# Topological Characterization of Hamming and Dragonfly Networks and its Implications on Routing

Cristóbal Camarero, Enrique Vallejo and Ramón Beivide,, University of Cantabria

Current HPC and datacenter networks rely on large-radix routers. Hamming graphs (Cartesian products of complete graphs) and dragonflies (two-level direct networks with nodes organized in groups) are some direct topologies proposed for such networks. The original definition of the dragonfly topology is very loose, with several degrees of freedom such as the inter- and intra-group topology, the specific global connectivity and the number of parallel links between groups (or trunking level).

This work provides a comprehensive analysis of the topological properties of the dragonfly network, providing balancing conditions for network dimensioning, as well as introducing and classifying several alternatives for the global connectivity and trunking level. From a topological study of the network, it is noted that a Hamming graph can be seen as a canonical dragonfly topology with a large level of trunking. Based on this observation and by carefully selecting the global connectivity, the Dimension Order Routing (DOR) mechanism safely used in Hamming graphs is adapted to dragonfly networks with trunking. The resulting routing algorithms approximate the performance of minimal, non-minimal and adaptive routings typically used in dragonflies, but without requiring virtual channels to avoid packet deadlock, thus allowing for lower-cost router implementations. This is obtained by selecting properly the link to route between groups, based on a graph coloring of the network routers. Evaluations show that the proposed mechanisms are competitive to traditional solutions when using the same number of virtual channels, and enable for simpler implementations with lower cost. Finally, multilevel dragonflies are discussed, considering how the proposed mechanisms could be adapted to them.

Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Network topology*; C.1.2 [**PROCESSOR ARCHITECTURES**]: Multiple Data Stream Architectures—*Interconnection architectures*; B.4.3 [**INPUT/OUTPUT AND DATA COMMUNICATIONS**]: Interconnections—*Topology*

General Terms: Interconnection networks

Additional Key Words and Phrases: Hamming graph, dragonfly network, topology, deadlock-freedom, routing

**ACM Reference Format:**
Cristóbal Camarero, Enrique Vallejo, and Ramón Beivide, 2014. Topological characterization of Hamming and dragonfly networks and its implications on routing. *ACM Trans. Architec. Code Optim.* 11, 4, Article 39 (December 2014), 25 pages.
DOI: http://dx.doi.org/10.1145/10.1145/2677038


This work has been supported by Spanish FPU grant AP2010-4900, the Spanish Science and Technology Commission (CICYT) under contracts TIN2010-21291-C02-02 and TIN2013-46957-C2-2-P, the European Union FP7 under Agreements ICT-288777 (Mont-Blanc) and ERC-321253 (RoMoL), the European HiPEAC Network of Excellence and the JSA no. 2013-119 as part of the IBM/BSC Technology Center for Supercomputing agreement.

DOI: http://dx.doi.org/10.1145/10.1145/2677038

## 1. INTRODUCTION

Technology trends suggest that the use of high-radix routers [Kim et al. 2005] is the most cost-efficient alternative for the interconnection networks typically used in datacenters and High-Performance Computers (HPC). An *interconnection network* is defined by its topology, routing, flow control and deadlock avoidance mechanisms, along with other technological aspects such as the used media and router design. However, very often the *topology* and *network* terms are used interchangeably in the literature. The *topology* of a network defines how the different routers are connected. An *indirect* topology (or network) employs transit routers, to which no computing node is connected. Typical examples of these are the tree and folded Clos topologies. Conversely, a *direct* topology does not employ transit routers, so each network router has one or more computing nodes directly connected to it. When all the network links are point-to-point, as often occurs today in HPC and datacenter networks, the topology can be completely defined using a graph. The graph degree, $\Delta$, is determined by the radix of the network routers, not considering the connections to the computing nodes. Frequent direct topologies proposed for HPC and datacenters are those based on meshes, tori, dragonflies [Kim et al. 2008] and Hamming graphs (also known as flattened butterflies [Kim et al. 2007]). Among these, dragonflies and Hamming graphs are suitable for their use with high-radix routers, and they will be studied in detail in this paper. Some important issues of the network topology are its scalability for a given diameter and degree (in the graph theory literature known as the degree-diameter problem), its edge- and vertex-transitivity properties, which guarantee network symmetries and balanced resource usage, as well as the simplicity of the deadlock avoidance mechanisms. All these aspects are discussed next.

The degree-diameter (or $d-k$) problem consists in finding a graph $G$ for a given degree $\Delta$ and diameter $k$ with the maximum number of nodes $N(\Delta,k)$. An upper bound in $N(\Delta,k)$ is the Moore bound, of value $M(\Delta,k)=\frac{\Delta(\Delta-1)^k-2}{\Delta-2}$, [Hoffman and Singleton 1960]. Graphs reaching this bound are called Moore graphs. Optimizing the degree-diameter problem provides the largest possible network with optimal performance under uniform traffic. However, practical constraints such as regularity of the topology and fine-grain scalability[1], convenient layouts and cable length, number of computing nodes per router (or *concentration level*), routing mechanisms and performance under alternative traffic patterns make that other topologies with a lower amount of network nodes become more attractive.

Network symmetries imply graph automorphisms. An automorphism of a graph $G=(V,E)$ is a bijection $f: V \mapsto V$ such that for any edge $\{x,y\} \in E$, there is an edge $\{f(x),f(y)\} \in E$. Then $G$ is said *vertex-transitive* if for any pair of vertices $x_1,x_2 \in V$ there is an automorphism $f$ such that $f(x_1)=x_2$; and $G$ is said *edge-transitive* if for any edges $\{x_1,y_1\},\{x_2,y_2\} \in E$ there is an automorphism such that $\{f(x_1),f(y_1)\}=\{x_2,y_2\}$. These graph symmetries are interesting properties since they guarantee equalized resource utilization and allow for systematic analysis of diverse network characteristics.

Routing and deadlock avoidance mechanisms play a significant role in the router design and complexity. Distance-based deadlock avoidance mechanisms are frequently employed in low-diameter networks such as dragonflies. These mechanisms, based on a original design by Günther [Günther 1981], employ as many virtual channels per router input port as the longest path allowed in the network. Thus, allowing longer paths for nonminimal routing increases the router area and complexity. In general,

[1]Being able to construct topologies for many sizes. For example, the binary hypercube requires $2^n$ routers, and hence, it is not fine-grain scalable.

deadlock-free routing mechanisms which do not impose dependencies on the number and use of virtual channels are desirable.

This paper characterizes and compares Hamming and dragonfly topologies, studying their scalability, their respective degrees of freedom and providing a systematic characterization of each graph including balancing conditions that lead to a uniform use of network resources under uniform traffic. The relationship between the Hamming graph and the dragonfly topology is studied, showing that the former can be seen as a dragonfly topology with an extremely high level of *trunking*. Based on this relationship, the dimension-ordered deadlock-free routing (DOR) mechanism used in Hamming graphs, which does not rely on virtual channels (VCs), is adapted to dragonflies. Minimal and non-minimal routing mechanisms of this type are introduced for dragonflies with trunking $t \geq 2$ and $t \geq 4$ respectively. These mechanisms rely on routing restrictions and therefore they decouple the number and use of virtual channels from deadlock avoidance. An evaluation shows that the proposed mechanisms are competitive with state-of-the-art alternatives, without imposing minimal VC requirements on the router design.

On the other hand, high-radix is the norm for current HPC discrete routers, forthcoming designs such as Intel's Knights Landing and future Xeon chips will implement on-chip routers [Hazra 2014]. In such designs, the router competes with on-chip cores, memories and I/O for the chip resources, including the pin bandwidth. This will necessarily lead to lower-radix routers. Scaling to large networks based on low-radix switches can be accomplished using multi-level dragonflies. Such designs will be studied in the last part of the paper, compared to previously proposed routing mechanisms.

The rest of the paper is organized as follows. Section 2 presents related work in the area. Sections 3 and 4 introduce and characterize the Hamming and dragonfly topologies. Section 5 focuses on dragonflies with trunking in the global level. Section 6 introduces two novel deadlock-free routing mechanisms for dragonflies with trunking, based on coloring the underlying graphs, which are evaluated in Section 7. To finish the contributions, Section 8 makes some remarks about the scalability and routing of multi-level dragonfly networks, discussing how to adapt the previous proposals for such cases. Finally, Section 9 concludes the paper.

## 2. RELATED WORK

The Moore bound sets a limit on the degree-diameter problem. A thorough survey of the problem and Moore graphs can be found in [Miller and Sirán 2013]. For diameter $k = 1$ the complete graphs $K_{\Delta+1}$ equal the bound. Their simplicity and systematic existence make them very interesting; however they are subject to technological constraints given the large degree necessary to reach a high number of nodes. For diameter $k = 2$ there are only 2 or 3 Moore graphs [Hoffman and Singleton 1960]: the Petersen graph ($\Delta = 3$, $N = 10$), the Hoffman-Singleton graph ($\Delta = 7$, $N = 50$) and an hypothetical graph with $\Delta = 57$ and $N = 3250$ whose existence is still an open problem. This sporadic existence of Moore graphs complicates scalability, being very difficult to decide which topology to use for a given network size. The problem can be relaxed by considering only the asymptotic behaviour. This relaxed problem consists in finding for every diameter $k$ an infinite family of graphs with about $\Delta^k$ vertices. Such graphs exist for $k = 1$ (complete graphs), $k = 2$ [Brown 1966; Brahme et al. 2013], $k = 3$ and $k = 5$ [Delorme 1985]. They are conjectured to exist for any diameter, but even the best general bounds are exponential in $k$. The work in [Brahme et al. 2013] seems to be the first to propose one of these families as interconnection networks, but fails to address many practical problems.

The Hamming graph [Mulder 1982] has been studied extensively. Other names for this graph, or for topologies based on it, are *rook's graph*, *K-cube* [LaForge et al. 2003],

*generalized hypercube* [Bhuyan and Agrawal 1984], *flattened butterfly* [Kim et al. 2007] and *HyperX topology* [Ahn et al. 2009]. This graph has been also considered in [Ahn et al. 2013] as one of the base topologies for an intra-switch network[2]. The dragonfly network was first introduced in [Kim et al. 2008]. Different routing mechanisms for dragonflies that better adapt to the traffic pattern or reduce the implementation cost have been proposed in other works, [Jiang et al. 2009; García et al. 2012; García et al. 2013a]. Industrial implementations have been the IBM PERCS [Arimilli et al. 2010] and Cray Cascade [Faanes et al. 2012].

Network dimensioning typically seeks to balance the utilization of the network resources to maximize performance. Resource usage being balanced or not depends on the topology, traffic pattern and routing employed. Under uniform traffic and minimal routing, Square Hamming graphs and Dragonflies with twice as many local ports as global ports per router are balanced [Kim et al. 2008], as will be detailed later. By contrast, an unbalanced design such as a rectangular Hamming graph would provide reduced performance caused by the bottleneck in the scarcest resources. However, even a balanced network can easily saturate under adverse traffic using minimal routing. This occurs when all the traffic concentrates on some few links, which leads to severe congestion. Valiant routing [Valiant 1982] selects a random intermediate router; traffic is first sent minimally to the intermediate router and then minimally to the final destination. This randomizes the network load, balancing the use of links, but doubles the utilization of the resources, halving its maximum throughput. Alternatively, task placement randomization [Bhatele et al. 2011] avoids hotspots by randomizing communications. Given the disparity of performance depending on the traffic pattern and routing, Hamming and dragonfly networks typically require adaptive routing mechanisms which rely on minimal routing for uniform traffic and revert to Valiant routing for adverse traffic patterns. Several of such adaptive routing mechanisms have been proposed in the literature, [Kim et al. 2007; Kim et al. 2008; Jiang et al. 2009; García et al. 2012; García et al. 2013a].

