IEEE Copyright Notice

Accepted to be Published in: IEEE Transactions on Parallel and Distributed Systems (TPDS)

High-Performance Routing with Multipathing and Path Diversity in Supercomputers and Data Centers

Maciej Besta[1], Jens Domke[2], Marcel Schneider[1], Marek Konieczny[3],
Salvatore Di Girolamo[1], Timo Schneider[1], Ankit Singla[1], Torsten Hoefler[1]

[1]Department of Computer Science, ETH Zurich; [2]RIKEN Center for Computational Science (R-CCS)
[3]Faculty of Computer Science, Electronics and Telecommunications; AGH-UST

Abstract—The recent line of research into topology design focuses on lowering network diameter. Many low-diameter topologies such as Slim Fly or Jellyfish that substantially reduce cost, power consumption, and latency have been proposed. A key challenge in realizing the benefits of these topologies is *routing*. On one hand, these networks provide shorter path lengths than established topologies such as Clos or torus, leading to performance improvements. On the other hand, the number of shortest paths between each pair of endpoints is much smaller than in Clos, but there is a large number of non-minimal paths between router pairs. This hampers or even makes it impossible to use established multipath routing schemes such as ECMP. In this work, to facilitate high-performance routing in modern networks, we analyze existing routing protocols and architectures, focusing on how well they exploit the diversity of minimal and non-minimal paths. We first develop a taxonomy of different forms of support for multipathing and overall path diversity. Then, we analyze how existing routing schemes support this diversity. Among others, we consider multipathing with both shortest and non-shortest paths, support for disjoint paths, or enabling adaptivity. To address the ongoing convergence of HPC and "Big Data" domains, we consider routing protocols developed for both HPC systems and for data centers as well as general clusters. Thus, we cover architectures and protocols based on Ethernet, InfiniBand, and other HPC networks such as Myrinet. Our review will foster developing future high-performance multipathing routing protocols in supercomputers and data centers.

1 Introduction and Motivation

Fat tree [105] and related networks such as Clos [43] are the most commonly deployed topologies in data centers and supercomputers today, dominating the landscape of Ethernet clusters [120, 78, 159]. However, many low-diameter topologies such as Slim Fly or Jellyfish that substantially reduce cost, power consumption, and latency have been proposed. These networks improve the cost-performance tradeoff compared to fat trees. For instance, Slim Fly is $\approx 2\times$ more cost- and power-efficient at scale than fat trees, simultaneously delivering $\approx 25\%$ lower latency [19].

A key challenge in realizing the benefits of these topologies is routing. On one hand, due to their lower diameters, these networks provide shorter path lengths than fat trees and other traditional topologies such as torus. However, as illustrated by our recent research efforts [26], *the number of shortest paths between each pair of endpoints is much smaller than in fat trees*. Selected results are illustrated in Figure 1. In this figure, we compare established three-level fat trees (FT3) with representative modern low-diameter networks: Slim Fly (SF) [19, 30] (a variant with diameter 2), Dragonfly (DF) [101] (the "balanced" variant with diameter 3), Jellyfish (JF) [147] (with diameter 3), Xpander (XP) [159] (with diameter ≤ 3), and HyperX (Hamming graph) (HX) [3] that generalizes Flattened Butterflies (FBF) [100] with diameter 3. As observed [26], *"in DF and SF, most routers are connected with* one *minimal path. In XP, more than 30% of routers are connected with* one *minimal path."* In the corresponding JF networks (i.e., random Jellyfish networks constructed using the same number of identical routers as in the corresponding non-random topology), *"the results are more leveled out, but pairs of routers with one shortest part in-between still form large fractions. FT3 and HX show the highest diversity."* We conclude that in all the considered low-diameter topologies, ***shortest paths fall short:*** at least a large fraction of router pairs are connected with ***only one*** shortest path.

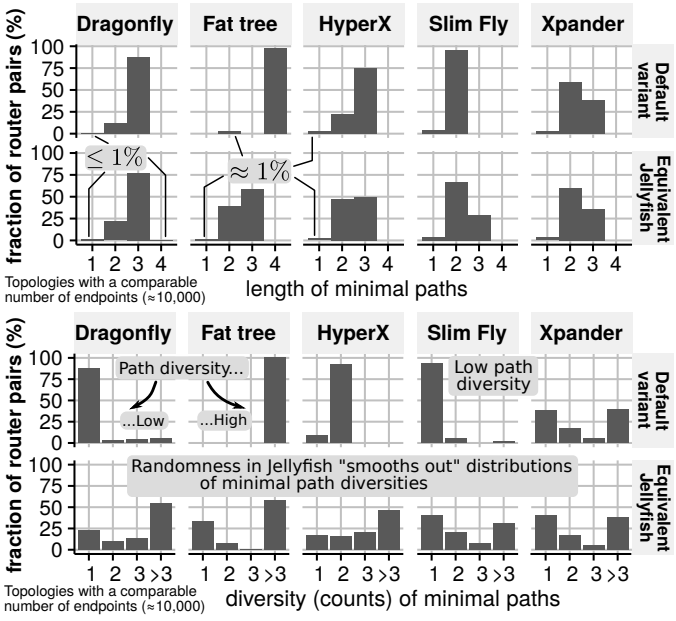

Fig. 1: Distributions of lengths and counts of shortest paths in low-diameter topologies and in fat trees. When analyzing counts of minimal paths between a router pair, we consider disjoint paths (no shared links). An equivalent Jellyfish network is constructed using the same number of identical routers as in the corresponding non-random topology (a plot taken from our past work [26]).

Simultaneously, **these low-diameter topologies offer high diversity of *non-minimal paths*** [26]. They provide at least three disjoint "almost"-minimal paths (i.e., paths that

are one hop longer than their corresponding shortest paths) per router pair (for the majority of pairs). For example, in Slim Fly (that has the diameter of 2), 99% of router pairs are connected with multiple non-minimal paths of length 3 [26].

The above properties of low-diameter networks place unprecedented design challenges for performance-conscious routing protocols. First, as shortest paths fall short, one must resort to non-minimal routing, which is usually more complex than the minimal one. Moreover, as topologies lower their diameter, their link count is also reduced. Thus, even if they do indeed offer more than one non-minimal path between pairs of routers, the corresponding routing protocol must carefully use these paths in order not to congest the network (i.e., the path diversity is still a scarce resource demanding careful examination and use). Third, a shortage of shortest paths means that one cannot use established multipath routing[1] schemes such as Equal-Cost Multi-Path (ECMP) [82], which usually assume that *different paths between communicating entities are minimal and have equal lengths*. Restricting traffic to these paths does not utilize the path diversity of low-diameter networks.

In this work, to facilitate overcoming these challenges and to propel designing high-performance routing for modern interconnects, we develop a taxonomy of different forms of support for path diversity by a routing design. These forms of support include (1) enabling multipathing using both (2) shortest and (3) non-shortest paths, (4) explicit consideration of disjoint paths, (5) support for adaptive load balancing across these paths, and (6) genericness (i.e., being applicable to different topologies). We also discuss additional aspects, for example whether a given design uses multipathing to enhance its *resilience*, *performance*, or both.

Then, we use this taxonomy to categorize and analyze a wide selection of existing routing designs. Here, we consider two fundamental classes of routing designs: simple routing *building blocks* (e.g., ECMP [82] or Network Address Aliasing (NAA)) and routing *architectures* (e.g., PortLand [120] or PARX [54]). While analyzing respective routing architectures, we include and investigate the architectural and technological details of these designs, for example whether a given scheme is based on the simple Ethernet architecture, the full TCP/IP stack, the InfiniBand (IB) stack, or other HPC designs. This enables network architects and protocol designers to gain insights into supporting path diversity in the presence of different technological constraints.

We consider protocols and architectures that originated in both the HPC and data center as well as general networking communities. This is because all these environments are important in today's large-scale networking landscape. While the most powerful Top500 systems use vendor-specific or InfiniBand (IB) interconnects, more than half of the Top500 (e.g., in the June 2019 or in the November 2019 issues) machines [55] are based on Ethernet, see Figure 2. We observe similar numbers for the Green500 list. The importance of Ethernet is increased by the *"convergence of HPC and Big Data"*, with cloud providers and data center operators aggressively aiming for high-bandwidth and low-latency fabrics [159, 78, 162]. Another example is Mellanox, with its Ethernet sales being higher than those for InfiniBand in recent years [132]. At the same time, IB's sales have been growing by 27% year-over-year [170]. Thus, our analysis can facilitate developing multipath routing in both IB-based supercomputers but also in a broad landscape of cloud computing infrastructure such as data centers.

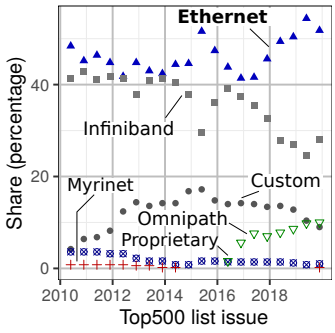

Fig. 2: The share of different interconnect technologies in the Top500 systems (a plot taken from our past work [26]).

Complementary Analyses There exist surveys on multipathing [134, 11, 175, 7, 158, 106, 146]. Yet, none focuses on multipathing and path diversity offered by routing in data centers or supercomputers. For example, Lee and Choi describe multipathing in the general Internet and telecommunication networks [103]. Li et al. [106] also focus on the general Internet, covering aspects of multipath transmission related to all TCP/IP stack layers. Singh et al. [146] cover only a few multipath routing schemes used in data centers, focusing on a broad Internet setting. Moreover, some works are dedicated to performance evaluations of a few schemes for multipathing [2, 84]. Next, different works are dedicated to multipathing in sensor networks [7, 11, 134, 175]. Finally, there are analyses of other aspects of data center networking, for example energy efficiency [145, 16], optical interconnects [95], network virtualization [90, 15], overall routing [40], general data center networking with focus on traffic control in TCP [122], low-latency data centers [107], the TCP incast problem [136], bandwidth allocation [41], transport control [172], general data center networking for clouds [165], congestion management [92], reconfigurable data center networks [62], and transport protocols [152, 133]. We complement all these works, focusing solely on *multipath routing in supercomputers, data centers, and small clusters.* As opposed to other works with broad focus, *we specifically target the performance aspects of multipathing and path diversity*. Our survey is the first to deliver a taxonomy of the path diversity features of routing schemes, to categorize existing routing protocols based on this taxonomy, and to consider both traditional TCP/IP and Ethernet designs, but also protocols and concepts traditionally associated with HPC, for example multipathing in InfiniBand [130, 67].

