$

Fig. 7. Model No. 3: Maximize the MCF rate with the assumption of path-length based control for both MIN and VLB paths

topology	# of switches	# of PEs	# of MIN	# of VLB
$dfly(2,4,2,9)$	36	72	1	28
$dfly(3,6,3,19)$	114	342	1	102
$dfly(4,8,4,33)$	264	1,056	1	248
$dfly(5,10,5,51)$	510	2,550	1	490
$dfly(5,10,5,26)$	260	1,300	2	960
$dfly(5,10,5,11)$	110	550	5	2250
$dfly(5,10,5,6)$	60	300	10	4000
Cascade	576	2,304	96	3,538,944

Table 3. Topologies used in the validation

Model No. 4 and Model No. 5

Model No. 4 assumes path-length-based random control on MIN paths and all random control on VLB paths. Model No. 5 assumes all random control on both VLB and MIN paths. These two models uses less variables than all of the earlier models. Their LP formulations are straight-forward extensions of those for Models No. 1, 2, and 3, and are omitted.

4 Model Validation

We implemented the six models for the general Dragonfly topology as well as for the Cascade topology. Each implemented model takes in a topology, a routing scheme and a traffic pattern as inputs and generates an LP formulation file. The LP formulation is then fed into IBM's CPLEX optimizer [16] to find the maximum MCF rate for each of our experiment instances.

We have also extended Booksim [12] to support UGAL-G for $dfly(p,a,h,g)$ and the Cascade topology. Then, simulation results on the same network configurations are obtained to validate the models. We assume single-flit packets and a 2.5x speedup for router crossbar over network links. The latency of each network link is set to 10 cycles. To ensure deadlock-free routing, we allocate three virtual channels for the Dragonfly topology in the same way as described in [1], and ten virtual channels for the Cascade topology. The buffer size of each virtual channel is set to 256 flits. For each data point, the network is warmed-up for 40,000 cycles and network statistics are collected for another 10,000 cycles. In Booksim, all processing nodes inject traffic to the network at a same *injection rate*. During each simulation run, We gradually increment the injection rate until the packet queues across the network becomes saturated. Once the network is saturated, we record the corresponding injection rate as the maximum concurrent throughput of that run.

The topologies considered are summarized in Table 3. Two types of topologies are used: the load-balanced Dragonfly with fully connected intra-group topology described in $dfly(p,a,h,g)$ denotation, and the 6-group Cascade topology. The difference between these two topologies is in the number of MIN and VLB paths that are available. The

number of MIN and VLB paths in $dfly(p,a,h,g)$ is $(a \times h)/(g-1)$ and $(a^3 \times h^2 \times (g-2))/(g-1)^2$ respectively, as shown in Section 2. In the Cascade topology, a packet can go in either X or Y dimension first within each group and there are 24 global links between each group pair. Hence, the number of MIN paths between two nodes in different groups can be up to $2 \times 24 \times 2 = 96$. The number of VLB paths in Cascade is much larger. Using $4 \times 96 = 384$ potential intermediate switches, the number of VLB paths for each flow can be up-to $96 \times 96 \times 384 = 3,538,944$. As discussed earlier, the number of MIN and VLB paths affects how UGAL-G controls the paths.

In the experiments on $dfly(p,a,h,g)$, one MIN path and one VLB path are randomly chosen as candidate paths for each packet, same as in the original UGAL proposal [6]. On the Cascade topology, we consider 2 MIN and 2 VLB candidate paths in consistency with the current Cascade routing scheme [2].

The results for two types of traffic patterns are reported, the random permutation patterns where each node sends to and receives from at most one other destination and source respectively, and the random shift pattern where compute node i sends to compute node $(i+x) \bmod |PE|$ where x is a random number. Results for other patterns yield similar trends.