Networks built on Hamming graphs are deadlock-free under DOR. Valiant routing can be made deadlock-free when DOR is employed for each half of the path using different VCs, requiring two of them. For dragonflies, most of the previous proposals adapt the distance-based mechanism by Günther [Günther 1981], employing as many VCs per router port as the longest path allowed in the network. When local and global links are always traversed in the same sequence, their VCs can be considered independently, leading to 2/1 VCs (local/global) required for minimal routing and 4/2 for Valiant routing [Valiant 1982; Prisacari et al. 2014]. In [Kim et al. 2008] the authors reduce this number to 3/2 by misrouting traffic to an intermediate group instead of an specific intermediate router, but in this way the traffic is not completely uniform and pathological performance problems can arise [García et al. 2012]. In OFAR [García et al. 2012] a simple deadlock-free escape network is embedded in the dragonfly and packets have the option to move to the escape network to avoid deadlock. Hence, in each port only 1 or 2 VCs are necessary (depending if it belongs to the escape subnetwork). However, this mechanism does not guarantee bounded paths per se, and requires a congestion management mechanism to avoid saturation in the escape subnetwork, [García et al. 2013]. Restricted Local Misrouting (RLM, [García et al. 2013a]) allows for local misrouting within any group of a canonical dragonfly without increasing the number of required VCs. This is implemented by forbidding certain combinations of two local hops which would generate cycles, in a similar way to how our routing mechanisms for dragonflies with trunking introduced in Section 6 select the global links that guaran-

[2][Ahn et al. 2013] also considers *local* and *global* topologies as we do in this work, but they refer to the intra-switch topology and the traditional topology between switches, instead of per-group and intra-group.

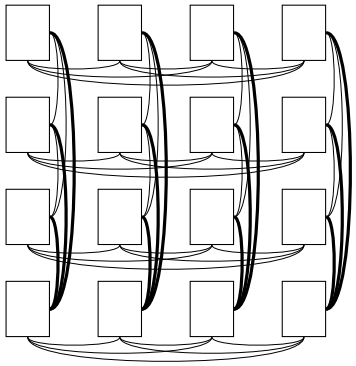


Fig. 1: Hamming graph $K_4 \square K_4$ with vertices arranged in rows and columns.

tee deadlock avoidance. Opportunistic Local Misrouting (OLM, [García et al. 2013a]) allows for cyclic dependencies to appear when applying local misrouting, but it guarantees that an alternative safe escape path always exists at any hop in the network.

The use of multiple virtual channels besides providing deadlock-freedom helps to reduce Head-of-Line Blocking (HoLB). However, they entail a significant cost. Not only they increase the area and power requirements for the router, but also make some router allocator stages more complex, leading to lower router frequencies and reduced throughput, [Peh and Dally 2001]. For this reason, multiple works propose alternatives to avoid or reduce the number of VCs in network routers, such as [Wang et al. 2013; García et al. 2012]. HoLB is typically mitigated in these cases employing internal speedup, such as in [Arimilli et al. 2010; Faanes et al. 2012].

## 3. HAMMING GRAPHS

This section defines Hamming graphs, their properties, some alternative isomorphic definitions and the main routing mechanisms proposed for networks based on it.

The *Hamming distance* between two vectors is the number of components in which the vectors differ. Given a space $S$ over which the Hamming distance is defined, the *Hamming graph* is defined as the graph with $S$ as vertex set in which two vertices are connected if and only if their Hamming distance is 1. For the Hamming distance the only relevant characteristics of the space are the number of components (dimensions) and the possible values of each component, this is, it can be assumed that the space is $\mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_n}$[3] for some integers $m_i$. Figure 1 shows a representation of the Hamming graph over $\mathbb{Z}_4 \times \mathbb{Z}_4$.

The Hamming graph is isomorphic to the Cartesian product of complete graphs $K_{m_1} \square \cdots \square K_{m_n}$. This is, the Cartesian product is defined as having two vertices connected if and only if, for some component, they are connected in the corresponding factor and the other components are equal. Formally, for a pair of graphs $G_1$, $G_2$, their Cartesian product $G_1 \square G_2$ is the graph with vertices $V(G_1 \square G_2) = V(G_1) \times V(G_2)$ where the vertices $(x_1, y_1)$ and $(x_2, y_2)$ are connected if and only if $x_1 = x_2 \wedge \{y_1, y_2\} \in E(G_2)$ or $y_1 = y_2 \wedge \{x_1, x_2\} \in E(G_1)$. As in the complete graph all vertices are connected, in the Hamming graph every vertex is connected with any other which differs in exactly one component. The Hamming graph is also isomorphic to the Cayley graph over the

[3]Here $\mathbb{Z}_m$ denotes the set of integers modulo $m$, which in other texts is written $\frac{\mathbb{Z}}{m\mathbb{Z}}$.

Abelian group $(\mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_n}, +)$, with generator set $\bigcup_{i=1}^{n} \{xe_i \mid x \in \mathbb{Z}_{m_i} \setminus \{0\}\}$, where $e_i$ is the vector with 1 as the $i$-th component and 0 otherwise.

This paper focuses in the bidimensional case, i.e. the Hamming graph over the space $\mathbb{Z}_a \times \mathbb{Z}_b$, for any pair of integers $a, b$. This Hamming graph is a diameter $k = 2$, $\Delta$-regular graph, for $\Delta = a + b - 2$, comprising $ab$ vertices. In the square case, this corresponds to $\frac{1}{4}\Delta^2 + \Delta + 1$ vertices. For Cayley graphs over Abelian groups of diameter 2 there is an upper bound of $\frac{1}{2}\Delta^2 + \Delta + 1$ vertices; the current best construction inside this family is the given in [Macbeth et al. 2012], which achieves $\frac{3}{8}(\Delta^2 - 4)$ vertices, about $\frac{3}{4}$ of the bound. Square Hamming graphs have about $1/2$ of this bound, so while they are not the best, they have a good position among Cayley graphs over Abelian groups, while existing for any even degree. Each of these vertices represents one router in the network, to which $\Delta_0$ compute nodes are attached (also known as *concentration level*). Thus, each router requires $R = \Delta + \Delta_0 = \Delta_0 + a + b - 2$ ports.

In a network with a Hamming topology up to $\Delta_0 = \min(a, b)$ compute nodes per router can be connected without bisection bandwidth limitations under uniform traffic. A larger concentration value can introduce network bottlenecks due to oversubscription. For a proof, assume without loss of generality $a < b$ and consider the traffic from the region $\{(x, y) \mid 0 \leq x < \frac{a}{2}, 0 \leq y < b\}$ into $\{(x, y) \mid \frac{a}{2} \leq x < a, 0 \leq y < b\}$, with $a$ even for simplicity. Each region contains $\frac{ab}{2}$ routers, each router attached to $\Delta_0$ compute nodes. As the regions have the same size, the probability of having a destination in the other one is $\frac{1}{2}$. Thus $\frac{ab}{4}\Delta_0$ packets must traverse the links joining the regions each cycle. The number of these links is $b \cdot \frac{a}{2} \cdot \frac{a}{2}$; thus, to avoid saturation $\frac{ab}{4}\Delta_0 \leq \frac{ba^2}{4}$ is required, which simplifies to $\Delta_0 \leq a$. Then, in a balanced Hamming network with $a = b = \Delta_0$, there are $a^3$ compute nodes for routers of radix $R = 3a - 2$. Then, for a given radix $R$ the network connects up to $\left(\frac{R+2}{3}\right)^3$ compute nodes.

Like all Cayley graphs, the Hamming graph is vertex-transitive [Akers and Krishnamurthy 1989]. This can be seen with the automorphism $f(v) = v + v_2 - v_1$ for some vertices $v_1, v_2$, for which $f(v_1) = v_2$. The edges from $(x, y)$ to $(x', y)$ can be naturally denoted as $a$-edges and the edges from $(x, y)$ to $(x, y')$ as $b$-edges, corresponding to the two different dimensions in the Hamming graph. Under uniform traffic, a minimal network path will have one $a$-link with a probability $\frac{(a-1)b}{ab-1}$ and one $b$-link with probability $\frac{(b-1)a}{ab-1}$, which are both almost 1. Thus, in order to balance the use of the network links, the required condition is to have the same number of links per dimension $(a = b)$, which corresponds to a square Hamming graph. Indeed, the Hamming graph is edge-transitive if and only if it is square. The sufficient condition is simple, if $a = b$ there exists an automorphism which maps each vertex $(x, y)$ into $(y, x)$. For the necessary condition, assume without loss of generality that $a < b$; then every $a$-link is included in some $K_a$ subgraph but not in any $K_b$ subgraph, thus $a$-links cannot be mapped into $b$-links. An unbalanced (not edge-transitive) implementation has less links in the shorter dimension, which becomes a performance bottleneck because of their higher utilization.

Networks based on the Hamming graph are deadlock-free under a DOR policy. This imposes restrictions on the paths that packets can follow, but not on the number of VCs employed by routers. Alternatively, distance-based deadlock avoidance mechanisms could be used without routing restrictions if the routers employ at least two VCs: one for the first hop and the other for the second. Finally, it is interesting to note that perfect error-correcting Hamming codes based in this graph directly translate into solutions for the resource placement problem in Hamming networks (in a analogous way to [Bae and Bose 1996]).

## 4. DRAGONFLY TOPOLOGIES

This section presents the dragonfly topology analyzing its multiple degrees of freedom. Next, it discusses how some dragonfly topologies are subgraphs of a bidimensional Hamming graph. Finally, it introduces a formal definition of the canonical dragonfly topology with several alternatives for its global link arrangement.

The *dragonfly network* was proposed in [Kim et al. 2008] as a two-level hierarchical direct network. A dragonfly topology has $b$ groups $(0, \ldots, b-1)$ each group being composed of $a$ routers $(0, \ldots, a-1)$. Routers within a group (first level) are connected by short, cheap, electrical *local* links. Different groups (second level) are connected by long, expensive, optical *global* links. The definition of the dragonfly in [Kim et al. 2008] is, purposely, very loose, focusing on technological and economical aspects, rather than providing a closed definition of the underlying graph. Thus, from a formal point of view, multiple different topologies can be considered as variants of the dragonfly.

Apart from the parameters $a$ and $b$, there are three degrees of freedom in the definition of a dragonfly topology:

(1) *local topology*: the connectivity pattern of the routers within a group,
(2) *global topology*: the connectivity pattern between the different groups, and
(3) *global link arrangement*: the specific router on each group to which each global link connects.

The diameter $k$ of the dragonfly topology depends on the diameters of the global topology $k_g$ and local topology $k_l$ as follows: $k \leq k_g + (k_g + 1)k_l = k_g + k_g k_l + k_l$. That is, a limit of $k_g$ global links, $k_g + 1$ visited groups, with at most $k_l$ local links in each of the visited groups. In order to minimize the diameter, the complete graph can be employed as both local and global topologies, leading to $k = 3$. Furthermore, the complete graphs reach the Moore bound and thus are very good candidates considering scalability. This choice of topologies has been the one of previous proposals [Kim et al. 2008; Arimilli et al. 2010; Jiang et al. 2009; García et al. 2012; García et al. 2013a] and hence we will call *canonical dragonfly* to the dragonfly network using complete graphs $K_a$ and $K_b$ in both local and global topologies. The canonical dragonfly, for $k = 3$, asymptotically reaches $4/27$ of the vertices of the Moore bound. This value is only exceeded by graphs designed to reach the bound; mainly the family introduced in [Delorme 1985], which has severe practical inconveniences, such as restricting $\Delta - 1$ to odd powers of 2.