2 Fundamental Notions

We first outline fundamental notions: *network topologies*, *network stacks*, and associated *routing concepts* and *designs*.

While we do not conduct any theoretical investigation, we state – for clarity – a network model used implicitly

[1] **Multipath** routing indicates a routing protocol that uses more than one path in the network, for at least one pair of communicating endpoints. We consider multipathing both within a single flow/message (e.g., as in spraying single packets across multiple paths, cf. § 4.6), and multipath across flows/messages (e.g., as in standard ECMP, where different flows follow different paths § 4.4). **Path diversity** indicates whether a given network topology offers multiple paths between different routers (i.e., has potential for speedups from multipath routing).

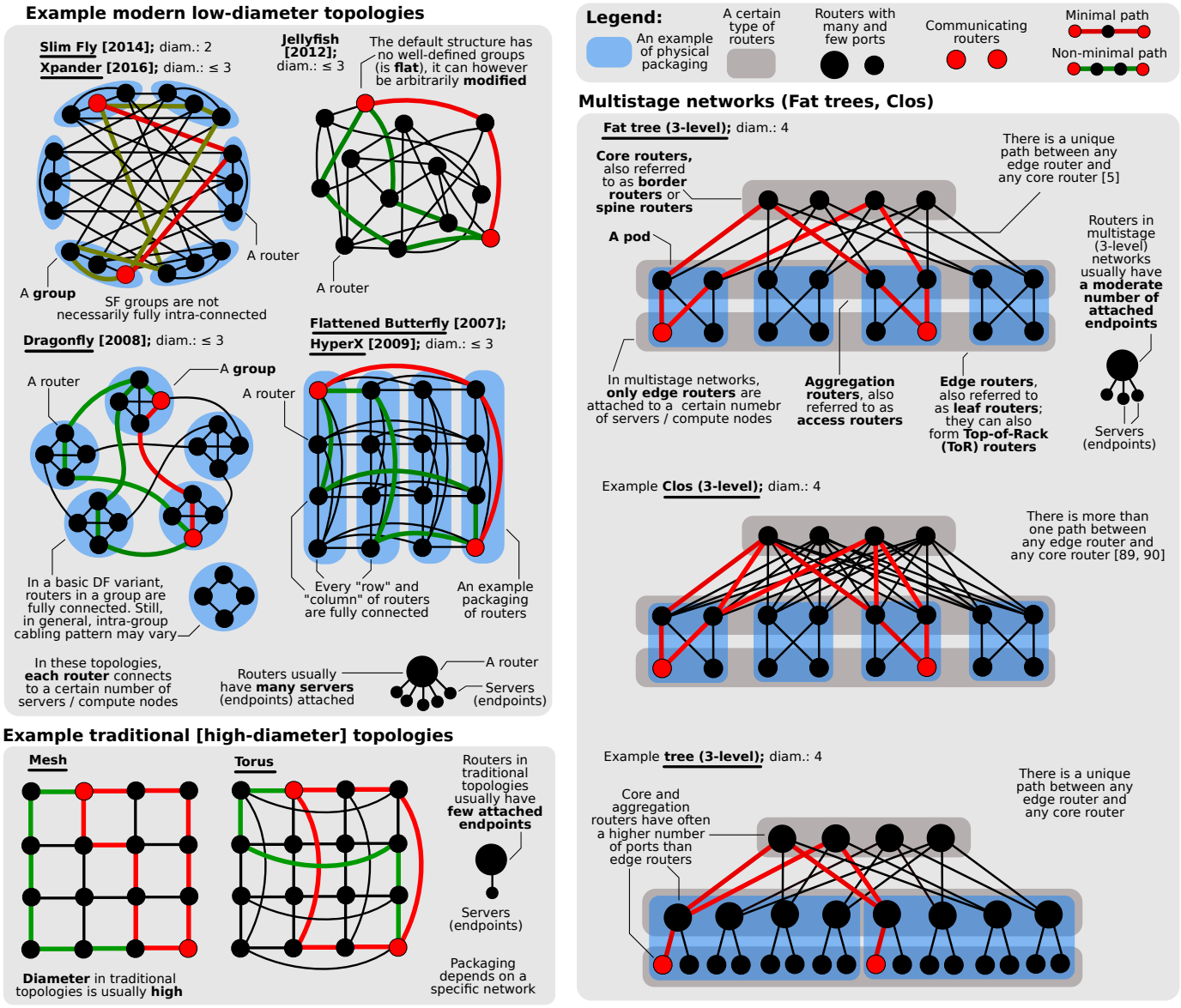

Fig. 3: Illustration of network topologies related to the routing protocols and schemes considered in this work. Red color indicates an example shortest path between routers. Green color indicates example alternative non-minimal paths. Blue color illustrates grouping of routers.

in this work. We model an interconnection network as an undirected graph $G = (V, E)$; V and E are sets of routers, also referred to as nodes ($|V| = N_r$), and full-duplex inter-router physical links. Endpoints (also referred to as servers or compute nodes) are *not* modeled explicitly.

2.1 Network Topologies

We consider routing in different network topologies. The most important associated topologies are in Figure 3. We only briefly describe their structure that is used by routing architectures to enable multipathing (a detailed analysis of respective topologies in terms of their path diversity is available elsewhere [26]). In most networks, routers form *groups* that are intra-connected with the same pattern of cables. We indicate such groups with the blue color.

Many routing designs are related to **fat trees (FT)** [105] and **Clos (CL)** [43]. In these networks (broadly referred to as "**multistage topologies (MS)**"), a certain fraction of routers is attached to endpoints while the remaining routers are only dedicated to forwarding traffic. A common realization of these networks consists of three stages (layers) of routers: *edge* (*leaf*) routers, *aggregation* (*access*) routers, and *core* (*spine*, *border*) routers. Edge and aggregation routers are additionally grouped into *pods*, to facilitate physical layout (cf. Fig. 3). Only edge routers connect to endpoints. Aggregation and core routers only forward traffic; they enable multipathing. The exact form of multipathing depends on the topology variant. Consider a pair of communicating edge routers (located in different pods/groups). In fat trees, multipathing is enabled by selecting *different core routers* and *different aggregation routers* to forward traffic between the same communicating pair of edge routers. Importantly, after fixing the core router, *there is a unique path between the communicating edge routers*. In Clos, in addition to such multipathing enabled by selecting different core routers, *one can also use different paths between a specific edge and core router*. Finally, simple trees are similar to fat trees in that fixing different core routers enables multipathing; still, *one cannot multipath by using different aggregation routers*.

The most important **modern low-diameter networks** are Slim Fly (SF) [19], Dragonfly (DF) [101], Jellyfish

(JF) [147], Xpander (XP) [159], and HyperX (Hamming graph) (HX) [3]. Other proposed topologies in this family include Flexfly [166], Galaxyfly [104], Megafly [60], projective topologies [37], HHS [14], and others [135, 98, 124]. All these networks have different structure and thus different potential for multipathing [26]; in Figure 3, we illustrate example paths between a pair of routers. Importantly, in most of these networks, *unlike in fat trees, different paths between two endpoints usually have different lengths* [26].

Finally, many routing designs can be used with any topology, including **traditional ones** such as meshes.

2.2 Routing Concepts and Related

We often refer to three interrelated sub-problems for routing: Ⓟ **Path selection**, Ⓡ **Routing** itself, and Ⓛ **Load balancing**. Path selection Ⓟ determines *which paths* can be used for sending a given packet. Routing itself Ⓡ answers a question on *how* the packet finds a way to its destination. Load balancing Ⓛ determines *which path (out of identified alternatives)* should be used for sending a packet *to maximize performance* and *minimize congestion*.

2.3 Routing Schemes

We consider routing schemes (designs) that can be loosely grouped into specific **protocols** (e.g., OSPF [116]), **architectures** (e.g., PortLand [120]), and general **strategies and techniques** (e.g., ECMP [82] or spanning trees [126]). Overall, a protocol or a strategy often addresses a *specific* networking problem, rarely more than one. Contrarily, a routing architecture usually delivers a *complete routing solution* and it often addresses more than one, and often all, of the above-described problems. All these designs are almost always developed in the context of a specific **network stack**, also referred to as **network architecture**, that we describe next.

2.4 Network Stacks

We focus on data centers and high-performance systems. Thus, we target Ethernet & TCP/IP, and traditional HPC networks (InfiniBand, Myrinet, OmniPath, and others).

2.4.1 Ethernet & TCP/IP

In the TCP/IP protocol stack, two layers of addressing are used. On **Layer 2 (L2)**, *Ethernet* (*MAC*) addresses are used to uniquely identify endpoints, while on **Layer 3 (L3)**, *IP* addresses are assigned to endpoints. Historically, the Ethernet layer is not supposed to be routable: MAC addresses are only used within a bus-like topology where no routing is required. In contrast, the IP layer is designed to be routable, with a hierarchical structure that allows scalable routing over a worldwide network (the *Internet*). More recently, vendors started to provide routing abilities on the Ethernet layer for pragmatic reasons: since the Ethernet layer is effectively transparent to the software running on the endpoints, such solutions are easy to deploy. Additionally, the Ethernet interconnect of a cluster can usually be considered homogeneous, while the IP layer is used to route *between* networks and needs to be highly interoperable.

Since Ethernet was not designed to be routable, there are several restrictions on routing protocols for Ethernet: First, the network *cannot* modify any fields in the packets (control-data plane separation is key in self-configuring Ethernet devices). There is no mechanism like the *TTL* field in the IP header that allows the network to detect cyclic routing. Second, Ethernet devices come with pre-configured, effectively random addresses. This implies that there is no structure in the addresses that would allow for a scalable routing implementation: Each switch needs to keep a lookup table with entries for each endpoint in the network. Third, since the network is expected to self-configure, Ethernet routing schemes must be robust to the addition and removal of links. These restrictions shape many routing schemes for Ethernet: Spanning trees are commonly used to guarantee loop-freedom under any circumstances, and more advanced schemes often rely on wrapping Ethernet frames into a format more suitable for routing at the edge switches [120].

Another intricacy of the TCP/IP stack is that flow control is only implemented in **Layer 4 (L4)**, the transport layer. This means that the network is not supposed to be aware of and responsible for load balancing and resource sharing; rather, it should deliver packets to the destination on a best-effort basis. In practice, most advanced routing schemes violate this separation and are *aware* of TCP flows, even though flow control is still left to the endpoint software [78]. Many practical problems are caused by the interaction of TCP flow control with decisions in the routing layer, and such problems are often discussed together with routing schemes, even though they are completely independent of the network topology (e.g., the *TCP incast problem*).