The general observations in the experiments include the following: individual control in general overestimates the throughput; all random control in general underestimates the throughput; and the path-length-based random control gives good estimation for a wide range of Dragonfly variations. In particular, Model No. 3 that assumes path-length-based random control for both MIN and VLB paths, which has a low complexity with a small number of variables for each flow, achieves good prediction for a wide range of Dragonfly topologies (within 10% of prediction errors in all cases in our study).

Figure 8 shows the average modeling and simulation results for five random permutation patterns on maximum size $dfly(p,a,h,a \times h+1)$ networks of different sizes. For these topologies, since the number of MIN paths for each flow is only 1, Model No. 1 is equivalent to Model No. 3, and Models No. 2, 4, and 5 are equivalent. As can be seen from the figure, the throughput with UGAL-G across all topologies is significantly worse than the throughput predicted by Model No. 0. This indicates that for these topologies, UGAL-G cannot fully control the MIN and VLB paths to maximize its throughput. The figure also shows that the throughput with UGAL-G is significantly better than that predicted with Model No. 2. This indicates that UGAL-G has better control than all random over VLB paths. Across all topologies, the throughput predicted by Models No. 1 and No. 3 closely matches the simulation with the prediction errors ranging from 4.3% to 8.6%. Figure 9 shows prediction and simulation results for each individual random permutation on $dfly(2,4,2,9)$. As can be seen from the figure, the trend for the prediction with each model is exactly the same as that in Figure 8. Results on other similar $dfly(p,a,h,g)$ instances are similar.

Figure 10 shows the average modeling and simulation results for five random permutation patterns on Dragonfly topologies with the same group $group(5,10,5)$, but different numbers of groups: $dfly(5,10,5,6)$ with 6 groups, $dfly(5,10,5,11)$ with 11 groups, and so forth. These topologies have the same structure with different numbers of global links connecting each pair of groups, which affects the number of MIN and VLB

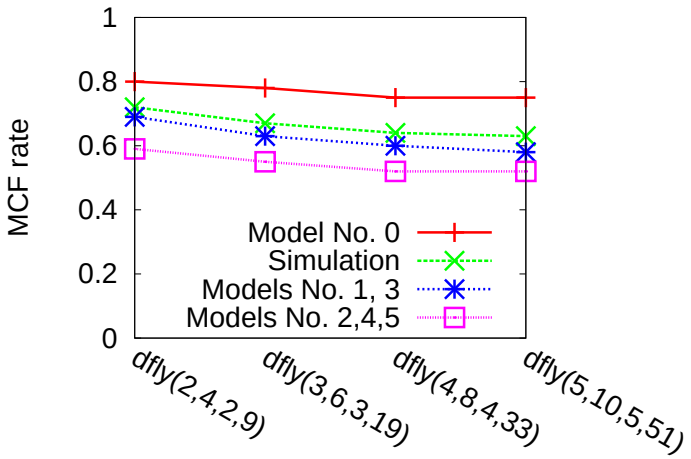


Fig. 8. The modeling and simulation results for random permutation patterns on $dfly(p,a,h,a\times h+1)$

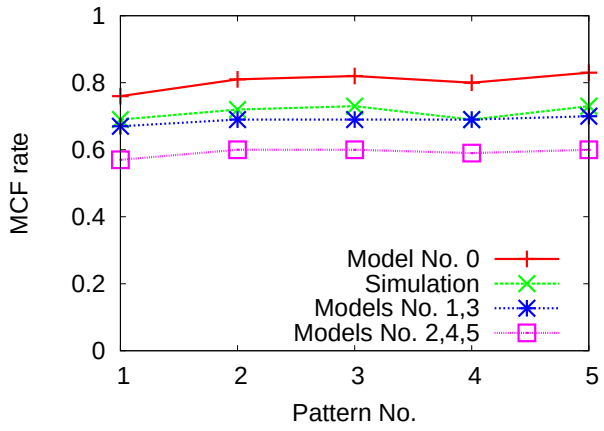


Fig. 9. The modeling and simulation results for individual random permutation patterns on $dfly(2,4,2,9)$

