RETROSPECTIVE: Technology-Driven, Highly-Scalable Dragonfly Topology

John Kim[1] William J. Dally[2] Steve Scott[3] Dennis Abts[2]
[1]KAIST [2]NVIDIA [3]Microsoft

I. Dragonfly Overview

This paper was published at ISCA 2008 and proposed the Dragonfly topology for large-scale systems. With exponentially increasing pin bandwidth, it was demonstrated during the early 2000s that bandwidth can be better exploited with *high-radix* routers where the bandwidth is partitioned among a larger number of ports.[O15] [1] Given the availability of high-radix routers, previously common topologies (e.g., 2D and 3D tori) were not necessarily suitable while other topologies (e.g., fat tree or folded-Clos) resulted in high cost. An alternative high-radix topology was proposed in ISCA 2007 (Flattened Butterfly (FBFLY) [O14]) to exploit high-radix routers, but had some limitations. FBFLY scalability was limited by the switch radix, and to scale the topology, higher dimensional FBFLYs were needed, which increased the network diameter. In addition, most channels in the topology were effectively longer "global" channels, resulting in high network cost.

To overcome the limitations of FBFLY, the Dragonfly topology was proposed in which a collection of routers was used to create a "group" or a *virtual* high-radix router with very high effective radix. The higher radix enabled better scalability while reducing the number of global channels. The Dragonfly topology is effectively a hierarchical topology that consists of an intra-group topology and an inter-group topology; however, a key difference compared to traditional hierarchical topologies is that intra- and inter-group topologies are not necessarily interconnected hierarchically but are connected in parallel – e.g., some ports of the routers are used for the intra-group topology while other ports are used for the inter-group topology. The Dragonfly topology was *technology-driven*, as it exploited (cheap) electrical signaling for local connectivity within the group and minimized the necessary routing over (expensive) optical cables used for global channels between the groups. Dragonfly was also *highly scalable*, as with radix-64 switches and using a 1D FBFLY for both the intra- and inter-group topologies, a Dragonfly can scale to over a quarter million endpoints with a network diameter of only three switch-to-switch hops.

The Dragonfly topology not only reduces the network diameter and cost but also provides high path diversity that includes both minimal and non-minimal routes. Thus, a critical component of the Dragonfly topology is a load-balancing routing algorithm that exploits that path diversity. A key enabling technology for the high-radix Dragonfly topology is global adaptive routing [O29] where both minimal and non-minimal routes are load-balanced based on local congestion information and the minimal (and non-minimal) hop count. Non-minimal routes are selected based on Valiant's routing where a random intermediate group (and random global channel) is chosen. Thus, routing consists of multiple stages, including both intra-group routing and inter-group routing as well as routing within the intermediate group for non-minimal routing.

[1] [O..] refers to the bibliography in the original paper.

II. Dragonfly Impact

The Dragonfly topology work has had a significant impact on both industry and academia. The Dragonfly topology was used in both the Cray XC and Cray EX supercomputers. The Cray XC system employed a radix-48 switch with a 2D FBFLY topology within the group and 1D FBFLY between groups [3]. The Cray EX systems use the recent Cray/HPE Slingshot Ethernet interconnect. Slingshot's radix-64 switches enable the Cray EX systems to use a 1D FBFLY for both intra- and inter-group topologies, while still achieving very high scale. The Cray EX system was selected for all three initial US Exascale systems, including the Frontier Supercomputer at Oak Ridge National Laboratory (the fastest supercomputer on the Top500 as of June 2023).

The scale-out accelerator system from Groq [1] also leveraged the Dragonfly topology where a collection of accelerators (or tensor streaming processors) were used as a group to scale the system. However, instead of a hardware-based topology, a *software*-managed Dragonfly system was proposed where the same Dragonfly topology was leveraged but "routing" was no longer necessary as data movements were "scheduled."

The original Dragonfly proposal did not restrict the type of topology for either the intra- or the inter-group topology, but an exemplary fully-connected topology was used to minimize overall network diameter. As a result, the Dragonfly can easily be modified with different inter-group and/or intra-group topologies. A variation of the Dragonfly, referred to as the Dragonfly+ [9], was proposed by Mellanox wherein a fat-tree was used within each group to provide higher intra-group bisection bandwidth and scaling. This has been deployed in various Infiniband-based systems. That same variation of the Dragonfly was proposed as Megafly [4] by Intel. The low-latency Dragonfly topology was also leveraged in the Aquila [10] experimental datacenter network from Google.

The goal of reducing network diameter continued with other follow-on work, including the Slimfly [2] topology, which achieves a theoretical diameter of two switch-to-switch hops. The impact of hierarchical, technology-driven topology is also not limited to large-scale systems but can be applied to on-chip networks. Firefly [7] is one example where an on-chip network was proposed in which local communication was done electrically while nanophotonics was used for global on-chip communication.