Alternative implementations to the canonical dragonfly also exist, such as in Cray Cascade [Faanes et al. 2012], where a complete graph is used for the global topology and a rectangular 2D Hamming graph is used for the local one. In Section 8 we will discuss how this topology can be considered as a 3-level dragonfly. Topologies can employ parallel links between routers (*trunking*) what will be considered later in Section 5 and, unless otherwise noted, it is not employed in the dragonfly.

The degree of the topology $\Delta$ can be divided into the two levels. The degree associated to the first level is denoted by $\Delta_1$, this is, the number of local links connected to each router. Analogously, $\Delta_2$ represents the number of global links connected to each router. Thus, the topology has degree $\Delta = \Delta_1 + \Delta_2$ and the routers have a total number of ports or radix $R = \Delta_0 + \Delta_1 + \Delta_2$, treating computing nodes as a level 0. In any canonical dragonfly $b = a\Delta_2 + 1$ and $a = \Delta_1 + 1$. To achieve a balanced use of resources under uniform traffic, this is, to have similar load in local and global links, the condition $2\Delta_2 \approx \Delta_1$ needs to hold; the balancing condition proposed in [Kim et al. 2008] is $a = 2\Delta_2$, whereas up to $\Delta_0 = \Delta_2$ compute nodes can be connected to each router without saturating the network under uniform traffic. In a canonical dragonfly this imposes a relation between the group size $a$ and the number of groups $b$. Given a group size $a$, the network is balanced only for the corresponding number of groups $b$; with less groups

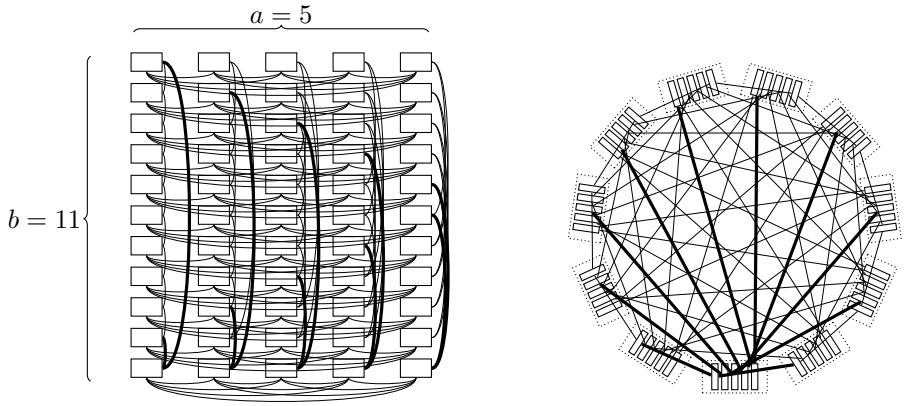



Fig. 2: Two layouts of the same dragonfly topology which is a subgraph of $K_5 \square K_{11}$, with $\Delta_2 = 2$, with nodes organized in rows and columns (left, each row corresponds to a different group) or groups (right). Global links leaving group 0 are in bold.

there would be too few global links which would become a bottleneck, and with more groups the local links would be the bottleneck. The second case should be typically forbidden by design by setting a maximum system size, but the first one is common in not fully populated systems which can be upgraded by installing more groups. In such case, balanced networks with a low number of groups $b$ can be built using trunking; the corresponding balancing conditions will be discussed in detail in Section 5.1. In a balanced canonical dragonfly without trunking, the routers have radix $R \approx 2a$, there are about $a^3/2$ routers and about $a^4/4$ compute nodes. Then, for a given radix $R$ the network comprises up to $\approx R^4/2^6$ compute nodes.

Proposed routing mechanisms in the canonical dragonfly are hierarchical, routing first to the destination group and then to the destination node. The *minimal routing* introduced in [Kim et al. 2008] first locates the global link between the source and destination groups; the path consists of one local link $l$ to the router with the required global link, then the global link $g$ itself and finally a local link $l$ to the destination; this is denoted as a *lgl* route. Using such hierarchical routing (instead of a flat routing) avoids paths with only two global links *gg*, which can be shorter in terms of hops but typically have higher latency because of the longer physical length of global links. Most deadlock-free routing mechanisms for dragonflies rely on an ordered use of virtual channels. Each hop of a path employs a different VC in an increasing order, thus avoiding cyclic dependencies. Since minimal paths are always of type *lgl*, or a subset of them in the same order (but never *gll* or *llg*), using minimal routing, local ports employ two different VCs and global ports do not need to employ VCs.

In some cases, a canonical dragonfly topology is a subgraph of the rectangular Hamming graph $K_a \square K_b$. Figure 2 presents an example with $a = 5$, $b = 11$ and $\Delta_2 = 2$. The local topology of each dragonfly group corresponds to each of the complete graphs $K_a$, whereas the global topology links need to connect vertices in the same position of each group in order to belong to the original Hamming graph. Thus, $a$ independent graphs, $G_0, G_1, \ldots, G_{a-1}$, define the global link connectivity. In order to build a canonical dragonfly, each of these $G_i$ graphs needs to have $b$ vertices $V(G_i) = \{0, 1, \ldots, b-1\}$ and degree $\Delta_2$, which exists if and only if $b \geq \Delta_2 + 1$ and $2|b\Delta_2$. The global topology composed as the union of all $G_i$'s needs to be a complete graph $K_b$ so the result is a canonical dragonfly; the *union* of graphs over the same set of vertices is the graph containing all the edges of the factors, this is, $E(\bigcup_i G_i) = \{e \mid e \in E(G_i) \text{ for some } i\}$. Thus

the problem is to decompose $K_b$ into $a$ graphs, $G_0, \ldots, G_{a-1}$, of degree $\Delta_2$. Systematic decompositions can be found easily for $b$ odd and $\Delta_2$ even. For $\Delta_2 = 2$, as in Figure 2, $K_b$ can be decomposed into $\frac{b-1}{2}$ cycles for $b$ odd, [Hilton 1984]. For $\Delta_2 > 2$ even, several of such cycles can be merged into each of the $G_i$. Although only for certain parameters, as it will be further discussed in Subsection 4.1.3, these subgraphs of Hamming graphs are the only vertex-transitive canonical dragonfly arrangements which we have encountered.

### 4.1. Global Link Arrangement and Network Symmetries

Given a canonical dragonfly, there exist $b^{2^{O(a\Delta_2)}}$ possible arrangements for the global links. This subsection discusses link arrangements in general and a few specific cases: *consecutive*, *palmtree*, and *circulant-based* in which the topology is a subgraph of the Hamming graph, as introduced above. Finally, it presents a brief discussion on the selection of an arrangement. Arrangements with trunking will be presented in Section 5.2.

In general, any arrangement can be implemented as follows:

(1) For each group $g$, partition the set of groups other than $g$, into $a$ subsets (sets of groups) of cardinality $\Delta_2$. Then assign one subset to each router of $g$.
(2) For every pair of groups $A, B$, find in $A$ the router assigned to group $B$ and in $B$ the router assigned to group $A$. Then, add a global link between the routers found.

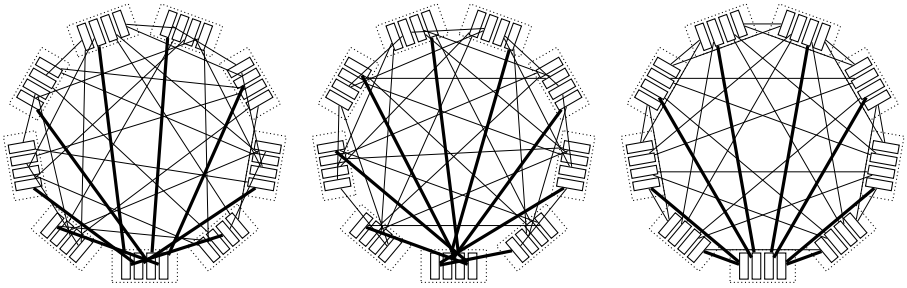
A *random* arrangement makes the choices in the first step at random. An example is presented in Figure 3a. Any network configuration admits being implemented in this way, although sometimes there is a simpler ad hoc implementation.

*4.1.1. Consecutive Arrangement.* The *consecutive* allocation of global links consists on connecting the routers in each group in consecutive order, with the groups in the network also in consecutive order, starting always from group 0 and skipping those links with source and destination being in the same group. Specifically, the vertex $i$ in group $j$ is connected for every integer $k = 0, \ldots, \Delta_2 - 1$ with the vertex $\lfloor \frac{j-1}{\Delta_2} \rfloor$ of the group $g = i\Delta_2 + k$ if $g < j$ and with the vertex $\lfloor \frac{j}{\Delta_2} \rfloor$ of the group $g + 1$ otherwise. Although not explicitly indicated, this consecutive arrangement can be inferred from the figures in [Kim et al. 2008]. Figure 3b shows an example for $a = 4$.

*4.1.2. Palmtree Arrangement.* The *palmtree*[4] arrangement presents the same global connectivity pattern in each group of the system. In this arrangement, vertex $i$ in group $j$ is connected to vertices $a - 1 - i$ in groups $j - i\Delta_2 - 1, j - i\Delta_2 - 2, \ldots, j - i\Delta_2 - \Delta_2 \pmod b$. Although not explicitly indicated, the palmtree arrangement can be inferred from the figures in [García et al. 2012]. A palmtree for $a = 4$ is included in Figure 3c.

The palmtree arrangement presents notable symmetries. The clearest one is the rotational symmetry given by the automorphism defined by sending the vertex $x$ in group $y$, $(x, y)$, to $(x, y + 1 \pmod b)$. This rotation shows that groups are equivalent. Another symmetry is given by $f(x, y) = (a - 1 - x, -y \pmod b)$, which is a reflection in each group. Therefore, there are at most $a/2$ classes of vertices modulo the equivalence relation induced by automorphisms. Interestingly, for any pair of vertices of the same class of these $a/2$ classes, there is a path between them using only global links. Reciprocally, each global link connects vertices of the same class.

[4]The name is inspired by the similarity of the links leaving each group with the Palm Islands in Dubai, which are shaped as a palmtree.



(a) Random (b) Consecutive (c) Palmtree

Fig. 3: Three arrangements for $a = 4, b = 9, \Delta_2 = 2$ with nodes organized in groups.

*4.1.3. Circulant-based Arrangement.* This arrangement is a particular case in which the dragonfly network is a subgraph of the Hamming graph, as introduced above in this section. We restrict the study to the case in which $\Delta_2$ is even and each of the subgraphs $G_i$ is a circulant graph. In this arrangement, vertex $i$ in group $j$ is connected to vertices $i$ in groups $j \pm (i\Delta_2/2+1), j \pm (i\Delta_2/2+2), \ldots, j \pm (i\Delta_2/2+\Delta_2/2) \pmod b$. In the example of Figure 2 with $a = 5, b = 11, \Delta_2 = 2$, each graph $G_i$ (corresponding to column $i$) contains the edges from $x$ to $x \pm i \pmod b$ and thus it is a circulant graph.

Interestingly, this arrangement has the property that for $\Delta_2 = 2$ if $b$ is a prime number, the resultant topology is vertex-transitive. To see that, consider the following automorphisms: an automorphism $f$ that maps the vertex $(i, j)$ into $(i, j + 1 \pmod b)$ and an automorphism $g$ which maps the vertex $(i, j)$ into $(\min\{2i + 1, b - 3 - 2i\}, 2j \pmod b)$. The automorphism $f$ cycles the groups, and hence, there are at most $a$ classes of vertices. Then, if $b$ is odd, $g$ is an automorphism, and if $b$ is a prime number then $g$ acts transitively into the vertices of the group 0. Thus, for $b$ prime there is only one class modulo isomorphism and the graph is vertex-transitive or node-symmetric.