Traditional Ethernet is *lossy*: when packet buffers are full, packets are dropped. Priority Flow Control (PFC) [47] addresses this by allowing a switch to notify another (upstream) switch *with special "pause" frames* to stop sending frames until further notice, if the buffer occupancy in the first switch is above a certain threshold. Another extension of Ethernet towards technologies traditionally associated with HPC is the incorporation of Remote Direct Memory Access (RDMA) using the RDMA over Converged Ethernet (RoCE) [86] protocol, which enriches the Ethernet with the RDMA communication semantics.

2.4.2 InfiniBand

The InfiniBand (IB) architecture is a switched fabric design and is intended for high-performance and system area network (SAN) deployment scales. Up to 49,151 endpoints (physical or virtual), addressed by a 16 bit *local identifier* (LID), can be arranged in a so called **subnet**, while the remaining address space is reserved for multicast operations within a subnet. Similar to the modern datacenter Ethernet (L2) solutions, these IB subnets are routable to a limited extent with switches supporting unicast and multicast forwarding tables, flow control, and other features which do not require modification of in-flight packet headers. Theoretically, multiple subnets can be connected by IB routers — performing the address translation between the subnets — to create larger SANs (effectively **L3** domains), but this impedes performance due to the additionally required *global routing header* (GRH) and is rarely used in practice.

IB natively supports RDMA and atomic operations. The necessary (for high performance) lossless packet forwarding within IB subnets is realized through link-level, credit-based flow control [45]. Software-based and latency impeding solutions to achieve reliable transmissions, as for example in TCP, are therefore not required. While switches have the capability to drop deadlocked packets that reside for extended

time periods in their buffers, they cannot identify livelocks, such as looping unicast or multicast packets induced by cyclic routing. Hence, the correct and acyclic routing configuration is offloaded to a centralized controller, called *subnet manager*, which configures connected IB devices, calculates the forwarding tables with implemented topology-agnostic or topology-aware routing algorithms, and monitors the network for failures. Therefore, most routing algorithms either focus on *minimal path length* to guarantee loop-freedom, or are derivatives of the Up*/Down* routing protocol [110, 61] which can be viewed as a generalization of the spanning tree protocol of Ethernet networks. Besides this oblivious, *destination-based* routing approach, IB also supports *source-based* routing, but unfortunately only for a limited traffic class reserved for certain management packets.

The subnet manager can configure the InfiniBand network with a few flow control features, such as quality-of-service to prioritize traffic classes over others or congestion control mechanism to throttle ingest traffic. However, adhering to the correct service levels or actually throttling the packet generation is left to the discretion of the endpoints. Similarly, in sacrifice for lowest latency and highest bandwidth, IB switches have limited support for common capabilities found in Ethernet, for example VLANs, firewalling, or other security-relevant functionality. Consequently, some of these have been implemented in software at the endpoints on top of the IB transport protocol, e.g., TCP/IP via IPoIB, whenever the HPC community deemed it necessary.

2.4.3 Other HPC Network Designs

Cray's **Aries** [6] is a packet-switched interconnect designed for high performance and deployed on the Cray XC systems. Aries adopts a dragonfly topology, where nodes within groups are interconnected with a two-dimensional all-to-all structure (i.e., routers in one dragonfly group effectively form a flattened butterfly, cf. Figure 3). Being designed for high-performance systems, it allows nodes to communicate with RDMA operations (i.e., put, get, and atomic operations). The routing is destination-based, and the network addresses are tuples composed by a node identifier (18-bit, max 262,144 nodes), the memory domain handle (12-bit) that identifies a memory segment in the remote node, and an offset (40-bit) within this segment. The Aries switches employ wormhole routing [44] to minimize the per-switch required resources. Aries does not support VLANs or QoS mechanisms, and its stack design does not match that of Ethernet. Thus, we define the (software) layer at which the Aries routing operates as **proprietary**.

Slingshot [149] is the next-generation Cray network. It implements a DF topology with fully-connected groups. Slingshot can switch two types of traffic: RoCE (using **L3**) and proprietary. Being able to manage RoCE traffic, a Slingshot system can be interfaced directly to data centers, while the proprietary traffic (similar to Aries, i.e., RDMA-based and small-packet headers) can be generated from within the system, preserving high performance. Cray Slingshot supports VLANs, QoS, and endpoint congestion mitigation.

IBM's **PERCS** [12] is a two-level direct interconnection network designed to achieve high bisection bandwidth and avoid external switches. Groups of 32 compute nodes (made of four IBM POWER7 chips) are fully connected and organized in supernodes. Each supernode has 512 links connecting it to other supernodes. Depending on the system size (max 512 supernodes), each supernode pair can be connected with one or multiple links. PERCS supports RDMA, hardware-accelerated collective operations, direct-cache (L3) network access, and enables applications to switch between different routing modes. Similarly to Aries, PERCS routing operates on a *proprietary* stack.

We also summarize other HPC oriented proprietary interconnects. Some of them are no longer manufactured; we include them for the completeness of our discussion of path diversity. Myricom's **Myrinet** [35] is a local area *massively parallel processor* network, designed to connect thousands of small compute nodes. A more recent development, **Myrinet Express (MX)** [66], provides more functionalities in its network interface cards (NICs). **Open-MX** [70] is a communication layer that offers the MX API on top of the Ethernet hardware. Quadrics' **QsNet** [128, 129] integrates local memories of compute nodes into a single *global virtual address space*. Moreover, Intel introduced **OmniPath** [34], an architecture for a tight integration of CPU, memory, and storage units. Other HPC interconnects are Atos' Bull eXascale Interconnect (BXI) [48] and EXTOLL's interconnect [119]. Many of these architectures feature some form of programmable NICs [35, 129]. Finally, there exist routing protocols for specific low-diameter topologies, for example for SF [171] or DF [111]. However, they usually do not support multipathing or non-minimal routing.

2.5 Focus of This Work

In our investigation, *we focus on **routing***. Thus, in the Ethernet and TCP/IP landscape, we focus on designs associated with Layer 2 (L2, Data Link Layer) and Layer 3 (L3, Internet Layer), cf. § 2.4.1. As most of congestion control and load balancing are related to higher layers, we only describe such schemes whenever they are parts of the associated L2 or L3 designs. In the InfiniBand landscape, we focus on the subnet and L3 related schemes, cf. § 2.4.2.

3 Taxonomy of Routing Schemes

We first identify criteria for categorizing the considered routing designs. We focus on how well these designs utilize path diversity. These criteria are used in Tables 1–2. Specifically, we analyze whether a given scheme enables using (1) arbitrary **shortest** paths and (2) arbitrary **non-minimal** paths. Moreover, we consider whether a studied scheme enables (3) **multipathing** (between two hosts) and whether these paths can be (4) **disjoint**. Finally, we investigate (5) the support for **adaptive load balancing** across exposed paths between router pairs and (6) compatibility with an **arbitrary topology**. In addition, we also indicate the **location** of each routing scheme in the networking stack[2]. We also indicate whether a given multipathing scheme focuses on **performance** or **resilience** (i.e., to provide backup paths in the event of failures). Next, we identify whether supported paths come **with certain restrictions**, e.g., whether they are offered only within a spanning tree. Finally, we also broadly categorize the analyzed routing schemes into **basic**

[2]We consider protocols in both Data Link (L2) and Network (L3) layers. However, we abstract away hardware details and use a term "router" for both L2 switches and L3 routers, *unless describing a specific switching protocol (to avoid confusion).*

Routing Scheme (Name, Abbreviation, Reference)	Related concepts (§ 2.2)	Stack Layer (§ 2.4)	Features of schemes SP	NP	MP	DP	ALB	AT	Additional remarks and clarifications
General routing building blocks (classes of routing schemes)									
Simple Destination-based routing	R	L2, L3	👍	👍*	✖	✖	✖	👍	*Care must be taken not to cause cyclic dependencies
Simple Source-based routing (SR)	R	L2, L3	👍	👍	👍*	👍*	✖	👍	Source routing is difficult to deploy in practice, but it is more flexible than destination-based routing. *As endpoints know the physical topology, multipathing should be easier to realize than in destination routing.
Simple Minimal routing	P	L2, L3	👍	✖	✖	✖	✖	👍	Easy to deploy, numerous designs fall in this category
Specific routing building blocks (concrete protocols or concrete protocol families)									
Equal-Cost Multipathing (ECMP) [82]	R L	L3	👍	✖	👍	✖	✖	👍	In ECMP, all routing decisions are local to each switch.
Spanning Trees (ST) [126]	P	L2	👍*	👍*	✖	✖	✖	👍	*The ST protocol offers shortest paths but only within one spanning tree.
Packet Spraying (PR) [50]	L	L2, L3	👍	✖	👍	✖	✖	👍	One selects output ports with round-robin [50] or randomization [143].
Virtual LANs (VLANs)	P	L2	👍*	👍*	✖	✖	✖	👍	*VLANs by itself does not focus on multipathing, and it inherits spanning tree limitations, but it is a key part of multipathing architectures.
IP Routing Protocols	R	L2, L3	👍	✖	✖	✖	✖	👍	Examples are OSPF [116], IS-IS [123], EIGRP [125].
Location–Identification Separation (LIS)	R	L2, L3	👍*	👍*	✖*	✖*	✖	👍	*LIS by itself does not focus on multipathing and path diversity, but it may facilitate developing a multipathing architecture.
Valiant load balancing (VLB) [160]	R P L	L2, L3	✖	👍	✖	✖	✖	👍	—
UGAL [101]	R P L	L2, L3	👍	👍	👍*	✖	👍	👍	UGAL means Universal Globally-Adaptive Load balanced routing.
Network Address Aliasing (NAA)	L	L3, subnet	👍*	👍*	👍*	👍*	👍*	👍	NAA is based on IP aliasing in Ethernet networks [131] and virtual ports via LID mask control (LMC) in InfiniBand [87, Sec. 7.11.1]. *Depending on how a derived scheme implements it.
Multi-Railing	P	L2, L3, subn.	👍*	👍*	👍	👍*	👍*	👍	*Depending on how a derived scheme implements it.
Multi-Planes	P	L2, L3, subn.	👍*	👍*	👍	👍	👍*	👍	*Depending on how a derived scheme implements it.