III. What went right & What could be improved

The hierarchical organization proposed for Dragonfly was based on the cost difference between electrical and optical channels based on an early/mid-2000s cost model. While signaling technology has continued to evolve over the past 15 years, the cost difference still exists today (i.e., higher cost of optical cables per unit bandwidth) and the impact of technology is still a driving factor for large-scale system designs. In addition, while not explicitly mentioned in the original paper, the packaging hierarchy of systems (e.g., boards, chassis, racks, etc.) lends itself well to the Dragonfly organization. For example, the packaging locality of a single chassis or a single rack can exploit electrical channels to form a group while the inter-chassis or inter-rack communication can employ optical channels. Real systems that have implemented Dragonflies also exploit such packaging locality – e.g., a chassis was used as one dimension within a group in the Cray XC system and multiple chassis were interconnected across two cabinets electrically to form a 2D FBFLY group.

The main benefit of the Dragonfly is not only in lower network diameter but is its improvement in performance per unit cost relative to alternative topologies. Compared to the fat-tree (folded-Clos) topology, which has been commonly used since the early 2000s, the Dragonfly can provide significantly improved performance per cost by reducing the number of switches and halving the number of required optical links. A key challenge in the Dragonfly topology is the global adaptive routing that is necessary to load-balance the global channels. The original work identified some of the unique challenges of global adaptive routing, including how *indirect* adaptive routing is necessary as global congestion information is not readily available at the source router where the routing decision is made. This led to the development of progressive adaptive routing [6] in the follow-on work and influenced the routing algorithm implemented in the Cray systems.

One limitation of the original paper was how the adaptive decision was made. In the original paper, an adversarial traffic pattern for minimal routing consisted of a traffic pattern where nodes from a group (G_i) send all of their traffic to nodes in a neighboring group (G_{i+1}). Non-minimal routing where a random intermediate group is selected (and effectively selecting a random global channel) was sufficient to load-balance the topology for this traffic pattern. However, while this decision of randomly selecting an intermediate group load balances the global channel, it can lead to bottlenecks within intermediate groups for other adversarial traffic patterns. For example, when traffic is between two groups (G_i and G_j) that are not necessarily neighboring groups, bottlenecks can actually occur *within* the intermediate group, and overall network throughput is limited by the local channels in the intermediate group. The impact of alternative adversarial traffic patterns was observed by Garcia et al. [5] as well as authors of the original paper [11]. The routing limitation was addressed by randomizing the intermediate *node* and not necessarily the intermediate global channel to achieve the full benefits of Valiant's load-balancing.

IV. Dragonfly in 2023

The Dragonfly topology exploits the characteristics of modern interconnects (electrical and optical) and the advent of global adaptive routing to give the most cost-effective interconnect for large systems and can be found in many large high-performance computing systems, including the recent Frontier Exascale system. In short, the key benefit of the Dragonfly is its improvement in performance per unit cost compared to alternative topologies through direct connectivity. Recent data centers from Google (Jupiter network [8]) adopt similar principles as the Dragonfly as it employs a hybrid (hierarchical) approach with optical circuit switching that provides direct connectivity between the aggregation blocks. The benefits of the Dragonfly topology could be significantly reduced in the future if continued increases in signaling rates cause a transition to all-optical communication, minimizing the differences in cost between local and global network links.

References

[1] Dennis Abts et al. A software-defined tensor streaming multiprocessor for large-scale machine learning. In *Proceedings of ISCA*, 2022.

[2] Maciej Besta and Torsten Hoefler. Slim fly: A cost effective low-diameter network topology. In *Proceedings of SC*, 2014.

[3] Greg Faanes et al. Cray cascade: A scalable hpc system based on a dragonfly network. In *Proceedings of SC*, 2012.

[4] Mario Flajslik et al. Megafly: A topology for exascale systems. In *Proceedings of ISC*, 2018.

[5] Marina García et al. On-the-fly adaptive routing in high-radix hierarchical networks. In *Proceddings of ICPP*, 2012.

[6] Nan Jiang et al. Indirect adaptive routing on large scale interconnection networks. In *Proceedings of ISCA*, 2009.

[7] Yan Pan et al. Firefly: illuminating future network-on-chip with nanophotonics. In *Proceedings of ISCA*, 2009.

[8] Leon Poutievski et al. Jupiter evolving: Transforming google's datacenter network via optical circuit switches and software-defined networking. In *Proceedings of ACM SIGCOMM*, 2022.

[9] Alexander Shpiner et al. Dragonfly+: Low cost topology for scaling datacenters. In *IEEE HiPINEB Workshop*, 2017.

[10] Bingchuan Tian et al. Aquila: A practically usable verification system for production-scale programmable data planes. In *Proceedings of ACM SIGCOMM*, 2021.

[11] Jongmin Won et al. Overcoming far-end congestion in large-scale networks. In *Proceedings of HPCA*, 2015.