*4.1.4. Discussion on the Global Link Arrangement Selection.* While the global link arrangement is important to fully characterize a topology, simulations show that the impact of the selected arrangement on network performance under uniform traffic is, in general, negligible[5]. However, specific arrangements have different topological properties, such as symmetries and the possibility of defining multiple classes of vertices in the network, what can be exploited to simplify routing. The *palmtree* and any subgraph of the Hamming graph allow for a natural vertex coloring with $\frac{a}{2}$ (for $a$ even) and $a$ colors respectively, in a way such that every global link has the same color in its two endpoints. This property will be used by the routing mechanisms in Section 6.

Additionally, as studied in [García et al. 2012], for certain traffic patterns, pathological saturation of local links occurs when using the Valiant variant from [Kim et al. 2008], which does not employ local misrouting in the intermediate group. This occurs when all the nodes in group $i$ send traffic to nodes in group $i + \Delta_2 \pmod b$. The saturation arises in the intermediate group, in which almost all of the traffic received from the $\Delta_2$ global links from a router leaves the group using the same neighbour router. The single link between these routers becomes a performance bottleneck. With the *consecutive* or *palmtree* arrangements, all traffic received by router $i$ needs to leave by router $i + 1$, leading to a throughput limit of $1/\Delta_2$ *phits*/*node*/*cycle* (a *phit* is the

[5]Different traffic patterns such as global permutations could be impacted by the global link arrangement.

amount of data transferred on a link on a single cycle). In the *circulant-based* arrangement only $\Delta_2/2$ of such links would compete for the same local link, leading to a throughput limit of $2/\Delta_2$ *phits/node/cycle*. A *random* arrangement would typically eliminate this problem, at the cost of regularity in the network. In any case, such pathological performance issues can be solved using the original implementation of Valiant routing [Valiant 1982] (as discussed in [Prisacari et al. 2014] and implemented in [Faanes et al. 2012]) or allowing for local misrouting in the intermediate group (as in the OLM routing [García et al. 2013a] that we employ as a reference in Section 7).

## 5. DRAGONFLY TOPOLOGIES WITH GLOBAL TRUNKING

This section considers trunking in dragonfly topologies and discusses how the Hamming graph responds to the definition of a dragonfly topology with trunking. Based on this observation, Hamming graphs and canonical dragonfly topologies are considered as the two extreme possibilities of trunking and the spectrum between them is studied considering the corresponding balancing conditions.

The *trunking level* in a topology refers to the number of parallel links that are employed to increase the aggregated bandwidth, increasing also the number of router ports used. In a dragonfly topology, *local trunking* refers to parallel links between pairs of routers within a group. Such parallel links between pairs of routers are typically known as a LAG (Link Aggregation Group). This LAG could be also implemented in a Hamming topology to increase bandwidth between routers in the same row or column. The *global trunking* level $t$ is the number of global links between every pair of groups. In this case, there are multiple alternative implementations. Trunk links can join a single pair of routers (LAG), one router in a group and multiple routers in the other (often called as Multi-Chassis LAG, MC-LAG), or different pairs of routers.

Unless otherwise noted, we will always refer to global trunking between different pairs of routers, that increases both bandwidth and reliability. As discussed in [Faanes et al. 2012], trunking is required to retain optimal global bandwidth in systems with less groups than the maximum allowed. A dragonfly with trunking is specified by the number of routers per group $a$, the groups $b$, the global links per router $\Delta_2$, the global link arrangement and the global trunking $t > 1$ ($t = 1$ for a canonical dragonfly without trunking as defined in Section 4). Dragonflies with global trunking obey the relation:

$$a\Delta_2 = t(b-1). \tag{1}$$

The vertices of a Hamming graph $K_a \square K_b$ can be partitioned into $b$ groups by defining the group $y$ as the set $\{(x, y) \mid x \in \mathbb{Z}_a\}$. Clearly these groups are subgraphs isomorphic to $K_a$ and hence the Hamming graph satisfies the definition of the canonical dragonfly topology (complete graphs for local and global topologies) with trunking $t = a$. Between groups $y_1$ and $y_2$ there is the set of $a$ global links $\big\{\{(x, y_1), (x, y_2)\} \mid x \in \mathbb{Z}_a\big\}$.

Therefore from a topological point of view, the Hamming graph $K_{a'} \square K_{b'}$ is a trunked dragonfly with parameters $a = a'$, $b = b'$, $t = a'$ and $\Delta_2 = b - 1$; using a specific global link arrangement which connects all routers in the same position of each group. An example of the Hamming graph represented as a trunked dragonfly can be seen in Figure 4.

### 5.1. Balancing Conditions for the Trunked Dragonfly

The requirements for a balanced trunked dragonfly are studied in detail in this subsection, considering uniform traffic and minimal routing. As discussed before, non-uniform traffic can be made uniform by randomizing it (like Valiant routing) or by other means such as randomizing task placement, so it is not considered in this analysis. The detailed balancing conditions are derived from calculating the average dis-

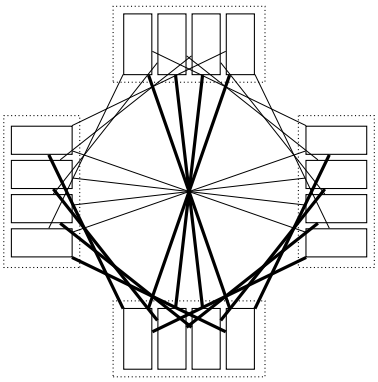


Fig. 4: Hamming graph $K_4 \square K_4$ with nodes organized in groups.

tance on each type of links and relating it to the number of links of each type to be used, considering network trunking.

Let $\mathrm{avg}$ be the average distance, this is, the quotient between the length and the number of all possible minimal paths. This distance can be divided into $\mathrm{avg} = \mathrm{avg}_1 + \mathrm{avg}_2$, with $\mathrm{avg}_1$ being the average number of hops in local links and $\mathrm{avg}_2$ in global links. A similar relation can be established with the total number of edges $|E| = |E_1| + |E_2|$. A balanced network requires

$$\frac{\mathrm{avg}_1}{|E_1|} = \frac{\mathrm{avg}_2}{|E_2|}.$$

Let $\alpha = \frac{\mathrm{avg}_2}{\mathrm{avg}_1}$ represent the relation between the use of each type of links under uniform traffic. Thus, $\alpha$ also represents the relation between the amount of links of each type (global, local) for a balanced network, $\Delta_2 = \alpha \cdot \Delta_1$. In order to approximate $\alpha$, it can be observed that for global links $\mathrm{avg}_2 = \frac{ab-a}{ab-1} \approx 1$. The average distance in local links $\mathrm{avg}_1$ can be derived from the number of possible minimal paths between two groups, ignoring the communication internal to a single group, as follows. There are $a^2$ pairs of source/destination vertices among two given groups; $t$ vertices of each group have a direct global edge to the other group and $a - t$ do not. Hence,

— $t$ pairs are connected by a direct global edge, which is their minimal path, *g*.
— $t(a - t)$ pairs begin at a vertex with a global edge to the other group and finish in one without such edge. Their minimal path is *gl*.
— $(a - t)t$ pairs begin at a vertex without a global edge to the other group but finish in one with such edge. Their minimal path is *lg*.
— $t(t - 1)$ pairs begin and finish at vertices with global edges between the groups, but they are different. Their two possible minimal paths are *lg* and *gl*.
— $(a - t)(a - t)$ pairs begin and finish at vertices without direct global edges. The $t$ minimal paths are *lgl*.

Thus, ignoring the traffic local to a group, $\mathrm{avg}_1 \approx (t^2 - t - 2ta + 2a^2)a^{-2}$. Removing low order terms it becomes $\mathrm{avg}_1 \approx 1 + (\frac{t}{a} - 1)^2$ and $\alpha$ can be approximated as

$$\alpha \approx \frac{1}{1 + (\frac{t}{a} - 1)^2}. \tag{2}$$

Thus, in the Hamming graph $t = a$ and $\alpha = 1$, whereas in the canonical dragonfly $t = 1$ and $\alpha$ tends to $\alpha \to 1/2$ for $a \to \infty$. The approximate dragonfly balancing conditions

Table I: Examples of dimensioning the number of groups $b$ of a network with $a = 4$ routers per group, for different levels of trunking $t$ as in Figure 5. Networks with less groups (middle column) require more trunking to be balanced.

| $t$ | Limit for $b$ using $\alpha = 1$ in (3) | $b$ for a balanced network, according to (4) | Limit for $b$ using $\alpha = 1/2$ in (3) | Actual network example |
|---|---|---|---|---|
| 1 | 13 | 8.7 | 7.5 | canonical dragonfly $b = 9$ |
| 2 | 7 | 5.8 | 4.0 | $b = 5$, $\Delta_2 = 2$ |
| 3 | 5 | 4.8 | 3.0 | $b = 5$, $\Delta_2 = 3$ |
| 4 | 4 | 4.0 | 2.5 | Hamming $K_4 \square K_4$. |

presented in Section 4 ($2\Delta_2 \approx \Delta_1$ or $a = 2\Delta_2$) no longer hold when the network employs trunking, since $\alpha$ becomes larger than $1/2$ so the ratio between global and local router ports needs to increase.

The parameter $\alpha$ and its relation with the number of edges of each type in a balanced network is:

$$\alpha = \frac{\text{avg}_2}{\text{avg}_1} = \frac{|E_2|}{|E_1|} = \frac{\frac{tb(b-1)}{2}}{b\frac{a(a-1)}{2}} = \frac{t(b-1)}{a(a-1)}. \tag{3}$$

From the expressions of $\alpha$ in 3 and 2, the following balancing condition is obtained:

$$b = 1 + \alpha\frac{a(a-1)}{t} = 1 + \frac{1}{1 + (\frac{t}{a} - 1)^2}\frac{a(a-1)}{t}. \tag{4}$$

The balancing condition (4) can be related with the cardinal equation (1):

$$a\Delta_2 = t(b-1) = a(a-1)\alpha. \tag{5}$$

In the extreme case of Hamming graphs, $t = a$ and $\alpha = 1$, and hence the balancing condition is $b = a$ or equivalently $\Delta_2 = a - 1$. This was already known since it is the case of the Hamming graph being edge-transitive.

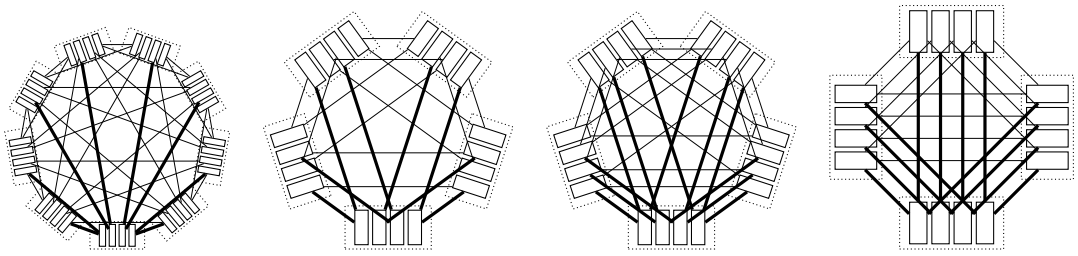
Table I shows in the middle column several dimensioning examples for groups of $a = 4$ routers, and in the sides the valid range for the number of groups $b$ that keep $\alpha \in [\frac{1}{2}, 1]$. Since the result from the balancing equation (4) is not necessarily integer, an approximation with integral values is presented on the right. The corresponding topologies can be seen in Figure 5. It can be observed that for $t = 1$ (no trunking) the balancing condition is close to the lower limit given for $\alpha = 1/2$ on its right, whereas for $t = a = 4$ (maximum trunking) the result is close to the upper limit for $\alpha = 1$ on its left. Also, it is clear that the less groups of a dragonfly are present, the higher the trunking level is required for the topology to be balanced.