TABLE 1: Comparison of simple routing building blocks (often used as parts of more complex routing schemes in Table 2). Rows are sorted chronologically. We focus on how well the compared schemes utilize path diversity. "Related concepts" indicates the associated routing concepts described in § 2.2. **"Stack Layer"** indicates the location of each routing scheme in the TCP/IP or InfiniBand stack (cf. § 2.4). **SP**, **NP**, **MP**, **DP**, **ALB**, and **AT** illustrate whether a given routing scheme supports various aspects of path diversity. Specifically: **SP**: A given scheme enables using arbitrary **shortest** paths. **NP**: A given scheme enables using arbitrary **non-minimal** paths. **MP**: A given scheme enables **multipathing** (between two hosts). **DP**: A given scheme considers **disjoint** paths. **ALB**: A given scheme offers **adaptive load balancing**. **AT**: A given scheme works with an **arbitrary topology**. 👍: A given scheme does offer a given feature. 👍: A given scheme offers a given feature in a limited way. ✖: A given scheme does not offer a given feature. *Explanations in remarks.

and **complex** ones. The former are usually specific protocols or classes of protocols, used as building blocks of the latter.

4 Simple Routing Building Blocks

We now present simple routing schemes, summarized in Table 1, that are usually used as building blocks for more complex routing designs. For each described scheme, we indicate *what aspects of routing (as described in § 2.2) this scheme focuses on:* P *path selection,* R *routing itself,* L *or load balancing*. We consider both *general* classes of schemes (e.g., overall destination-based routing) and also *specific* protocols (e.g., Valiant routing [160]).

Note that, in addition to schemes focusing on multipathing, we also describe designs that do *not* explicitly enable it. This is because these designs are often used as key building blocks of architectures that provide multipathing. An example is a simple spanning tree mechanism, that – on its own – does not enable any form of multipathing, but is a basis of numerous designs that enable it [8, 154].

4.1 Destination-Based Routing Protocols

The most common approach to R *routing* are destination-based routing schemes. Each router holds a *routing table* that maps any destination address to a next-hop output port. No information apart from the destination address is used, and the packet does not need to be modified in transit. In this setup, it is important to differentiate the physical network topology (typically modeled as an undirected graph, since all practically used network technologies use full-duplex links, cf. § 2.1) from the *routing graph*, which is naturally *directed* in destination-based schemes. In the routing graph, there is an edge from node a to node b iff there is a routing table entry at a indicating b as the next hop destination.

Simple destination-based routing protocols can only provide a single path between any source and destination, *but this path can be non-minimal*. For non-minimal paths, special care must be taken to not cause cyclic routing: this can happen when the routing tables of different routers are not consistent, cf. *property preserving network updates* [52]. In a configuration without routing cycles, the routing graph for a fixed destination node is a tree rooted in the destination.

4.2 Source Routing (SR)

Another R *routing* scheme is *source routing* (SR). Here, the route from source to destination is computed *at the source*, and then attached to the packet before it is injected into the network. Each switch then reads (and possibly removes) the next hop entry from the route, and forwards the packet there. Compared to destination based routing, this allows for far more flexible path selection [93]. Yet, now the endpoints need to be aware of the network topology to make viable routing choices.

Source routing is rarely deployed in practice. Still, it could enable superior routing decisions (compared to destination based routing) in terms of utilizing path diversity, as endpoints know the physical topology. There are recent proposals on how to deploy source routing in practice, for example with the help of OpenFlow [93], or with packet encapsulation (IP-in-IP or MAC-in-MAC) [72, 85, 73]. Source routing can also be achieved to some degree with Multiprotocol Label Switching (MPLS) [138], a technique in which a router forwards packets based on *path labels* instead of *network addresses* (i.e., the MPLS label assigned to a packet can represent a path to be chosen [138, 169]).

4.3 Minimal Routing Protocols

A common approach to P *path selection* is to only use *minimal* paths: Paths that are no longer than the shortest path between their endpoints. Minimal paths are preferrable for routing because they minimize network resources consumed for a given volume of traffic, which is crucial to achieve good performance at high load.

An additional advantage of minimal paths is that they guarantee loop-free routing in destination-based routing

schemes. For a known, fixed topology, the routing tables can be configured to always send packets along shortest paths. Since every hop along any shortest path will decrease the shortest-path distance to the destination by one, the packet always reaches its destination in a finite number of steps.

Basic minimal routing does not consider multipathing. However, schemes such as Equal-Cost Multipathing (ECMP) extend minimal routing to multipathing (§ 4.4).

4.4 Equal-Cost Multipathing (ECMP)

Equal-Cost Multipathing [82] routing is an extension of simple destination-based (R) routing that specifically exploits the properties of minimal paths. Instead of having only one entry per destination in the routing tables, multiple next-hop options are stored. In practice, ECMP is used with minimal paths, because using non-minimal ones may lead to routing loops. Now, any router can make an arbitrary choice among these next-hop options. The resulting routing will still be loop-free and only use minimal paths.

ECMP allows to use a greater variety of paths compared to simple destination-based routing. Since now there may be multiple possible paths between any pair of nodes, a mechanism for (L) *load balancing* is needed. Typically, ECMP is used with a simple, oblivious scheme similar to packet spraying (§ 4.6), but on a per-flow level to prevent packet reordering [42]: each switch chooses a pseudo-random next hop port among the shortest paths based on a hash computed from the flow parameters, aiming to obtain an even distribution of load over all minimal paths (some variations of such simple per-flow scheme were proposed, for example Table-based Hashing [153] or FastSwitching [176]). Yet, random assignments *do not* imply uniform load balancing in general, and more advanced schemes such as *Weighted Cost Multipathing* (WCMP) [173, 174] aim to improve this. In addition, ECMP natively does not support adaptive load balancing. This is addressed by many network architectures described in Section 5 and by direct extensions of ECMP, such as *Congestion-Triggered Multipathing* (CTMP) [150] or Table-based Hashing with Reassignments (THR) [42].

4.5 Spanning Trees (ST)

Another approach to (P) *path selection* is to restrict the topology to a spanning tree. Then, the routing graph becomes a tree of bi-directional edges which guarantees the absence of cycles as long as no router forwards packets back on the link that the packet arrived on. This can be easily enforced by each router without any global coordination. Spanning tree based solutions are popular for auto-configuring protocols on changing topologies. However, simple spanning tree-based routing can leave some links completely unused if the network topology is not a tree. Moreover, shortest paths within a spanning tree are not necessarily shortest when considering the whole topology. Spanning tree based solutions are an alternative to minimal routing to ensure loop-free routing in destination-based routing systems. They allow for non-minimal paths at the cost of not using network resources efficiently and have been used as a building block in schemes like SPAIN [117]. A single spanning tree does not enable multipathing between two endpoints. However, as we discuss in Section 5, different network architectures use spanning trees to enable multipathing [154].

4.6 Packet Spraying

A fundamental concept for (L) *load balancing* is per-packet load balancing. In the basic variant, *random packet spraying* [50], each packet is sent over a randomly chosen path selected from a (static) set of possible paths. The key difference from ECMP is that modern ECMP spreads flows, not packets. Typically, packet spraying is applied to multistage networks, where many equal length paths are available and a random path among these can be chosen by selecting a random upstream port at each router. Thus, simple packet spraying natively considers, enables, and uses multipathing.

In TCP/IP architectures, per-packet load balancing is often not considered due to the negative effects of packet reordering on TCP flow control; but these effects can still be reduced in various ways [50, 78], for example by spraying not single packets but series of packets, such as flowlets [162] or flowcells [79]. Moreover, basic random packet spraying is an *oblivious* load balancing method, as it does not use any information about network congestion. However, in some topologies, for example in fat trees, it can still guarantee optimal performance as long as it is used for all flows. Unfortunately, this is no longer true as soon as the topology looses its symmetry due to link failures [174].

4.7 Virtual LANs (VLANs)

Virtual LANs (VLANs) [108] were originally used for isolating Ethernet broadcast domains. They have recently been used to implement multipathing. Specifically, once a VLAN is assigned to a given spanning tree, changing the VLAN tag in a frame results in sending this frame over a different path, associated with a different spanning tree (imposed on the same physical topology). Thus, VLANs – in the context of multipathing – primarily address path selection (P).

4.8 Simple IP Routing

We explicitly distinguish a class of established IP routing protocols (R), such as OSPF [116] or IS-IS [123]. They are often used as parts of network architectures. Despite being generic (i.e., they can be used with any topology), they do not natively support multipathing.

4.9 Location–Identification Separation (LIS)

In Location–Identification Separation (LIS), used in some architectures, a routing scheme (R) *separates the physical location of a given endpoint from its logical identifier*. In this approach, the logical identifier of a given endpoint (e.g., its IP address used in an application) does not necessarily indicate the physical location of this endpoint in the network. A mapping between identifiers and addresses can be stored in a distributed hashtable (DHT) maintained by switches [99] or hosts, or it can be provided by a directory service (e.g., using DNS) [73]. This approach enables more scalable routing [58]. Importantly, it may facilitate multipathing by – for example – maintaining multiple *virtual* topologies defined by different mappings in DHTs [91].

4.10 Valiant Load Balancing (VLB)

To facilitate non-minimal (R) *routing*, additional information apart from the destination address can be incorporated into a destination-based routing protocol. An established and common approach is *Valiant routing* [160], where this additional information is an arbitrary intermediate router R that can be selected at the source endpoint. The routing is

divided into two parts: first, the packet is minimally routed to R; then, it is minimally routed to the actual destination. VLB has aspects of source routing, namely the choice of R and the modification of the packet in flight, while most of the routing work is done in a destination-based way. As such, VLB natively does not consider multipathing. VLB also incorporates a specific (P) *path selection* (by selecting the intermediate node randomly). This also provides simple, oblivious (L) *load balancing*.

4.11 Universal Globally-Adaptive Load Balanced (UGAL)

Universal Globally-Adaptive Load balanced (UGAL) [101] is an extension of VLB that enables more advantageous routing decisions (R) (P) in the context of load balancing (L). Specifically, when a packet is to be routed, UGAL either selects a path determined by VLB, or a minimum one. The decision usually depends on the congestion in the network. Consequently, UGAL considers multipathing in its design: consecutive packets may be routed using different paths.

4.12 Network Address Aliasing (NAA)

Network Address Aliasing (NAA) is a building block to support multipathing, especially in InfiniBand-based networks. Network Address Aliasing, also known as IP aliasing in Ethernet networks [131] or port virtualization via LID mask control (LMC) in InfiniBand [87, Sec. 7.11.1], is a technique that *assigns **multiple** identifiers to **the same** network endpoint*. This allows the routing protocols to increase the path diversity between two endpoints, and it was used both as a fail-over (enhancing resilience) [164] or for (L) load balancing the traffic (enhancing performance) [54]. In particular, due to the destination-based routing — where a path is only defined by the given destination address; as mandated by the InfiniBand standard [87] — this address aliasing is the only standard-conform and software-based solution to enable multiple disjoint paths between an IB source and a destination port.