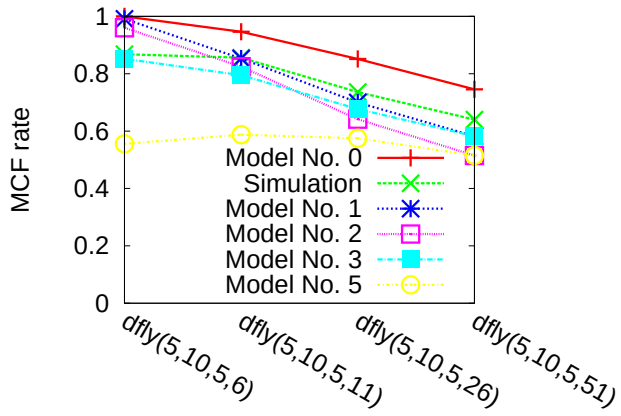


Fig. 10. The modeling and simulation results for random permutation patterns on different number of $group(5,10,5)$ groups

paths as shown in Table 3. Results for Model No. 4, which are in-between the results for Models No. 3 and and No. 5, are omitted to make the figure less dense. From the figure, it is evident that individual control overestimates the throughput when the number of paths in a group (MIN or VLB) is sufficiently large, while the all random control underestimates the throughput. The overall throughput estimation is a combination of the estimation of VLB paths and MIN paths. Thus, Model No. 0 overestimates the throughput for both VLB and MIN paths, resulting in consistent over-estimation of throughput for all cases. Similarly, Model No. 5 consistently underestimates the throughput for all cases. Model No. 3 consistently tracks the throughput obtained from simulation for different topologies. Notice that the overall throughput estimation is the combination of the estimation for MIN and VLB paths: over-estimating or under-estimating either MIN or VLB performance can sometimes dominate the overall prediction, resulting in prediction errors. For example, for $dfly(5,10,5,6)$ with 10 MIN paths per flow, Models No. 1 and No. 2 both overestimate the throughput for MIN by assuming individual control, resulting large overall prediction errors.

Figure 11 shows the average modeling and simulation results for five random shift patterns on the largest Dragonfly of different sizes $dfly(p,a,h,a\times h+1)$. This is one of the adversarial traffic patterns for Dragonfly. From the rate allocation perspective, however, it is clear what needs to happen to achieve high performance: use the VLB paths uniformly. As can be seen from the figure, even with the full control of the rate allocation for the patterns, the throughput is not much higher than treating all VLB paths the same. For this pattern, Model No. 0 only slightly overestimates the throughput while Models No. 2, 4, 5 only slightly underestimates the throughput. Models No. 1 and No. 3, nonetheless, produces the most accurate prediction. Figure 12 compares modeling and simulation results on Dragonfly topologies with the same group $group(5,10,5)$, but different number of groups. Very similar results to those in Figure 11 are observed.

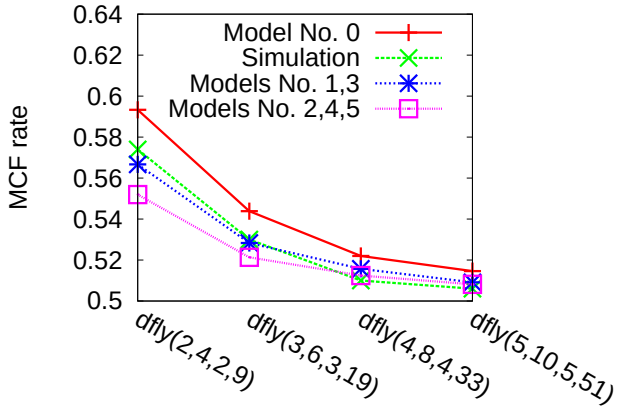


Fig. 11. The modeling and simulation results for random shift patterns on $dfly(p,a,h,a\times h+1)$