### 5.2. Arrangements for Dragonflies with Global Trunking

Trunking increases the number of possible arrangements of a dragonfly network. Subsection 4.1 introduced several possible global link arrangements for dragonflies without trunking. Those same configurations can be directly applied when using LAG, this is, multiple parallel links between each pair of linked routers. This section extends the arrangements presented in Subsection 4.1 to use trunking with disjoint pairs of routers for parallel links, to maximize fault tolerance. We denote such configurations as "extended."

In general, building a trunked dragonfly with an arbitrary arrangement requires:

(1) In each group, for each router select $\Delta_2$ (generally different) groups. Among all the routers of the group, each other group must have been selected exactly $t$ times.



(a) $t = 1$ (b) $t = 2$ (c) $t = 3$ (d) $t = 4$

Fig. 5: Dragonfly networks with extended palmtree arrangement; a=4 routers per group and b groups, according to Table I.

(2) For every pair of groups $A, B$, find in $A$ the $t$ routers which have selected $B$ and in $B$ the $t$ routers which have selected $A$. Then there are $t!$ ways to add the $t$ global links between the two collections of routers found.

*5.2.1. Extended Consecutive Arrangement.* The consecutive arrangement presented in 4.1.1 is generated adding global edges in a greedy way. However, for $t > 1$ any greedy strategy ends connecting some router to several other routers of the same remote group (what we denoted as multichassis-LAG). Since this section searches for solutions that connect disjoint pairs of routers for maximum fault tolerance, we do not present an extension of the consecutive arrangement.

*5.2.2. Extended Palmtree Arrangement.* A generalization of the palmtree arrangement is defined here for any trunking level which obeys equation (1). This configuration employs disjoint pairs of routers for each parallel link between groups. The router $x$ of group $y$ is connected by global links to the following $\Delta_2$ routers:

$$\{\big(a - x - 1, \mathrm{rem}(y + 1 + \mathrm{rem}((a - x - 1)\Delta_2 + k - 1, b - 1), b)\big) \mid k \in \{1, \ldots, \Delta_2\}\}$$

As in the base case, $(x, y) \mapsto (x, \mathrm{rem}(y + 1, b))$ and $(x, y) \mapsto (a - x - 1, \mathrm{rem}(b - y, b))$ are automorphisms of the extended palmtree. Hence, there are are most $\lfloor \frac{a}{2} \rfloor$ isomorphism classes. The graphs in Figure 5 employ such arrangement and, as stated before, they correspond to the examples in Table I. The graph obtained for $a = b$ is very similar to the Hamming graph but not isomorphic to it; this is the case of the last graph in Figure 5. A representation of such graph with nodes organized in rows and columns is presented in Figure 6, providing a visual comparison with the Hamming graph presented in Figure 1. It is remarkable that a *lg* path exists for any pair of routers with this last arrangement, enabling a deadlock-free DOR routing.

*5.2.3. Extended Circulant-based Arrangement.* Subsection 4.1.3 discussed the construction of canonical dragonflies as subgraphs of the Hamming graph, by finding a decomposition of the complete graph $K_b$ into $a$ regular subgraphs. When using trunking, the construction relies on finding a decomposition of $t > 1$ copies of a complete graph, this is, of a multigraph with $t$ edges between every pair of vertices.

The arrangements composed of multiple circulant graphs from Subsection 4.1.3 can be easily extended to the case of global trunking, under the restrictions of equation (1), even $\Delta_2$ and odd $b$. Specifically, the following connectivity pattern generates a dragonfly with trunking $t$: vertex $i$ in group $j$ is connected to vertices $i$ in groups $j \pm (\mathrm{rem}(\frac{\Delta_2}{2} i + k, \frac{b-1}{2}) + 1) \pmod b$ for every integer $k \in \{0, \ldots, \Delta_2/2 - 1\}$.

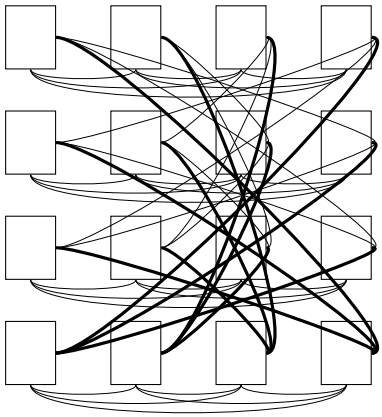


Fig. 6: Palmtree arrangement for $t = a = 4$; vertices organized in rows and columns.

## 6. DEADLOCK-FREE ADAPTIVE ROUTING IN DRAGONFLIES WITH TRUNKING

As discussed in Section 4, distance-based deadlock-free routing mechanisms proposed for dragonflies require as many VCs as hops allowed through a given type of network link. Such implementations can be costly and complex, and tie the number of VCs with the maximum path length. However, Hamming graphs allow for deadlock avoidance mechanisms based on route restrictions (DOR). Such a mechanism does not require VCs.

Section 5 showed how Hamming graphs and dragonflies with trunking can be seen as members of the same family. In this section, three alternative routing mechanisms are introduced for dragonflies with global trunking, based on a variation of the route restriction mechanism employed in Hamming graphs. A DOR mechanism is equivalent to coloring all the links in the network with one of two colors, according to their dimension, and following paths that obey a certain color order. Similarly, our mechanisms impose a selection of the global links in the path, from those $t$ specified by the trunking level. They rely on coloring the routers, which is possible when the global link configuration is an extended palmtree or a subgraph of the Hamming graph (as defined in Subsections 5.2.2 and 5.2.3), what highlights the importance of a careful selection of the global link connectivity.

DOR can be safely used with trunking level $t = a$. With $a > t \geq 2$, cyclic dependencies in minimal routing can be avoided by deciding which of the $t$ global links to use each time, based on two router colors and without relying on VCs; as we will see next, it requires $t \geq 2$. For adverse traffic patterns, a variant of Valiant routing (which sends traffic to an intermediate network router) can be implemented without VCs, requiring $t \geq 4$. These two mechanisms are oblivious. Finally, an adaptive mechanism can be implemented, which selects between the minimal or Valiant paths depending on network conditions, requiring again $t \geq 4$. These three mechanisms are detailed next and evaluated in Section 7.

### 6.1. Oblivious Minimal Deadlock-free Routing for $t \geq 2$

In this subsection a deadlock-free routing mechanism denoted “2-color minimal” is introduced for dragonflies with $t \geq 2$ global links between pairs of groups. Deciding which of the $t$ links to use for each packet can prevent deadlock; $t = 2$ will be assumed from

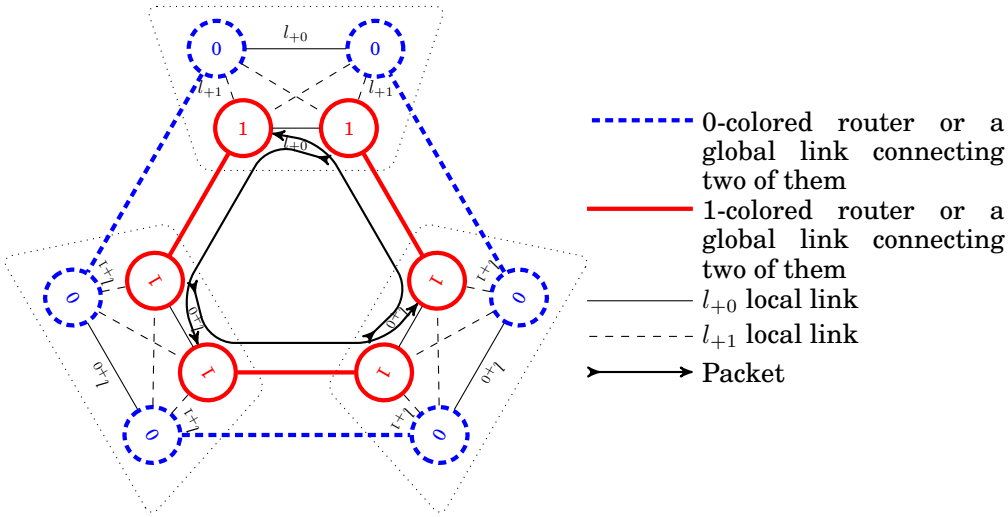



Fig. 7: Coloring of routers with 0 or 1 and the local links with +0 or +1. The cyclic dependency presented would be avoided using the color-ordering rules, since at least one of the messages must follow the $l_{+1}$ local channels.

here onwards although our mechanism is still valid for larger values of $t$. However, in such cases the proposals of the next subsection present several advantages.

Every router in the network will be colored with one of two colors, say 0 and 1. Considering an even number of routes per group $a$, half of them receive each color. Global links should be arranged so they only connect vertices with the same color, which implies a restriction in the global link arrangement. The *extended palmtree* for $a \geq 4$ and any subgraph of the Hamming graph for $a \geq 2$ satisfy this restriction since they divide vertices into several classes. Local links are labelled according to the difference of the color of their endpoints, modulo 2. Thus we have "+0" and "+1" local links, depending on whether they connect vertices with the same or different color, respectively. They will be denoted $l_{+0}$ and $l_{+1}$. A simple three-group example is presented in Figure 7.

The routing mechanism will vary depending on the respective colors of the source and destination routers. Routes with source and destination of different colors will need to employ up to one $l_{+0}$ and one $l_{+1}$ local links. The $l_{+0}$ link always will be selected in the source group and the $l_{+1}$ in the destination group. Implicitly, this forces the selection of the global link to be used, which will have the same color as the source router. This routing restriction prevents dependencies from $l_{+1}$ to $l_{+0}$ local links, which furthermore implies that any possible cyclic dependencies are completely composed either of $l_{+0}$ local links or of $l_{+1}$ local links. For routes in which endpoints have the same color, the path must contain two $l_{+0}$ or two $l_{+1}$ local links. Selecting which one is employed is done in a careful way to avoid deadlock. Our mechanism employs the $l_{+0}$ local links when the destination group index is larger than the source index and the $l_{+1}$ local links otherwise. Again, this implies a selection of the global link to traverse. Alternative orderings between the groups can be used, as long as they guarantee that directed cycles do not appear in the global topology.

The proposed mechanism is deadlock-free by construction: multiple paths between routers with different color never form cycles because they follow local links in an ascending order, and paths between routers of the same color never form cycles because they employ different links when the group index is increased or decreased. An exam-

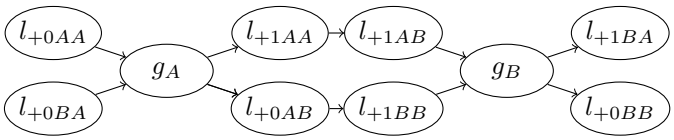



Fig. 8: A precedence of links using $t = 4$ which allows for routes *lgl* and *lglllgl*. Allowed paths flow from left to right, and parallel routes represent different alternatives, one of which is chosen depending on the labels of the source and destination routers.

ple is presented in Figure 7, in which three paths between routers of different groups always employ $l_{+0}$ local links. Such cycle is forbidden with the proposed mechanism, since at least one of the paths will decrease the group index and thus will be forced to employ the $l_{+1}$ links. Finally, it should be noted that under uniform traffic all links are used similarly. There are $(\frac{a}{2})^2$ "+1" and $2\binom{a/2}{2} = \frac{a}{2}(\frac{a}{2} - 1)$ "+0" local links per group. Their ratio tends to 1 for large $a$. Global link usage is completely balanced, according to the color of the source and destination routers.

### 6.2. Oblivious Minimal and Non-minimal Deadlock-free Routing for $t \geq 4$

Non-minimal Valiant routes like *lglllgl* are required to balance load and avoid bottlenecks under adverse traffic patterns. This section introduces a non-minimal routing for dragonflies with $t \geq 4$ global links between pairs of groups denoted as "4-color non-minimal," which does not need VCs for deadlock-freedom. Additionally, by traversing only the first or the second half of the allowed path, routes with a single global hop can be employed. Such routing will be denoted as "4-color minimal," despite using in some cases one extra local hop. This minimal routing is less restrictive than the previous mechanism for $t = 2$, what will be patent in the performance results of the next section.