4.13 Multi-Railing and Multi-Planes

Various HPC systems employ ***multi-railing***: using multiple injection ports per node into a single topology [75, 167]. Another common scheme is ***multi-plane*** topologies, where nodes are connected to a set of disjoint topologies, either similar [74] or different [112]. This is used to increase path diversity and available throughput. However, this increased level of complexity also comes with additional challenges for the routing protocols to utilize the hardware efficiently.

5 Routing Protocols and Architectures

We now describe representative networking architectures, focusing on their support for *path diversity* and *multipathing*[3], according to the taxonomy described in Section 3. Table 2 illustrates the considered architectures and the associated protocols. Symbols "👍", "👍", and "✖" indicate that a given design offers a given feature, offers a given feature in a limited way, and does not offer a given feature, respectively.

We broadly group the considered designs intro three classes. First (§ 5.1), we describe schemes that belong to the Ethernet and TCP/IP landscape and were introduced for the Internet or for small clusters, most often for the purpose of increasing resilience, with performance being only secondary target. Despite the fact that these schemes originally did not target data centers, we include them as many of these designs were incorporated or used in some way in the data center context. Second, we incorporate Ethernet and TCP/IP related designs that are specifically targeted at data centers or supercomputers (§ 5.2). The last class is dedicated to designs related to InfiniBand (§ 5.3).

5.1 Ethernet & TCP/IP (Clusters, General Networks)

In the first part of Table 2, we illustrate the Ethernet and TCP/IP schemes that are associated with small clusters and general networks. Chronologically, the considered schemes were proposed between 1999 and 2010 (with VIRO from 2011 and MLAG from 2014 being exceptions).

Multiple Spanning Trees (MSTP) [8, 46] extends the STP protocol and it enables creating and managing multiple spanning trees over the same physical network. This is done by assigning different VLANs to different spanning trees, and thus frames/packets belonging to different VLANs can traverse different paths in the network. There exist Cisco's implementations of MSTP, for example Per-VLAN spanning tree (PVST) and Multiple-VLAN Spanning Tree (MVST). Table-based Hashing with Reassignments (THR) [42] extends ECMP to a simple form of load balancing: it selectively reassigns some active flows based on load sharing statistics. Global Open Ethernet (GOE) [88, 89] provides virtual private network (VPN) services in metro-area networks (MANs) using Ethernet. Its routing protocol, per-destination multiple rapid spanning tree protocol (PD-MRSTP), combines MSTP [8] (for using multiple spanning trees for different VLANs) and RSTP [9] (for quick failure recovery). Viking [144] is very similar to GOE. It also relies on MSTP to explicitly seek faster failure recovery *and* more throughput by using a VLAN per spanning tree, which enables redundant switching paths between endpoints. TeXCP [96] is a Traffic Engineering (TE) distributed protocol for balancing traffic in intra-domains of ISP operations. It focuses on algorithms for path selection and load balancing, and briefly discusses a suggested implementation that relies on protocols such as RSVP-TE [13] to deploy paths in routers. TeXCP is similar to another protocol called MATE [56]. TRansparent Interconnection of Lots of Links (TRILL) [157] and Shortest Path Bridging (SPB) [5] are similar schemes that both rely on link state routing to, among others, enable multipathing based on multiple trees and ECMP. Ethernet on Air [139] uses the approach introduced by SEATTLE [99] to eliminate flooding in the switched network. They both rely on LIS and distributed hashtables (DHTs), implemented in switches, to map endpoints to the switches connecting these endpoints to the network. Here, Ethernet on Air uses its DHT to construct a routing substrate in the form of a Directed Acyclic Graph (DAG) between switches. Different paths in this DAG can be used for multipathing. VIRO [91] is similar in relying on the DHT-style routing. It mentions multipathing as a possible feature enabled by multiple virtual topologies built on top of a single physical network. Finally, MLAG [155] and MC-LAG [155] enable multipathing through link aggregation.

First, many of these designs enable *shortest paths*, but

[3] We encourage participation in this survey. In case the reader possesses additional information relevant for the contents, the authors welcome the input. We also encourage the reader to send us any other information that they deem important, e.g., architectures not mentioned in the current survey version.

Routing Scheme	Stack Layer	Features of schemes SP	NP	MP	DP	ALB	AT	Scheme used	Additional remarks and clarifications
Related to Ethernet and TCP/IP (small clusters and general networks):									
OSPF-OMP (OMP) [163]	L3	👍	✖	👍	✖	✖	👍	OSPF	Cisco's enhancement of OSPF to the multipathing setting. Packets from the same flow are forwarded using the same path.
MPA [118]	L3	👍*	👍*	👍*	✖	✖	👍	—	*MPA only focuses on algorithms for *generating* routing paths.
SmartBridge [137]	L2	👍	✖	✖	✖	✖	👍	ST	SmartBridges improves ST; packets are sent between hosts using the shortest possible path in the network.
MSTP [8, 46]	L2	👍*	✖*	👍	✖	✖	👍	ST+VLAN	*Shortest paths are offered only within spanning trees.
STAR [109]	L2	👍	✖	✖	✖	✖	👍	ST	STAR improves ST; frames are forwarded over alternate paths that are shorter than their corresponding ST path.
LSOM [65]	L2	👍	✖	✖	✖	✖	👍	—	LSOM supports mesh networks also in MAN. LSA manages state of links.
AMP [71]	L3	👍	✖	👍	✖	👍	👍	ECMP, OMP	AMP extends ECMP and OSPF-OMP.
RBridges [127]	L2	👍	✖	✖	✖	✖	👍	—	—
THR [42]	L3	👍	✖	👍	✖	👍	👍	ECMP	Table-based Hashing with Reassignments (THR) extends ECMP, it selectively reassigns some active flows based on load sharing statistics.
GOE [88]	L2	👍*	✖*	👍	✖	✖	👍	ST+VLAN	*Shortest paths are offered only within spanning trees. One spanning tree per VLAN is used. Focus on **resiliece**.
Viking [144]	L2	👍*	✖*	👍	✖	✖**	👍	ST+VLAN	*Shortest paths are offered only within spanning trees. One spanning tree per VLAN is used. **Viking uses elaborate load balancing, but it is static.
TeXCP [96]	L3	👍	👍	👍	👍	👍*	👍	—	Routing in ISP, path are computed offline, *load balancing selects paths based on congestion and failures.
CTMP [150]	L3*	👍	✖	👍	👍	👍	👍	ECMP	The scheme focuses on generating paths and on adaptive load balancing. It extends ECMP. *Path generation is agnostic to the layer.
SEATTLE [99]	L2	👍	✖	✖	✖	✖	👍	LIS (DHT)	Packets traverse the shortest paths.
SPB [5], TRILL [157]	L2	👍	✖*	👍	✖	✖	👍	—	—
Ethernet on Air [139]	L2	👍	✖	👍*	✖	✖	👍	LIS (DHT)	*Multipathing is used only for **resilience**.
VIRO [91]	L2–L3	👍	👍	👍*	✖	✖	👍	LIS (DHT)	*Multipathing could be enabled by using multiple virtual networks over the same underlying physical topology.
MLAG [155], MC-LAG [155]	L2	👍*	✖*	👍**	✖	✖	👍	—	*Not all shortest paths are enabled; **multipathing only for **resilience**.
Related to Ethernet and TCP/IP (data centers, supercomputers):									
DCell [77]	L2–L3	✖	👍	✖	✖	✖	✖ (RL)*	—	*DCell comes with a *specific* topology that consists of layers of routers.
Monsoon [72]	L2, L3	👍*	✖*	👍**	✖	✖	✖ (CL)†	VLB, SR, ECMP	*VLB is used in groups of edge routers. **ECMP is used only between border and access routers.
Work by Al-Fares et al. [4]	L3	👍	✖	👍	👍	👍	✖ (FT)	—	—
PortLand [120]	L2	👍	✖	👍*	✖	✖	✖ (FT)	ECMP	—
MOOSE [142]	L2	👍	✖	👍*	✖	✖	👍	OSPF-OMP**, LIS	*Only a brief discussion on augmenting the frame format for multipathing. **only mentioned as a possible mechanism for multipathing in MOOSE.
BCube [76]	L2–L3	👍	✖	👍	👍	✖	✖ (RL)*	—	*BCube comes with a *specific* topology that consists of layers of routers.
VL2 [73]	L3*	👍	✖	👍	✖	👍**	✖ (CL)	LIS, VLB, ECMP	*L3 is used but L2 semantics are offered. **TCP congestion control.
SPAIN [117]	L2	👍*	👍*	👍	👍	✖	👍	ST+VLAN	*SPAIN uses one ST per VLAN. Path diversity is limited by #VLANs supported in L2 switches.
Work by Linden et al. [161]	L3	👍	✖	👍	👍*	👍*	👍	ECMP	*These aspects are only mentioned. The whole design extends ECMP.
Work by Suchara et al. [156]	L3	👍	👍*	👍	👍	👍**	👍	—	*Support is implicit. **Paths are precomputed based on predicted traffic. The design focuses on fault tolerance but also considers performance.
PAST [154]	L2	👍*	👍*	👍**	👍	✖	👍	ST+VLAN, VLB	*PAST enables *either* shortest *or* non-minimal paths. **Limited or no multipathing.
Shadow MACs [1]	L2	👍	✖*	👍	✖	✖	👍	—	*Non-minimal paths are mentioned only in the context of **resilience**.
WCMP for DC [174]	L3	👍*	✖	👍*	✖	✖	✖ (MS)**	ECMP	WCMP uses OpenFlow [113]. *WCMP extends ECMP with hashing of flows based on link capacity. **Applicable to simple 2-stage networks.
Flexible fabric [93]	L3	👍	👍*	✖**	✖	✖	👍†	SR	*Non-minimal paths considered for **resilience** only. **Only mentioned. †Main focus is placed on leaf-spine and fat trees.
XPath [83]	L3	👍	👍*	👍	👍	👍**	👍	—	*Unclear scaling behavior. **XPath relies on default congestion control.
Adaptive load balancing*	L3	👍	✖	👍	✖	👍	✖ (MS)	PR	*Examples are DRILL [68] or DRB [38]
ECMP-VLB [97]	L3	👍	👍	👍	✖	👍	👍 (XP)*	ECMP, VLB	*Focus on the Xpander network.
FatPaths [26]	L2–L3	👍*	👍*	👍	👍	👍	👍**	PR***, ECMP†, VLAN†	*Simultaneous use of shortest and non-minimal paths. **Generally applicable but main focus is on low-diameter topologies. ***FatPaths sprays packets grouped in flowlets. †Only briefly described.
Related to InfiniBand and other traditionally HPC-related designs (data centers, supercomputers):									
Shortest path schemes*	subnet	✖	✖	👍**	✖	✖	👍	D-free	*They incl. Min-Hop [115], (DF-)SSSP [80, 51], and Nue [53]. **Only when combined with NAA.
MUD [110, 61]	subnet	👍	👍	👍	✖	✖	✖*	D-free	*Original proposals disregarded IB's destination-based routing criteria; hence, applicability is limited without NAA.
LASH-TOR [148]	subnet	✖	✖	👍	✖	✖	✖*	D-free	*Original proposals disregarded IB's destination-based routing criteria; hence, applicability is limited without NAA.
Multi-Routing [121]	subnet	👍*	✖	👍	👍	👍**	👍*	—	*Depends on #{network planes} and/or selected routing schemes. **Must be implemented in upper layer protocol, like MPI.
Adaptive Routing [114]	subnet	—	—	👍	—	👍	👍	—	Propriety Mellanox extension are outside of InfiniBand specification.
SAR [52]	subnet	✖	✖	✖	✖	👍*	✖ (FT)	NAA, D-free	*Theoretically in Phase 2 & 4 of 'Property Preserving Network Update'.
PARX [54]	subnet	👍	👍	👍	✖	👍*	✖ (HX)	NAA, D-free	*Implemented via upper layer protocol, e.g. modified MPI library.
Cray's Aries [6]	propr.	👍	👍	👍	✖	👍*	✖ (DF)	UGAL, D-free	*Link congestion information are propagated through the network and used to decide between minimal and non-minimal paths.
Cray's Slingshot [149]	L3 or propr.	👍	👍	👍	✖	👍*	✖ (DF)	UGAL, D-free	*Similar to Aries, adds endpoint congestion mitigation.
Myricom's Myrinet [35]	propr.	👍	👍	?	?	?	👍	SR, D-free	—
Atos' BXI [48]	propr.	👍	?	?	?	👍	👍	D-free	—
EXTOLL's architecture [119]	propr.	👍	?	?	?	👍	👍		—
Intel's OmniPath [34]	propr.	👍	👍	👍*	👍*	👍	👍	D-free	*No built-in support for enforcing packeting ordering across different paths
Quadrics' QsNet [128, 129]	propr.	👍	👍	👍*	👍*	👍	✖ (FT)	SR	*Unclear details on how to use multipathing in practice
IBM's PERCS [12]	propr.	👍	👍	👍	✖	👍*	✖ (DF)	UGAL, D-free	*Routing modes can be set on a per-packet basis.