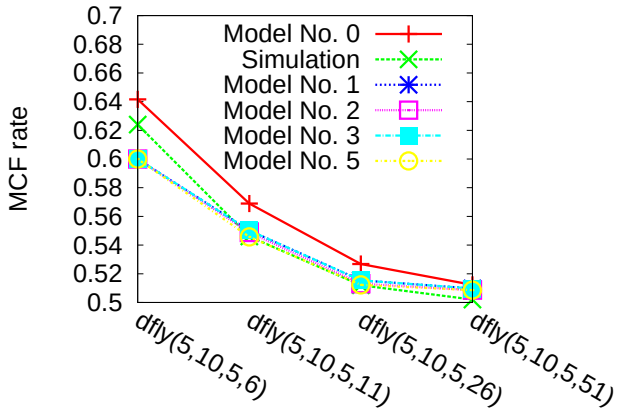
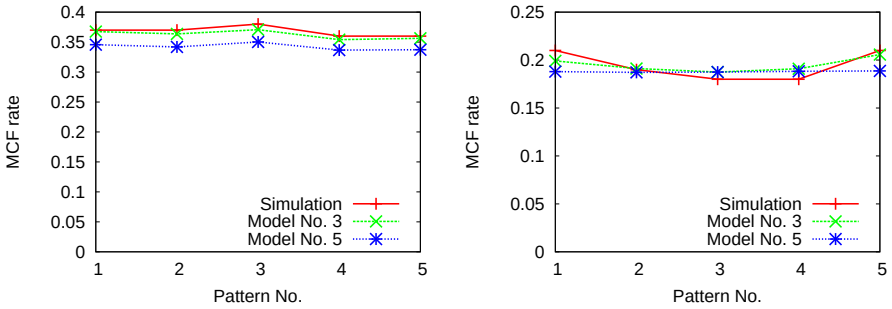


Fig. 12. The modeling and simulation results for random shift patterns on different number of $group(5,10,5)$ groups



(a) random permutation patterns (b) random shift patterns

Fig. 13. The modeling and simulation results on the 6-group Cascade topology

Figure 13(a) shows modeling and simulation results for five different random permutation patterns on the 6-group Cascade topology. We recall that the LP formulation given by Model No. 0 requires a unique variable for each unique path. Due to the large number of VLB and MIN paths in this topology, calculating the performance upper bound of UGAL-G in Cascade would then require solving LP with several billions of variables which is not computationally feasible. We, therefore, omit considering Model No. 0 on the Cascade system and compare UGAL-G performance with the remaining five models. In the experiments, Models No. 1 and No. 3 result in almost the same values while Models No. 2, No. 4, and No. 5 yield almost the same value. We only show the results for Models No. 3 and No. 5 in the figure for clarity. For this topology, the number of MIN and VLB paths are both very large. Models No. 1 and No. 3 differ in how the MIN paths are controlled: Model No. 1 assumes individual control of MIN paths while Model No. 3 assumes path-length based control. The fact that Models No. 1 and No. 3 yield similar results for the random permutation patterns indicates that fine-grain control of the MIN paths does not yield better throughput performance for this topology, which is likely due to the large number of links between each pair of groups. Models No. 2, No. 4 and No. 5 also only differ in how the MIN paths are controlled. Thus, similar logic applies. It is evident from Figure 13(a) that Model No. 3 and Model No. 1 predict the throughput performance on this topology very accurately. The prediction errors for the five random permutation patterns range from 0.0% to 2.6%. In fact, even Model No. 5 (as well as Models No. 2 and No. 4) has good prediction accuracy with errors up-to 7.0%. These results confirm that when the number of MIN and VLB paths are large, the control of UGAL-G over the MIN and VLB paths is group-based. Figure 13(b) shows modeling and simulation results for five random shift patterns on the same Cascade topology. The trend is very similar: UGAL-G performance is almost perfectly approximated by Model No. 3 and can be reasonably approximated with Model No. 5.