Like in the previous subsection, this new mechanism relies on a coloring of the routers that allows to classify and order the local and global links considering a directed graph. Unlike the previous mechanism in which the order was only relevant for local links, in this case the link order will be strict. Considering the possible paths, this requires 4 colors for routers, 2 colors for global links and 8 colors for local links; 6 colors for local links are not enough to generate a balanced use of the network links, as we will see later. The four router colors will be labelled with one number and one letter $\{0A, 0B, 1A, 1B\}$. Every global link joins routers of the same color, what is possible for the extended palmtree with $a \geq 8$ and for subgraphs of the Hamming graph with $a \geq 4$. Global links will receive one of two labels, $A$ or $B$, the same that of their endpoints. By contrast, local links receive one of eight labels. A local link from a router labelled $xP$ to a router labelled $yQ$ will be labelled as $+zPQ$ where $z \equiv y - x \pmod 2$, $P, Q \in \{A, B\}$ and $x, y, z \in \{0, 1\}$.

In order to provide a deadlock-free routing, an ordering of the links is required. That is, if $\alpha$ and $\beta$ are two classes of links with $\alpha \prec \beta$ ($\alpha$ preceding $\beta$), then in every possible route there will be at most one link of each class and then the link of class $\alpha$ will appear earlier in the route than the link of class $\beta$. The partial order of global links will always be $g_A \prec g_B$, as it is required in a complete graph $K_b$. Considering local links, our routing employs the complete ordering $l_{+0AA}, l_{+0BA} \prec g_A \prec l_{+1AA} \prec l_{+1AB} \prec l_{+0AB} \prec l_{+1BB} \prec g_B \prec l_{+1BA}, l_{+0BB}$, which allows for the paths shown in Figure 8 in which every node represents a link in the path.

This ordering allows to communicate every pair of nodes. The first local link is selected between $l_{+0AA}$ or $l_{+0BA}$ depending on the label $A$ or $B$ of the source router. Similarly, the last local link allows to select the $A/B$ label of the destination router and

Table II: Simulation parameters.

| | | | |
|---|---|---|---|
| Group size | $a = 24$ routers, 312 nodes | Link latency (local/global) | 10/100 cycles |
| Number of groups | $b = 79$ groups | Packet size | 8 phits |
| Network size | 24,648 computing nodes | Buffer capacity(local/global) | 32/256 phits |
| Router size (ports) | $R = 49$ | Virtual channels per port | 3 (except Valiant, 4) |
| | ($\Delta_0 = 13$, $\Delta_1 = 23$, $\Delta_2 = 13$) | Switching policy | Virtual Cut-Through |
| Global link arrangement | Extended palmtree | Router arbitration | Random |
| Router model | input-queued | Router speedup | No |

the middle branch allows to select the change +0/+1 of the whole path. For example, a route from $0A$ to $0A$ must be $l_{+0AA}, g_A, l_{+0AB}, l_{+1BB}, g_B, l_{+1BA}$. This ordering restricts the class of the middle router, $0B$ in the previous example, which illustrates the restriction of routes applied. However, for any pair of $0A$ source and destination routers in different groups, this mechanism allows to select any of the $0B$ routers in the network as the intermediate router of the Valiant path. Similar routes can be calculated for the other 15 color combinations of source-destination pairs. Thus, it is similar to Valiant [Valiant 1982] but with the intermediate node restricted to a fourth of the total nodes.

The same link ordering can be used for minimal routing. Depending on the labeling of the source and destination, either the first 3 links or the last 3 links will be used. For example, packets going minimally from a $0A$ router to a $1A$ router will need to use a route $l_{+0AA}, g_A, l_{+1AA}$ (first half); and to go minimally from a $0A$ to a $0A$ the path is $l_{+1AB}, g_B, l_{+1BA}$ (second half). A priori, one could expect a small loss of performance when using this routing, since some packets which could minimally route as *gl* increase their paths in routes *lgl* to satisfy the coloring criteria. However, as it will be observed in the next section, our deadlock-free routing algorithms perform similarly to the references and in some cases outperform them. It is also interesting to remark that the minimum routing mechanism for $t \geq 4$ performs better than the one for $t \geq 2$.

As both minimal and non-minimal routes are allowed with the same ordering of links, adaptive routing can be employed by selecting one of them at the source. This requires the use of some decision mechanism, such as UGAL [Singh 2005] and using congestion information from neighbors as in [Jiang et al. 2009].

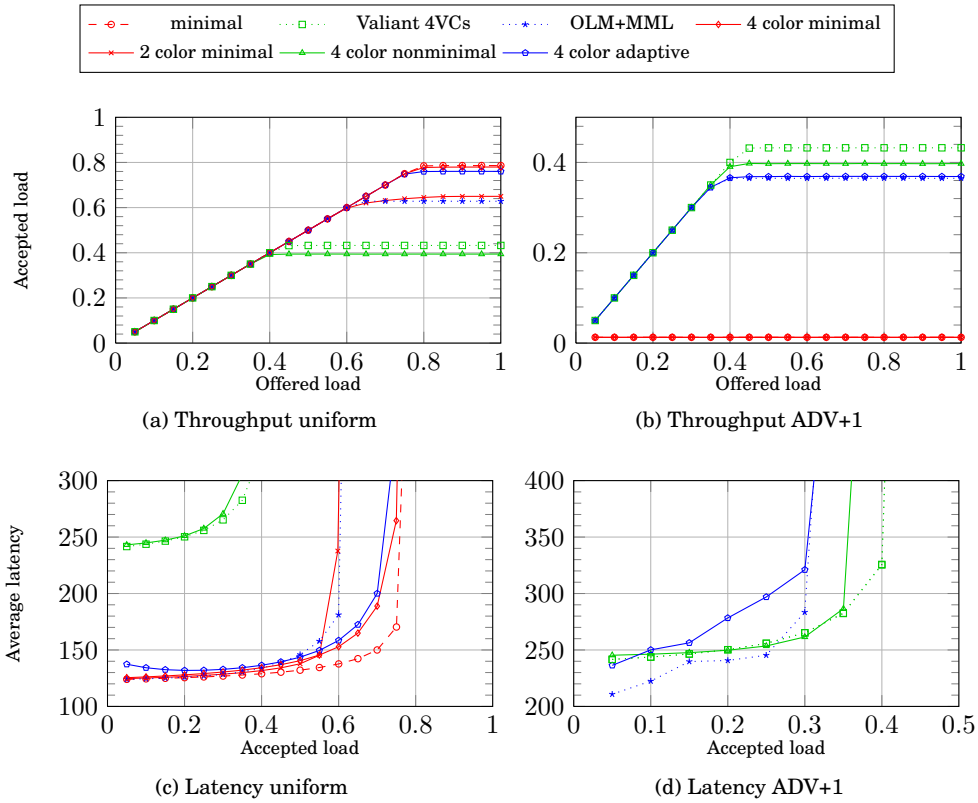
## 7. EVALUATION

This section shows the performance of the proposed routing mechanisms. We used the FSIN cycle-accurate network simulator [Perez and Miguel-Alonso 2005], modified to support trunked dragonflies with variable-length links. We have simulated a network of input-buffered routers with $a = 24$ routers per group and global trunking $t = 4$ using the *extended palmtree* arrangement. We rounded the number of groups to $b = 79$, what provides a balanced topology according to equation (4) requiring routers with $R = 49$ ports and leading to a total of 24,648 computing nodes. The complete set of parameters is presented in Table II and the routing mechanisms characteristics in Table III. We have implemented the following oblivious routing mechanisms:

— *Minimal*: Hierarchical routing first to the destination group and then to the destination router, as described in [Kim et al. 2008]. The global link of the path is selected as follows: if the source router has a direct link to the destination group, select it; otherwise, select an available link to any random router with a direct link to the destination. This mechanism is the reference for uniform traffic, although it only exploits 2/1 VCs, therefore suffering from more HoLB.
— *Valiant*, [Valiant 1982]: Nonminimal routing composed of two parts: Minimal to a random intermediate router and then minimal to the destination, as defined before. This is the reference mechanism for adversarial traffic.

Table III: Parameters of each routing mechanism.

| Routing | Adaptive | Min VCs (local/global) | Min. trunking |
|---|---|---|---|
| Minimal | No | 2/1 | 1 |
| Valiant | No | 4/2 | 1 |
| 2-color minimal | No | 1/1 | 2 |
| 4-color minimal/nonminimal | No | 1/1 | 4 |
| OLM | Yes | 3/2 | 1 |
| 4-color adaptive | Yes | 1/1 | 4 |




(a) Throughput uniform

(b) Throughput ADV+1

(c) Latency uniform

(d) Latency ADV+1

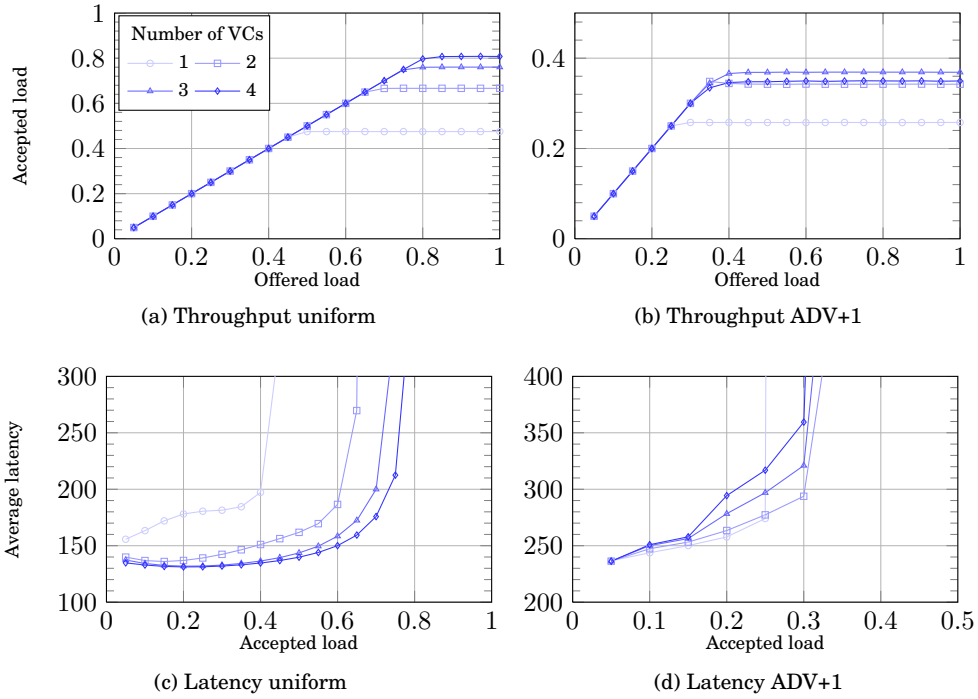
Fig. 9: Throughput and average latency for uniform and ADV+1 traffic.

— *2-color Minimal, 4-color Minimal* and *Nonminimal*: As described in Section 6.

Additionally, two adaptive mechanisms have been implemented:

— *OLM-MML*: An in-transit adaptive routing mechanism described in [García et al. 2013a], using the MM+L global misrouting policy from [García et al. 2013b]. This mechanism has been selected because it provides similar or better performance than the naïve PAR6/2 from the same paper, while requiring less virtual channels.
— *4-color adaptive*: The 4-color routing presented in Section 6.2, implementing Piggybacking [Jiang et al. 2009] to adaptively select between minimal (*lgl*) or nonminimal (*lgllgl*) paths at injection time, using information from the neighbour routers.