TABLE 2: **Routing architectures. Rows are sorted chronologically and then by topology/multipathing support. "Scheme used"** indicates incorporated building blocks from Table 1. **"Stack Layer"** indicates the location of a given scheme in the TCP/IP or InfiniBand stack (cf. § 2.4). **SP**, **NP**, **MP**, **DP**, **ALB**, and **AT** illustrate whether a given routing scheme supports various aspects of path diversity. Specifically: **SP**: A given scheme enables using arbitrary **shortest** paths. **NP**: A given scheme enables using arbitrary **non-minimal** paths. **MP**: A given scheme enables **multipathing** (between two hosts). **DP**: A given scheme considers **disjoint** (no shared links) paths. **ALB**: A given scheme offers **adaptive load balancing**. **AT**: A given scheme works with an **arbitrary topology**. 👍: A given scheme does offer a given feature. 👍: A given scheme offers a given feature in a limited way. ✖: A given scheme does not offer a given feature. *Explanations in remarks. MS, FT, CL, XP, and HX are symbols of topologies described in § 2.1. RL is a specific type of a network called "recursive layered" design, described in § 5.2.3. "?": Unknown. "**D-free**": deadlock-free.

a non-negligible number is limited in this respect by the used spanning tree protocol (i.e., the used shortest paths are not shortest with respect to the underlying physical topology). A large number of protocols alleviates this with different strategies. For example, SEATTLE, Ethernet on Air, and VIRO use DHTs that virtualize the physical topology, enabling shortest paths. Other schemes, such as SmartBridge [137] or RBridges [127], directly enhance the spanning tree protocol (in various ways) to enable shortest paths. Second, many protocols also support *multipathing*. Two most common mechanisms for this are either ECMP (e.g., in AMP or THR) or multiple spanning trees combined with VLAN tagging (e.g., in MSTP or GOE). However, *almost no schemes* explicitly support *non-minimal paths*[4]*, disjoint paths, or adaptive load balancing*. Yet, they *all* work on *arbitrary topologies*. All these features are mainly dictated by the purpose and origin of these architectures and protocols. Specifically, most of them were developed with the main goal being *resilient to failures* and *not* higher performance. This explains – for example – almost no support for *adaptive load balancing in response to network congestion*. Moreover, they are all restricted by the technological constraints in general Ethernet and TCP/IP related equipment and protocols, which are historically designed for the general Internet setting. Thus, they have to support *any* network topology. Simultaneously, many such protocols were based on spanning trees. This dictates the nature of multipathing support in these protocols, often using some form of multiple spanning trees (MSTP, GOE, Viking) or "shortcutting" spanning trees (VIRO).

5.2 Ethernet & TCP/IP (Data Centers, Supercomputers)

The designs associated with data centers and supercomputers are listed in the second part of Table 2.

5.2.1 Multistage (Fat Tree, Clos, Leaf-Spine) Designs

One distinctive group of architectures target multistage topologies. A common key feature of all these designs is multipathing based on multiple paths of equal lengths leading via core routers (cf. § 2.1). Common building blocks are ECMP, VLB, and PR; however, details (of how these blocks are exactly deployed) may vary depending on, for example, the specific targeted topology (e.g., fat tree vs. leaf-spine), the targeted stack (e.g., bare L2 Ethernet vs. the L3 IP setting), or whether a given design uses off-the-shelf equipment or rather proposes some HW modifications. Importantly, these designs focus on multipathing with shortest paths because multistage networks offer a rich supply of such paths. They often offer some form of load balancing.

Monsoon [72] provides a hybrid L2–L3 Clos design in which all endpoints in a datacenter form a large single L2 domain. L2 switches may form multiple layers, but the last two layers (access and border) consist of L3 routers. ECMP is used for multipathing between access and border routers. All L2 layers use multipathing based on selecting a random intermediate switch in the uppermost L2 layer (with VLB). To implement this, Monsoon relies on switches that support *MAC-in-MAC tunneling (encapsulation)* [85] so that one may forward a frame via an intermediate switch.

[4]While schemes based on spanning trees strictly speaking enable non-minimal paths, this is not a mechanism for path diversity per se, but limitation dictated by the fact that the used spanning trees often do not enable shortest paths.

PortLand [120] uses fat trees and provides a complete L2 design; it simply assumes standard ECMP for multipathing.

Al-Fares et al. [4] also focus on fat trees. They provide a complete design based on L3 routing. While they only briefly mention multipathing, they use an interesting solution for spreading traffic over core routers. Specifically, they propose that each router maintains a *two-level routing table*. Now, a destination address in a packet may be matched based on its prefix ("level 1"); this matching takes place when a packet is sent to an endpoint in the same pod. If a packet goes to a different pod, the address hits a special entry leading to routing table "level 2". In this level, matching uses the address *suffix* ("right-hand" matching). The key observation is that, while simple prefix matching would force packets (sent to the same subnet) to use the same core router, suffix matching enables selecting different core routers. The authors propose to implement such routing tables with ternary content-addressable memories (TCAM).

VL2 [73] targets Clos and provides a design in which the infrastructure uses L3 but the services are offered L2 semantics. VL2 combines ECMP and VLB for multipathing. To send a packet, a random core router is selected (VLB); ECMP then is used to further spread load across available redundant paths. Using an intermediate core router in VLB is implemented with IP-in-IP encapsulation.

There is a large number of load balancing schemes for multistage networks. The majority focus on the transport layer details and are outside the scope of this work; we outline them in Section 6 and coarsely summarize them in Table 2. An example design, DRB [38], offers round-robin packet spraying and it also discusses how to route such packets in Clos via core routers using IP-in-IP encapsulation.

5.2.2 General Network Designs

There are also architectures that focus on *general* topologies; some of them are tuned for certain classes of networks but may in principle work on any topology [26]. In contrast to architectures for multistage networks, designs for general networks rarely consider ECMP because it is difficult to use ECMP in a context of a general topology, without the guarantee of a rich number of redundant shortest paths, common in Clos or in a fat tree. Instead, they often resort to some combination of ST and VLANs.

SPAIN [117] is an L2 architecture that focuses on using commodity off-the-shelf switches. To enable multipathing in an arbitrary network, SPAIN (1) precomputes a set of redundant paths for different endpoint pairs, (2) merges these paths into trees, and (3) maps each such tree into a separate VLAN. Different VLANs may be used for multipathing between endpoint pairs, assuming used switches support VLANs. While SPAIN relies on TCP congestion control for reacting to failures, it does not offer any specific scheme for load balancing for more performance.

MOOSE [142] addresses the limited scalability of Ethernet; it simply relies on orthogonal designs such as OSPF-OMP for multipathing.

PAST [154] is a complete L2 architecture for general networks. Its key idea is to use a single spanning tree per endpoint. As such, it does not explicitly focus on ensuring multipathing between *pairs* of endpoints, instead focusing on providing path diversity at the granularity of a desti-

nation endpoint, by enabling computing different spanning trees, depending on bandwidth requirements, considered topology, etc.. It enables shortest paths, but also supports VLB by offering algorithms for deriving spanning trees where paths to the root of a tree are not necessarily minimal. PAST relies on ST and VLAN for implementation.

There are also works that focus on encoding a diversity of paths available in different networks. For example, Jyothi et al. [93] discuss encoding arbitrary paths in a data center with OpenFlow to enable flexible fabric, XPath [83] compresses the information of paths in a data center so that they can be aggregated into a practical number of routing entries, and van der Linden et al. [161] discuss how to effectively enable source routing by appropriately transforming selected fields of packet headers to ensure that the ECMP hashing will result in the desired path selection.