Other patterns and other Dragonfly topologies have also been studied. The results have the similar trend: individual control consistently overestimates the performance although the level of over-estimation differs based on the topology; all random control

consistently underestimates the performance; and the path-length-based random control, which takes the three distinguished features of UGAL-G described in Section 3 into consideration, consistently tracks the performance across a wide range of topologies. These results have two indications. First, UGAL-G has group-based control when the number of MIN and VLB paths is sufficiently large. Second, path-length-based control for both MIN and VLB paths (Model No. 3) is sufficient to model UGAL-G accurately on different Dragonfly topologies. As a result, the LP formulation only needs a small number of variables (at most 6 for $dfly(p,a,h,g)$ and 12 for Cray Cascade) to model each flow; and the models can be used to obtain throughput performance for large systems with tens of thousands of flows.

5 Related Work

Since the Dragonfly network was first introduced, it has been clear that a globally adaptive routing scheme is needed. In the seminal work by Kim et al [1], the authors propose selecting a random intermediate group to route non-minimally in order to load-balance adversarial traffic patterns over global channels. Jiang proposes several adaptive routing heuristics that approximate UGAL-G [6]. Improvements over the original UGAL-based scheme have been developed. Garcia et al. [15] are the first to address local congestion inside Dragonfly groups and proposed allowing non-minimal routing on both intra- and inter-group communication in their OFAR routing scheme. OFAR-CM [17] proposes throttling packet injection at local nodes as well as routing through an escape subnetwork to mitigate congestion on OFAR routing at the cost of additional hops. Opportunistic Local Misrouting (OLM) [18] allows non-minimal routing on both local and global levels of the Dragonfly hierarchy and the routing decision may be updated at any hop. Improvements for load estimation with UGAL-based routing scheme have also been developed [19, 20]. Existing research on UGAL-based routing mainly focuses on improving the effectiveness of the routing scheme. Jain et al. [21] provide an iterative model to predict the link utilization and thus, estimate throughput of UGAL-G routing on large-scale Dragonfly networks. Their model uses a bandwidth approximation scheme assuming all flows have a fair of bandwidth on each link, which is known to under-estimate throughput with a multi-path routing. Our work is different from the existing research in that we develop efficient throughput performance models using linear programming that give more insights about rate allocation control of UGAL on Dragonfly designs.

6 Conclusion

We develop a set of throughput models for UGAL-G on the Dragonfly topology based on the level of control that UGAL-G has on the MIN and VLB paths, and identify a robust model that is both accurate and efficient for a large number of Dragonfly variations. The model not only provides a mechanism to predict the aggregate throughput performance for large scale Dragonfly networks, but also reveals (1) that even with the

precise global information, UGAL-G is unable to achieve a fine-grain control over individual paths that are available, and (2) that UGAL-G in general allocates rates to groups of paths.

The Dragonfly topology has a large number of variants. The level of control that UGAL has over its paths is largely determined by the number of MIN and VLB paths, which in turn is decided by the topology. This work in general indicates that higher level of control can be achieved by UGAL-G when the number of MIN (VLB) paths is small, and that the level of control decreases as the number of MIN (VLB) paths increases. More research is necessary to determine the relationship between the number of available MIN and VLB paths and the level of control that UGAL has over the paths.