Figure 9 shows latency and throughput results under minimal and adverse traffic patterns. As expected, minimal routings give the best results for uniform traffic but




(a) Throughput uniform (b) Throughput ADV+1

(c) Latency uniform (d) Latency ADV+1

Fig. 10: Throughput and average latency for uniform and ADV+1 traffics varying the number of virtual channels

accept an insignificant amount of the adverse traffic. 2 and 4 colors minimal routings obtain approximately the same latency with a slight advantage for the 2-color implementation at medium loads, but higher throughput for the 4-color variant. The lower throughput of the 2-color variant comes from some groups using local links of type $l_{+0}$ in most of the first local hops, especially in those with a low group index: $l_{+0}$ is used in the first local hop both when the source and destination colors are the same, and when they differ and the destination group has a higher index. An alternative more balanced ordering between each pair of groups (without introducing cyclic dependencies) would mitigate this effect. Interestingly, the 4-color adaptive mechanism throughput is close to the maximum, despite being adaptive (and thus making some erroneous decisions).

The 4-color nonminimal implementation is relatively close to the performance of Valiant, despite its route restrictions and the use of one VC less for HoLB, which explain the lower throughput. The adaptive variant reacts very well to the traffic pattern, almost reaching the accepted load of its respective oblivious minimal and non-minimal counterparts. It also accepts more load than the OLM reference for adaptive routing. However latency is higher than OLM and Valiant especially for adverse traffic; this comes from worse decisions using source-routing in the 4-color mechanism, compared to in-transit adaptivity of OLM, which can save intermediate hops when no congestion is detected (for ADV+1 traffic, one hop in the intermediate group).

Figure 10 shows the performance of 4-color adaptive using different number of virtual channels. As the mechanism does not restrict VCs at all, any number of buffers can be employed, unlike the fixed requirement imposed by other policies. Furthermore, there is freedom in the way to use the VCs; in these simulations the VC is selected

randomly among those available. The HoLB problem is seen to have a great impact especially when not using VCs (i.e. 1 VC), but the performance with just 2 VCs noticeably improves. It is interesting to see that for adversarial traffic the delay increases with the number of VCs; this is explained by the higher capacity of the network buffers before the bottleneck, which increases the number of stored packets. When the buffer count is low, explicit mechanisms could be employed to mitigate the HoLB performance issues, such as internal router speedup or virtual-output queueing [Tamir and Frazier 1988]. Such mechanisms are widely known and they have not been explored in this work.

## 8. 3-LEVEL DRAGONFLIES

While dragonfly networks provide a very competitive scalability, larger networks can be built if the number of hierarchy levels is increased at the cost of a longer diameter. Alternatively, very large systems can be built based on moderate-radix routers (such as the integrated routers discussed in the introduction) if multiple levels are employed. This section explores the properties of 3-level dragonflies. More levels could be considered but the analysis would be similar and as we will see, the scalability grows very quickly with the number of levels, making configurations with more than 3 levels unlikely.

For 3-level dragonflies, links can be considered as local ($l$ or 1), medium ($m$ or 2) and global ($g$ or 3). For notation, a 1-level group contains $a$ routers, a 2-level group contains $b$ 1-level groups and there are $c$ 2-level groups in the whole network. The degree will be extended to $\Delta = \Delta_1 + \Delta_2 + \Delta_3$. Two trunking levels can be considered in this case: $t_2$ will be the number of 2-links between every pair of 1-groups and $t_3$ will be the number of 3-links between every pair of 2-level groups.

The network average distance can be decomposed as $\mathrm{avg} = \mathrm{avg}_1 + \mathrm{avg}_2 + \mathrm{avg}_3$. Similar to the 2D trunked dragonfly studied in Section 5.1, balancing conditions can be derived from a calculation of the relations between the average distance on each type of link. It can be defined as $\alpha = \frac{\mathrm{avg}_2}{\mathrm{avg}_1}$ and $\beta = \frac{\mathrm{avg}_3}{\mathrm{avg}_2}$. The equations of size and balance (5) generalize easily:

$$a\Delta_2 = t_2(b-1) \approx a(a-1)\alpha; ab\Delta_3 = t_3(c-1) \approx t_2 b(b-1)\beta$$

From which the following expressions of the degrees are obtained:

$$\Delta_2 \approx (a-1)\alpha \approx \Delta_1\alpha; \Delta_3 \approx (a-1)\alpha\beta \approx \Delta_2\beta$$

In 3-level dragonflies a medium-link arrangement and a global-link arrangement can be defined. Any combination could be chosen such as (random, palmtree) or (random, random). The definition for the global-link arrangement equals the one in the 2-level case only when $t_2 = 1$, otherwise it needs some adaption.

In this 3-level case, minimal routes are in general *lmlglml*, thus requiring up to 4 VCs. The classical Valiant [Valiant 1982] (using an intermediate router) duplicates the route and would need up to 8 VCs. Shortened versions as in [Kim et al. 2008] can be defined; using an intermediate 1-level group the routes would be *lmlglmlml-glml* requiring only 7 VCs; and using as intermediate a 2-level group routes would be *lmlglmlglml* requiring only 6 VCs. However only the original Valiant routing makes traffic completely uniform. This large number of VCs can be reduced by increasing one or both of the trunking levels, and applying the studied coloring techniques.

Considering two levels, a family of topologies between the Hamming graph and the dragonflies was built in Section 5 by modifying the parameter $t$. This is depicted by the horizontal line on top of Figure 11. With three levels, there are two parameters ($t_2$ and $t_3$) that can be modified, what extends the design space to a plane, represented in the lower part of the same Figure. Some of the most remarkable properties of this family of

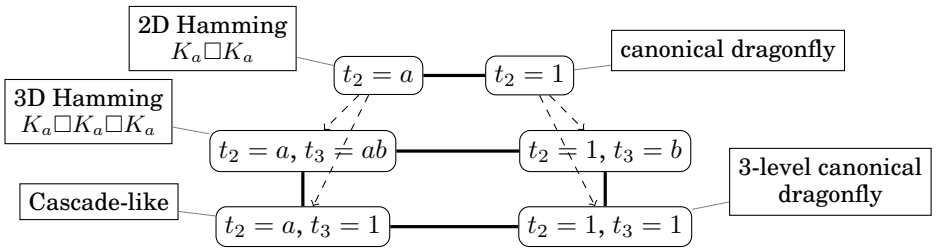



Fig. 11: Classification of 3-level networks. Nodes correspond to extreme cases. Solid lines correspond to changes in one of the trunking levels. Dotted arrows represent the increase from two to three dimensions, where a trunking level for the new dimension must be chosen.

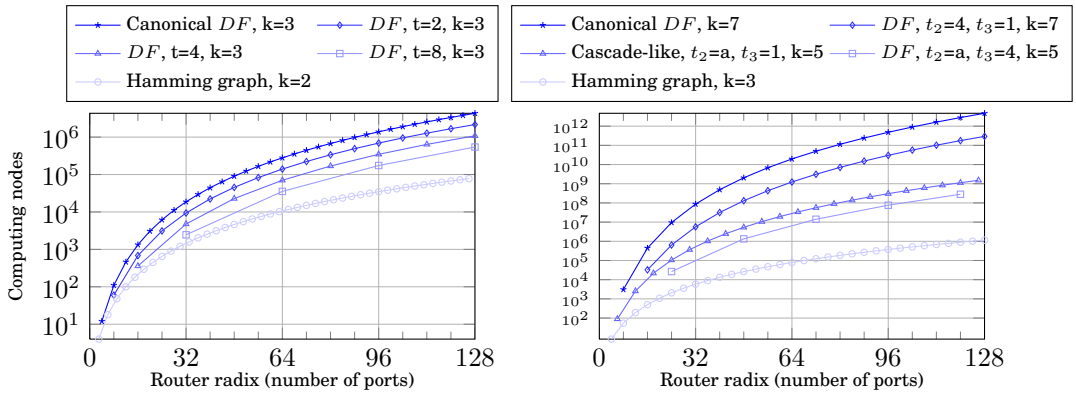
Table IV: Characteristics of the extreme cases (respect to trunking) of 2D and 3D balanced dragonflies. $a$, $b$ and $c$ routers per dimension. $\Delta_0$ compute nodes per router. Routers with $R$ ports (radix). Number of compute nodes approximate.

| name | $t_2$ | $t_3$ | balancing conditions | link use relations | routers | compute nodes | general route |
|---|---|---|---|---|---|---|---|
| 2-levels | | | | | $ab$ | $ab\Delta_0$ | |
| canonical dragonfly | 1 | - | $b = 1 + a(a-1)/2$ | $\alpha \approx 1/2$ | $\approx a^3/2$ | $(R^4 + 4R^3 + 12R^2)/2^6$ | *lgl* |
| Hamming $K_a \square K_a$ | $a$ | - | $a = b$ | $\alpha = 1$ | $a^2$ | $(R+2)^3/3^3$ | *lg* |
| 3-levels | | | | | $abc$ | $abc\Delta_0$ | |
| 3-level canonical dragonfly | 1 | 1 | $b = 1+a(a-1)/2$, $c = 1 + b(b-1)/2$ | $\alpha \approx \beta \approx 1/2$ | $\approx a^7/16$ | $R^4(R+2)^4/2^{14}$ | *lmlglml* |
| Cascade-like | $a$ | 1 | $a = b$, $c-1 = a^2(a-1)/2$ | $\alpha = 1$, $\beta \approx 1/2$ | $\approx a^5/2$ | $(R^6+12R^5+54R^3)/(2^2 3^6)$ | *lmglm* |
| — | 1 | $b$ | $c - 1 = b - 1 \approx a(a-1)/3$ | $\alpha \approx 1/3$, $\beta = 1$ | $\approx a^5/3^2$ | $R^6/(2^6 3^3)$ | *lmlgl* |
| Hamming $K_a \square K_a \square K_a$ | $a$ | $ab$ | $a = b = c$ | $\alpha = \beta = 1$ | $a^3$ | $(R+3)^4/2^8$ | *lmg* |

networks are presented in Table IV. There are three corner cases which are very relevant, being the first of them the canonical 3-level dragonfly without any trunking. The opposite case is the 3D Hamming graph $K_a \square K_b \square K_c$, which is balanced for $a = b = c$. Notably, according to this classification there exists another 3-level corner configuration (using $t_2 = a$) which employs 2D Hamming graphs in the two lower levels, but no trunking in the highest level. This is equivalent to a 2-level dragonfly in which a Hamming graph is used for the local group topology, as in Cray Cascade [Faanes et al. 2012]. It is due to the fact that the Hamming graph can be seen as a 2-level dragonfly as discussed in Section 5. Interestingly, their design combines route-restriction and distance-based deadlock-avoidance mechanisms (DOR in the 2D Hamming and increasing order of VCs otherwise).

The remaining corner case in the design space (the one with no name in Table IV) employs $t_3 = b$, as many global 3-level links as 2-level groups. With such trunking and a proper arrangement, a global link leads directly to the destination 1-level group, shortening minimal routes to *lmlgl*. This leads to $\beta = 1$ and $\alpha \gtrsim 1/3$. Larger values of trunking, up to $t_3 = ab$ could shorten paths to *lmlg*, but this clearly overdimensions the network.

In a $n$-level network, up to $\Delta_0 = \Delta_n$ compute nodes can be connected per router with maximum throughput. Larger values $\Delta_0 > \Delta_n$ lead to oversubscribed networks




(a) 2-level Dragonflies with trunking level $t$.

(b) 3-level Dragonflies with trunking levels $t_1, t_2$.