Some recent architectures focus on high-performance routing in low-diameter networks. ECMP-VLB is a simple routing scheme suggested for Xpander topologies [97] that, as the name suggests, combines the advantages of ECMP and VLB. Finally, FatPaths [26] targets general low-diameter networks. It (1) divides physical links into *layers* that form acyclic directed graphs, (2) uses paths in different layers for multipathing. Packets are sprayed over such layers using flowlets. FatPaths discusses an implementation based on address space partitioning, VLANs, or ECMP.

5.2.3 Recursive Networks

Some architectures, besides routing, also come with novel "recursive" topologies [77, 76]. The key design choice in these architectures to obtain path diversity is to use multiple NICs per server and connect servers to one another.

5.3 InfiniBand

We now describe the IB landscape. We omit a line of common routing protocols based on shortest paths, as they are not directly related to multipathing, but their implementations in the IB fabric manager natively support NAA; these routings are MinHop [115], SSSP [80], Deadlock-Free SSSP (DFSSSP) [51], and a DFSSSP variant called Nue [53].

5.3.1 Multi-Up*/Down* (MUD) routing

Numerous variations of Multi-Up*/Down* routing have been proposed, e.g., [110, 61], to overcome the bottlenecks and limitations of Up*/Down*. The idea is to utilize a set of Up*/Down* spanning trees—each starting from a different root node—and choose a path depending on certain criteria. For example, Flich at al. [61] proposed to select two roots which either give the highest amount of non-minimal or the highest amount of minimal paths, and then randomly select from those two trees for each source-destination pair. Similarly, Lysne et al. [110] proposed to identify multiple root nodes (by maximizing the minimal distance between them), and load-balance the traffic across the resulting spanning trees to avoid the usual bottleneck near a single root. Both approaches require NAA to work with InfiniBand.

5.3.2 LASH-Transition Oriented Routing (LASH-TOR)

The goal of LASH-TOR [148] is not directly path diversity, however it is a byproduct of how the routing tries to ensure deadlock-freedom (an essential feature in lossless networks) under resource constraints. LASH-TOR uses the LAyered Shortest Path routing for the majority of source-destination pairs, and Up*/Down* as fall-back when LASH would exceed the available virtual channels. Hence, assuming NAA to separate the LASH (minimal paths) from the Up*/Down* (potentially non-minimal path), one can gain limited path diversity in InfiniBand.

5.3.3 Multi-Routing

Multi-routing can be viewed as an extension of the multi-plane designs outlined in § 2.1. In preliminary experiments, researchers have tried if the use of different routing algorithms on similar network planes can have an observable performance gain [121]. Theoretically, additionally to the increased, non-overlapping path diversity resulting from the multi-plane design, utilizing different routing algorithms within each plane can yield benefits for certain traffic patterns and load balancing schemes, which would otherwise be hidden when the same routing is used everywhere.

5.3.4 Adaptive Routing (AR)

For completeness, we list Mellanox's adaptive routing implementation for InfiniBand as well, since it (theoretically) increases path diversity and offers load balancing within the more recent Mellanox-based InfiniBand networks [114]. However, to this date, their technology is proprietary and outside of the IB specifications. Furthermore, Mellanox's AR only supports a limited set up topologies (tori-like, Clos-like and their Dragonfly variation).

5.3.5 Scheduling-Aware Routing (SAR)

Similar to LASH-TOR, the path diversity offered by SAR was not intended as multipathing feature or load balancing feature [52]. Using NAA with LMC $= 1$, SAR employs a primary set of shortest paths, calculated with a modified DF-SSSP routing [51], and a secondary set of paths, calculated with the Up*/Down* routing algorithm. Whenever SAR reroutes the network to adapt to the currently running HPC applications, the network traffic must temporarily switch to the fixed secondary paths to avoid potential deadlocks during the deployment of the new primary forwarding rules. Hence, during each deployment, there is a short time frame where multipathing is intended, but (theoretically) the message passing layer could also utilize both, the primary and secondary paths, simultaneously, outside of the deployment window without breaking SAR's validity.

5.3.6 Pattern-Aware Routing for HyperX (PARX)

PARX is the only known, and practically demonstrated, routing for InfiniBand which intentionally enforces the generation of minimal and non-minimal paths, and mixes the usage of both for load-balancing reasons [54], while still adhering to the IB specifications. The idea of this routing is an emulation of AR capabilities with non-AR techniques/technologies to overcome the bottlenecks on the shortest path between IB switch located in the same dimension of the HyperX topology. PARX for a 2D HyperX, with NAA and LMC $= 2$, offers between 2 and 4 disjoint paths, and adaptively selects minimal or non-minimal routes depending on the message size to optimize for either message latency (with short payloads) or throughput for large messages.

5.4 Other HPC Network Designs

Cray's **Aries** and **Slingshot** adopt the adaptive UGAL routing to distribute the load across the network. When using minimal paths, the packets are sent directly to the dragonfly

destination group. With non-minimal paths, instead, packets are first minimally routed to an intermediate group, then minimally routed to the destination group. Within a group, packets are always minimally routed. Routing decisions are taken on a per-packet basis. They consist in selecting a number of minimal and non-minimal paths, evaluating the load on these paths, and finally selecting one. The load is estimated by using link load information propagated through the network [101]. Applications can select different "biasing levels" for the adaptive routing (e.g., bias towards minimal routing), or disable the adaptive routing and always use minimal or non-minimal paths.

In IBM's **PERCS**, shortest paths lengths vary between one and three hops (i.e., route within the source supernode; reach the destination supernode; route within the destination supernode). Non-minimal paths can be derived by minimally-routing packets towards an intermediate supernode. The maximum non-minimal path length is five hops. As pairs of supernodes can be connected by more than one link, multiple shortest paths can exist. PERCS provides three routing modes that can be selected by applications on a per-packet basis: non-minimal, with the applications defining the intermediate supernode; round-robin, with the hardware selecting among the multiple routes in a round-robin manner; randomized (only for non-minimal paths), where the hardware randomly chooses an intermediate supernode.

Quadrics' **QsNet** [128, 129] is a source routed interconnect that enables, to some extent, multipathing between two endpoints, and comes with adaptivity in switches. Specifically, a single routing table (deployed in a QsNet NIC called "Elan") translates a processor ID to a specification of a path in the network. Now, as QsNet enables loading several routing tables, one could encode different paths in different routing tables. Finally, QsNet offers hardware support for broadcasts, and for multicasts to physically contiguous QsNet endpoints.

Intel's **OmniPath** [34] offers two mechanisms for multipathing between any two endpoints: different paths in the fabric or different virtual lanes within the same physical route. However, the OmniPath architecture itself does not prescribe specific mechanisms to select a specific path. Moreover, it does not provide any scheme for ensuring packet ordering. Thus, when such ordering is needed, the packets must use the same path, or the user must provide other scheme for maintaining the right ordering.

Finally, the specifications of Myricom's **Myrinet** [35] or **Open-MX** [70], Atos' **BXI** [48], and EXTOLL's interconnect [119] do not disclose details on their support for multipathing. Myrinet does use source routing and works on arbitrary topologies. Both BXI and EXTOLL's design offer adaptive routing to mitigate congestion, but it is unclear if multipathing is used.

6 Related Aspects of Networking

Congestion control & load balancing are strongly related to the transport layer (L4). This area was extensively surveyed [168]. Thus, we do not focus on these aspects and we only mention them whenever necessary. Overall, such adaptive load balancing can be implemented using flows [57], flowcells (fixed-sized packet series) [79], flowlets (variable-size packet series) [162], and single packets [68]. In data centers, load balancing often focuses on flow and flowlet based adaptivity. This is because the targeted stack is often based on TCP that suffers performance degradation whenever packets become reordered. In contrast, HPC networks usually use packet level adaptivity, and research focuses on choosing good congestion signals, often with hardware modifications [64, 63].

Similarly to congestion control, we exclude **flow control** from our focus, as it is also usually implemented within L4.

Some works **analyze various properties of low-diameter topologies**, for example path length, throughput, and bandwidth [94]. Such works could be used in combination with our multipathing analysis when developing routing protocols and architectures that take advantage of different properties of a given topology.

7 Challenges

There are many challenges related to multipathing and path diversity support in HPC systems and data centers.

First, we predict a rich line of future routing protocols and networking architectures targeting recent low-diameter topologies. Some of the first examples are the FatPaths architecture [26] or the PARX routing [54]. However, more research is required to understand how to fully use the potential behind such networks, especially considering more effective congestion control and different technological constraints in existing networking stacks.

Moreover, little research exists into routing schemes suited specifically for particular types of workloads, for example deep learning [18], linear algebra computations [24, 31, 151, 102], graph processing [20, 30, 33, 28, 23, 69, 29, 25], and other distributed workloads [21, 22, 67] and algorithms [141, 140]. For example, as some workloads (e.g., in deep learning [17]) have more predicable communication patterns, one could try to gain speedups with multipath routing based on the structural network properties that are static or change slowly. Contrarily, when routing data-driven workloads such as graph computing, one could bias more aggressively towards adaptive multipathing, for example with flowlets [26, 162].

Finally, we expect the growing importance of various schemes enabling programmable routing and transport [10, 39]. Here, one line of research will probably heavily depend on OpenFlow [113] and, especially, P4 [36]. It is also interesting to investigate how to use FPGAs [32, 27, 59] or "smart NICs" [49, 81, 39] in the context of multipathing.

8 Conclusion

Developing high-performance routing protocols and networking architectures in HPC systems and data centers is an important research area. Multipathing and overall support for path diversity is an important part of such designs, and specifically one of the enablers for high performance. The importance of routing is increased by the prevalence of communication intensive workloads that put pressure on the interconnect, such as graph analytics or deep learning.

Many networking architectures and routing protocols have been developed. They offer different forms of support for multipathing, they are related to different parts of various networking stacks, and they are based on miscellaneous classes of simple routing building blocks or design princi-

ples. To propel research into future developments in the area of high-performance routing, we present the first analysis and taxonomy of the rich landscape of multipathing and path diversity support in the routing designs in supercomputers and data centers. We identify basic building blocks, we crystallize fundamental concepts, we list and categorize existing architectures and protocols, and we discuss key design choices, focusing on the support for different forms of multipathing and path diversity. Our analysis can be used by network architects, system developers, and routing protocol designers who want to understand how to maximize the performance of their developments in the context of bare Ethernet, full TCP/IP, or InfiniBand and other HPC stacks.

Acknowledgment The work was supported by JSPS KAKENHI Grant Number JP19H04119.