References

1. John Kim, Wiliam J Dally, Steve Scott, and Dennis Abts. Technology-driven, highly-scalable Dragonfly topology. In *ACM SIGARCH Computer Architecture News*, volume 36, pages 77–88. IEEE Computer Society, 2008.
2. Greg Faanes, Abdulla Bataineh, Duncan Roweth, Edwin Froese, Bob Alverson, Tim Johnson, Joe Kopnick, Mike Higgins, James Reinhard, et al. Cray Cascade: A scalable HPC system based on a Dragonfly network. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, page 103. IEEE Computer Society Press, 2012.
3. NERSC Cori supercomputer. http://www.nersc.gov/users/computational-systems/cori/.
4. Billy J. Archer and Manuel Vigil. The Trinity system. In *Nuclear Explosive Code Development Conference (NECDC)*, Los Alamos, New Mexico, October 20–24, 2014. Also appears as Los Alamos Technical Report LA-UR-15-20221.
5. Arjun Singh. *Load-Balanced Routing In Interconnection Networks*. PhD thesis, Stanford University, 2005.
6. Nan Jiang, John Kim, and William J. Dally. Indirect adaptive routing on large scale interconnection networks. *SIGARCH Comput. Archit. News*, 37(3):220–231, June 2009.
7. Open Networking Foundation. Sdn architecture. White Paper, ONF TR-502, June 2014. available at https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf.
8. Farhad Shahrokhi and D. W. Matula. The maximum concurrent flow problem. *J. ACM*, 37(2):318–334, April 1990.
9. S. A. Jyothi, A. Singla, P. B. Godfrey, and A. Kolla. Measuring and Understanding Throughput of Network Topologies. Nov. 2016. Accepted at The International Conference for High Performance Computing, Networking, Storage and Analysis (SC'16).
10. Ankit Singla, P. Brighten Godfrey, and Alexandra Kolla. High throughput data center topology design. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, April 2014.
11. P. Faizian, M. A. Mollah, X. Yuan, S. Pakin, and M. Lang. Random regular graph and generalized De Bruijn graph with *k*-shortest path routing. In *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 103–112, May 2016.
12. N. Jiang, J. Balfour, D. U. Becker, B. Towles, W. J. Dally, G. Michelogiannakis, and J. Kim. A detailed and flexible cycle-accurate network-on-chip simulator. In *Performance Analysis of Systems and Software (ISPASS), 2013 IEEE International Symposium on*, pages 86–96, April 2013.
13. NERSC Edison supercomputer. http://www.nersc.gov/users/computational-systems/edison/.

14. L. G. Valiant. A scheme for fast parallel communication. *SIAM Journal on Computing*, 11(2):350–361, 1982.
15. M. Garcia, E. Vallejo, R. Beivide, M. Odriozola, C. Camarero, M. Valero, G. Rodríguez, J. Labarta, and C. Minkenberg. On-the-fly adaptive routing in high-radix hierarchical networks. In *Parallel Processing (ICPP), 2012 41st International Conference on*, pages 279–288, Sept 2012.
16. IBM CPLEX optimizer. https://www.ibm.com/us-en/marketplace/ibm-ilog-cplex/.
17. M. Garcia, E. Vallejo, R. Beivide, M. Valero, and G. Rodríguez. OFAR-CM: Efficient Dragonfly networks with simple congestion management. In *High-Performance Interconnects (HOTI), 2013 IEEE 21st Annual Symposium on*, pages 55–62, Aug 2013.
18. M. Garcia, E. Vallejo, R. Beivide, M. Odriozola, and M. Valero. Efficient routing mechanisms for Dragonfly networks. In *Parallel Processing (ICPP), 2013 42nd International Conference on*, pages 582–592, Oct 2013.
19. J. Won, G. Kim, J. Kim, T. Jiang, M. Parker, and S. Scott. Overcoming far-end congestion in large-scale networks. In *High Performance Computer Architecture (HPCA), 2015 IEEE 21st International Symposium on*, pages 415–427, Feb 2015.
20. P. Fuentes, E. Vallejo, M. Garcia, R. Beivide, G. Rodríguez, C. Minkenberg, and M. Valero. Contention-based nonminimal adaptive routing in high-radix networks. In *Parallel and Distributed Processing Symposium (IPDPS), 2015 IEEE International*, pages 103–112, May 2015.
21. N. Jain, A. Bhatele, X. Ni, N. J. Wright, and L. V. Kale. Maximizing throughput on a dragonfly network. In *SC14: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 336–347, Nov 2014.