Fig. 12: Scalability of different network configurations.

and lower values to waste of the network maximum bandwidth. For these concentration values, routers with radix $R = \sum_{i=0}^{n} \Delta_i = \Delta_0 + \Delta$ are required. Then for 2-level networks it is obtained $R = (a-1)(1+2\alpha)$ and for 3-level balanced networks $R = (a-1)(1+\alpha+2\alpha\beta)$. Table IV summarizes the maximum number of compute nodes in a network ($abc\Delta_0$) for a given router radix $R$, along with balancing conditions and minimal routes employed in each case.

Figure 12 depicts the system size for different router radices and trunking levels, considering 2 and 3 levels. Notice the logarithmic vertical axis. Figure 12a corresponds to the upper line in Figure 11. The 2D Hamming graph ($t = a$, diameter $k = 2$) and the canonical 2-level dragonfly topology ($t = 1, k = 3$) are extreme cases. Between them are multiple alternatives with variable trunking and smaller size than the canonical dragonfly. As discussed before, trunking is required to build systems smaller than the maximum achievable size for a given router radix. Figure 12b represents the scalability of certain designs in the lower rectangle of Figure 11, scaling from a 3D Hamming graph ($t_2 = a$, $t_3 = a^2$, $k = 3$) to a Cascade-like Dragonfly topology ($t_2 = a, t_3 = 1, k = 5$) and then to a canonical 3-level Dragonfly without trunking (with diameter $k = 7$). These figures clearly highlight two issues: the trade off between diameter, degree and scale (the $d - k$ problem discussed in the introduction) and the need of global trunking to build systems that do not reach the maximum size for a given router and diameter, which can be in the order of millions of nodes.

## 9. CONCLUSIONS

Hamming graphs and dragonflies have been extensively studied in the technical literature. However, Hamming graphs have been revisited multiple times without recognizing its previous existence, whereas the dragonfly topology definition was intentionally very loose and not completely specified. In this work we have characterized topologically both networks including their balancing conditions and provided precise definitions for the dragonfly topology. The relation between both graphs has been studied. With a proper global link arrangement, canonical dragonfly topologies are subgraphs of Hamming graphs. On the other hand, Hamming graphs can be seen as an extreme case of a dragonfly network with trunking, showing that both networks are actually part of the same broader family.

Based on this classification, the typical deadlock-free DOR mechanism used in Hamming graphs has been adapted to dragonflies with trunking, based on a coloring and ordering of the network resources. Trunking $t = 2$ allows for 3-hop paths while trunking $t = 4$ allows for 6-hop paths and traffic randomization, in both cases without a restriction on the number or use of virtual channels in the system. Evaluations show that performance results are competitive with alternative mechanisms based on VCs, but they allow for implementations with more VCs to prevent Head-of-Line Blocking and increased performance, or less VCs to reduce implementation cost. The overall cost of this routing mechanism can be obviously higher than an equivalent VC-based routing, because it requires more router ports rather than more VCs. However, in many cases the required trunking is already employed to build a balanced dragonfly of a given size or for adding fault tolerance, so it would imply no extra cost. Finally, this routing would allow to leverage existing 2-level dragonfly router designs to multi-level dragonflies, thus increasing the maximum achievable system size with the same router design.

### Acknowledgments

We are thankful to Carmen Martínez for discussions over vertex-transitive arrangements, to Esteban Stafford for discussing, editing and proofreading the manuscript and to the anonymous reviewers for their valuable suggestions.

## REFERENCES

Jung Ho Ahn, Nathan Binkert, Al Davis, Moray McLaren, and Robert S. Schreiber. 2009. HyperX: Topology, Routing, and Packaging of Efficient Large-scale Networks. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis (SC '09)*. Article 41, 11 pages.

Jung Ho Ahn, Young Hoon Son, and John Kim. 2013. Scalable High-radix Router Microarchitecture Using a Network Switch Organization. *ACM Trans. Archit. Code Optim.* 10, 3, Article 17 (Sept. 2013), 25 pages.

S.B. Akers and B. Krishnamurthy. 1989. A Group-Theoretic Model for Symmetric Interconnection Networks. *IEEE Trans. Comput.* 38 (1989), 555–566.

Baba Arimilli, Ravi Arimilli, Vicente Chung, Scott Clark, Wolfgang Denzel, Ben Drerup, Torsten Hoefler, Jody Joyner, Jerry Lewis, Jian Li, Nan Ni, and Ram Rajamony. 2010. The PERCS High-Performance Interconnect. In *Proceedings of the 2010 18th IEEE Symposium on High Performance Interconnects (HOTI '10)*. IEEE Computer Society, Washington, DC, USA, 75–82.

M.M. Bae and B. Bose. 1996. Resource placement in torus-based networks. In *Parallel Processing Symposium, 1996., Proceedings of IPPS '96, The 10th International*. 327–331.

Abhinav Bhatele, Nikhil Jain, William D. Gropp, and Laxmikant V. Kale. 2011. Avoiding hot-spots on two-level direct networks. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC '11)*. ACM, New York, NY, USA, Article 76, 11 pages.

L.N. Bhuyan and D.P. Agrawal. 1984. Generalized Hypercube and Hyperbus Structures for a Computer Network. *Computers, IEEE Transactions on* C-33, 4 (April 1984), 323–333.

Dhananjay Brahme, Onkar Bhardwaj, and Vipin Chaudhary. 2013. SymSig: A low latency interconnection topology for HPC clusters. In *International Conference on High Performance Computing*. 462–471.

William G Brown. 1966. On graphs that do not contain a Thomsen graph. *Canad. Math. Bull* 9, 2 (1966), 1–2.

Charles Delorme. 1985. Grands Graphes de Degré et Diamètre Donnés. *European Journal of Combinatorics* 6, 4 (1985), 291 – 302.

G. Faanes, A. Bataineh, D. Roweth, T. Court, E. Froese, B. Alverson, T. Johnson, J. Kopnick, M. Higgins, and J. Reinhard. 2012. Cray Cascade: A scalable HPC system based on a Dragonfly network. In *High Performance Computing, Networking, Storage and Analysis (SC), 2012 International Conference for*. 1–9.

M. García, E. Vallejo, R. Beivide, M. Odriozola, C. Camarero, M. Valero, J. Labarta, and G. Rodríguez. 2013b. Global Misrouting Policies in Two-level Hierarchical Networks. In *Proceedings of the 2013 Interconnection Network Architecture: On-Chip, Multi-Chip (IMA-OCMC '13)*. ACM, New York, NY, USA, 13–16.

Marina García, Enrique Vallejo, Ramón Beivide, Miguel Odriozola, Cristóbal Camarero, Mateo Valero, Germán Rodriguez, Jesús Labarta, and Cyriel Minkenberg. 2012. On-the-Fly Adaptive Routing in High-Radix Hierarchical Networks. In *41st International Conference on Parallel Processing (ICPP)*. 279–288.

Marina García, Enrique Vallejo, Ramón Beivide, Miguel Odriozola, and Mateo Valero. 2013a. Efficient Routing Mechanisms for Dragonfly Networks. In *Parallel Processing (ICPP), 2013 42nd International Conference on*. 582–592.

Marina García, Enrique Vallejo, Ramón Beivide, Mateo Valero, and Germán Rodríguez. 2013. OFAR-CM: Efficient Dragonfly Networks with Simple Congestion Management. In *High-Performance Interconnects (HOTI), 2013 IEEE 21st Annual Symposium on*. 55–62.

K. Günther. 1981. Prevention of Deadlocks in Packet-Switched Data Transport Systems. *Communications, IEEE Transactions on* 29, 4 (Apr 1981), 512–524.

Raj Hazra. 2014. Accelerating Insights... in the Technical Computing Transformation. In *International Supercomputing Conference (ISC)*.

A.J.W Hilton. 1984. Hamiltonian decompositions of complete graphs. *Journal of Combinatorial Theory, Series B* 36, 2 (1984), 125 – 134.

A.J. Hoffman and R. R. Singleton. 1960. On Moore Graphs with Diameters 2 and 3. *IBM Journal of Research and Development* 4, 5 (Nov 1960), 497–504.

Nan Jiang, John Kim, and William J. Dally. 2009. Indirect Adaptive Routing on Large Scale Interconnection Networks. In *Proceedings of the 36th Annual International Symposium on Computer Architecture (ISCA '09)*. ACM, New York, NY, USA, 220–231.

John Kim, William J. Dally, and Dennis Abts. 2007. Flattened Butterfly: A Cost-efficient Topology for High-radix Networks. In *Proceedings of the 34th Annual International Symposium on Computer Architecture (ISCA '07)*. ACM, New York, NY, USA, 126–137.

John Kim, Wiliam J. Dally, Steve Scott, and Dennis Abts. 2008. Technology-Driven, Highly-Scalable Dragonfly Topology. In *Proceedings of the 35th Annual International Symposium on Computer Architecture (ISCA '08)*. IEEE Computer Society, Washington, DC, USA, 77–88.

John Kim, William J. Dally, Brian Towles, and Amit K. Gupta. 2005. Microarchitecture of a High-Radix Router. In *Proceedings of the 32Nd Annual International Symposium on Computer Architecture (ISCA '05)*. IEEE Computer Society, Washington, DC, USA, 420–431.

L.E. LaForge, K.F. Korver, and M.S. Fadali. 2003. What designers of bus and network architectures should know about hypercubes. *Computers, IEEE Transactions on* 52, 4 (April 2003), 525–544.

Heather Macbeth, Jana Šiagiová, and Jozef Širáň. 2012. Cayley graphs of given degree and diameter for cyclic, Abelian, and metacyclic groups. *Discrete Mathematics* 312, 1 (2012), 94 – 99. Algebraic Graph Theory - A Volume Dedicated to Gert Sabidussi on the Occasion of His 80th Birthday.

Mirka Miller and Jozef Sirán. 2013. Moore Graphs and Beyond: A survey of the Degree/Diameter Problem (2nd Ed). *The Electronic Journal of Combinatorics* (5 2013).

Henry Martyn Mulder. 1982. Interval-regular graphs. *Discrete Mathematics* 41, 3 (1982), 253–269.

Li-Shiuan Peh and William J. Dally. 2001. A Delay Model and Speculative Architecture for Pipelined Routers. In *Proceedings of the 7th International Symposium on High-Performance Computer Architecture (HPCA '01)*. IEEE Computer Society, Washington, DC, USA, 255–.

Francisco Javier Ridruejo Perez and José Miguel-Alonso. 2005. INSEE: An Interconnection Network Simulation and Evaluation Environment. In *Euro-Par*. 1014–1023.

Bogdan Prisacari, German Rodriguez, Marina Garcia, Enrique Vallejo, Ramon Beivide, and Cyriel Minkenberg. 2014. Performance Implications of Remote-only Load Balancing Under Adversarial Traffic in Dragonflies. In *Proceedings of the 8th International Workshop on Interconnection Network Architecture: On-Chip, Multi-Chip (INA-OCMC '14)*. ACM, New York, NY, USA, Article 5, 4 pages.

Arjun Singh. 2005. *Load-Balancing Routing in Interconnection Network*. PhD Dissertation. Standford University. Advisor(s) William J. Dally.

Y. Tamir and G. L. Frazier. 1988. High-performance Multi-queue Buffers for VLSI Communications Switches. In *Proceedings of the 15th Annual International Symposium on Computer Architecture (ISCA '88)*. IEEE Computer Society Press, Los Alamitos, CA, USA, 343–354.

L. Valiant. 1982. A Scheme for Fast Parallel Communication. *SIAM J. Comput.* 11, 2 (1982), 350–361.

Ruisheng Wang, Lizhong Chen, and Timothy Mark Pinkston. 2013. Bubble Coloring: Avoiding Routing- and Protocol-induced Deadlocks with Minimal Virtual Channel Requirement. In *Proceedings of the 27th International ACM Conference on Supercomputing (ICS '13)*. New York, 193–202.

Received May 2014; revised August 2014; accepted September 2014