REFERENCES

[1] K. Agarwal et al. “Shadow MACs: scalable label-switching for commodity ethernet”. In: *HotSDN'14*. 2014, pp. 157–162.

[2] S. Aggarwal and P. Mittal. “Performance Evaluation of Single path and Multipath regarding Bandwidth and Delay”. In: *Intl. J. Comp. App.* 145.9 (2016).

[3] J. H. Ahn et al. “HyperX: topology, routing, and packaging of efficient large-scale networks”. In: *ACM/IEEE Supercomputing*. 2009, p. 41.

[4] M. Al-Fares, A. Loukissas, and A. Vahdat. “A scalable, commodity data center network architecture”. In: *ACM SIGCOMM*. 2008, pp. 63–74.

[5] D. Allan et al. “Shortest path bridging: Efficient control of larger ethernet networks”. In: *IEEE Communications Magazine* 48.10 (2010).

[6] B. Alverson et al. “Cray XC series network”. In: *Cray Inc., White Paper WP-Aries 01-1112* (2012).

[7] A. A. Anasane and R. A. Satao. “A survey on various multipath routing protocols in wireless sensor networks”. In: *Procedia Computer Science* 79 (2016), pp. 610–615.

[8] ANSI/IEEE. “Amendment 3 to 802.1Q virtual bridged local area networks: Multiple spanning trees”. In: *ANSI/IEEE Draft Standard P802.1s/D11.2* (2001).

[9] ANSI/IEEE. “Virtual Bridged Local Area Networks Amendment 4: Provider Bridges”. In: *ANSI/IEEE Draft Standard P802.1ad/D1* (2003).

[10] M. T. Arashloo et al. “Enabling Programmable Transport Protocols in High-Speed NICs”. In: *NSDI*. 2020.

[11] H. D. E. Al-Ariki and M. S. Swamy. “A survey and analysis of multipath routing protocols in wireless multimedia sensor networks”. In: *Wireless Networks* 23.6 (2017).

[12] B. Arimilli et al. “The PERCS High-Performance Interconnect”. In: *Hot Interconnects 2010*. IEEE.

[13] D. Awduche et al. *RSVP-TE: extensions to RSVP for LSP tunnels*. 2001.

[14] S. Azizi, N. Hashemi, and A. Khonsari. “HHS: an efficient network topology for large-scale data centers”. In: *The Journal of Supercomputing* 72.3 (2016), pp. 874–899.

[15] M. F. Bari et al. “Data center network virtualization: A survey”. In: *IEEE Communications Surveys & Tutorials* 15.2 (2012), pp. 909–928.

[16] A. Beloglazov et al. “A taxonomy and survey of energy-efficient data centers and cloud computing systems”. In: *Advances in computers*. Vol. 82. Elsevier, 2011, pp. 47–111.

[17] T. Ben-Nun and T. Hoefler. “Demystifying parallel and distributed deep learning: An in-depth concurrency analysis”. In: *ACM CSUR* (2019).

[18] T. Ben-Nun et al. “A Modular Benchmarking Infrastructure for High-Performance and Reproducible Deep Learning”. In: *arXiv preprint arXiv:1901.10183* (2019).

[19] M. Besta and T. Hoefler. “Slim Fly: A Cost Effective Low-Diameter Network Topology”. In: ACM/IEEE Supercomputing. Nov. 2014.

[20] M. Besta and T. Hoefler. “Accelerating irregular computations with hardware transactional memory and active messages”. In: *ACM HPDC*. 2015.

[21] M. Besta and T. Hoefler. “Active access: A mechanism for high-performance distributed data-centric computations”. In: *ACM ICS*. 2015.

[22] M. Besta and T. Hoefler. “Fault tolerance for remote memory access programming models”. In: *ACM HPDC*. 2014, pp. 37–48.

[23] M. Besta and T. Hoefler. “Survey and taxonomy of lossless graph compression and space-efficient graph representations”. In: *arXiv preprint arXiv:1806.01799* (2018).

[24] M. Besta et al. “Communication-Efficient Jaccard Similarity for High-Performance Distributed Genome Comparisons”. In: *IEEE IPDPS* (2020).

[25] M. Besta et al. “Demystifying graph databases: Analysis and taxonomy of data organization, system designs, and graph queries”. In: *arXiv preprint arXiv:1910.09017* (2019).

[26] M. Besta et al. “FatPaths: Routing in Supercomputers and Data Centers when Shortest Paths Fall Short”. In: *ACM/IEEE Supercomputing* (2020).

[27] M. Besta et al. “Graph Processing on FPGAs: Taxonomy, Survey, Challenges”. In: *arXiv preprint arXiv:1903.06697* (2019).

[28] M. Besta et al. “Log (graph) a near-optimal high-performance graph representation”. In: *ACM PACT*. 2018, pp. 1–13.

[29] M. Besta et al. “Practice of Streaming and Dynamic Graphs: Concepts, Models, Systems, and Parallelism”. In: *arXiv preprint arXiv:1912.12740* (2019).

[30] M. Besta et al. “Slim NoC: A low-diameter on-chip network topology for high energy efficiency and scalability”. In: *ACM SIGPLAN Notices*. 2018.

[31] M. Besta et al. “Slimsell: A vectorizable graph representation for breadth-first search”. In: *IEEE IPDPS*. 2017, pp. 32–41.

[32] M. Besta et al. “Substream-Centric Maximum Matchings on FPGA”. In: *ACM/SIGDA FPGA*. 2019, pp. 152–161.

[33] M. Besta et al. “To push or to pull: On reducing communication and synchronization in graph computations”. In: *ACM HPDC*. 2017.

[34] M. S. Birrittella et al. “Intel® Omni-path architecture: Enabling scalable, high performance fabrics”. In: *IEEE HOTI*. 2015.

[35] N. J. Boden et al. “Myrinet: A gigabit-per-second local area network”. In: *IEEE micro* (1995).

[36] P. Bosshart et al. “P4: Programming protocol-independent packet processors”. In: *ACM SIGCOMM Computer Communication Review* 44.3 (2014), pp. 87–95.

[37] C. Camarero et al. “Projective networks: Topologies for large parallel computer systems”. In: *IEEE TPDS* 28.7 (2016), pp. 2003–2016.

[38] J. Cao et al. “Per-packet load-balanced, low-latency routing for clos-based data center networks”. In: *ACM CoNEXT*. 2013, pp. 49–60.

[39] A. Caulfield, P. Costa, and M. Ghobadi. “Beyond SmartNICs: Towards a fully programmable cloud”. In: *IEEE HPSR*. IEEE. 2018, pp. 1–6.

[40] K. Chen et al. “Survey on routing in data centers: insights and future directions”. In: *IEEE network* 25.4 (2011), pp. 6–10.

[41] L. Chen, B. Li, and B. Li. “Allocating bandwidth in datacenter networks: A survey”. In: *Journal of Computer Science and Technology* 29.5 (2014), pp. 910–917.

[42] T. W. Chim and K. L. Yeung. “Traffic distribution over equal-cost-multi-paths”. In: *IEEE International Conference on Communications*. Vol. 2. 2004, pp. 1207–1211.

[43] C. Clos. “A Study of Non-Blocking Switching Networks”. In: *Bell Labs Technical Journal* 32.2 (1953), pp. 406–424.

[44] W. J. Dally and B. P. Towles. *Principles and practices of interconnection networks*. Elsevier, 2004.

[45] W. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003. ISBN: 0122007514.

[46] A. F. De Sousa. “Improving load balance and resilience of ethernet carrier networks with ieee 802.1 s multiple spanning tree protocol”. In: *IEEE ICN/ICONS/MCL*. 2006.

[47] C. DeCusatis. *Handbook of fiber optic data communication: a practical guide to optical networking*. Academic Press, 2013.

[48] S. Derradji et al. “The BXI interconnect architecture”. In: *IEEE HOTI*. 2015.

[49] S. Di Girolamo et al. “Network-Accelerated Non-Contiguous Memory Transfers”. In: *arXiv preprint arXiv:1908.08590* (2019).

[50] A. Dixit et al. “On the impact of packet spraying in data center networks”. In: *INFOCOM, 2013 Proceedings IEEE*. IEEE. 2013, pp. 2130–2138.

[51] J. Domke, T. Hoefler, and W. Nagel. “Deadlock-Free Oblivious Routing for Arbitrary Topologies”. In: *IEEE IPDPS*. 2011.

[52] J. Domke and T. Hoefler. “Scheduling-Aware Routing for Supercomputers”. In: *ACM/IEEE Supercomputing*. 2016.

[53] J. Domke, T. Hoefler, and S. Matsuoka. “Routing on the Dependency Graph: A New Approach to Deadlock-Free High-Performance Routing”. In: *ACM HPDC*. 2016.

[54] J. Domke et al. “HyperX Topology: First At-Scale Implementation and Comparison to the Fat-Tree”. In: *ACM/IEEE Supercomputing*. 2019.

[55] J. J. Dongarra, H. W. Meuer, E. Strohmaier, et al. “TOP500 supercomputer sites”. In: *Supercomputer* 13 (1997), pp. 89–111.

[56] A. Elwalid et al. “MATE: MPLS adaptive traffic engineering”. In: *IEEE INFOCOM*. 2001.

[57] M. Al-Fares et al. “Hedera: Dynamic Flow Scheduling for Data Center Networks.” In: *NSDI*. Vol. 10. 2010, pp. 19–19.

[58] D. Farinacci et al. “The locator/id separation protocol (lisp) rfc 6830”. In: (2013).

[59] J. de Fine Licht et al. “Transformations of High-Level Synthesis Codes for High-Performance Computing”. In: *arXiv:1805.08288* (2018).

[60] M. Flajslik et al. “Megafly: A topology for exascale systems”. In: *International Conference on High Performance Computing*. Springer. 2018, pp. 289–310.

[61] J. Flich et al. “Improving InfiniBand Routing Through Multiple Virtual Networks”. In: *ISHPC*. 2002.

[62] K.-T. Foerster and S. Schmid. “Survey of Reconfigurable Data Center Networks: Enablers, Algorithms, Complexity”. In: *ACM SIGACT News* 50.2 (2019), pp. 62–79.

[63] M. Garcia et al. “Efficient Routing Mechanisms for Dragonfly Networks”. In: *42nd International Conference on Parallel Processing, ICPP*. 2013, pp. 582–592.

[64] M. Garcia et al. “On-the-Fly Adaptive Routing in High-Radix Hierarchical Networks”. In: *41st International Conference on Parallel Processing, ICPP*. 2012, pp. 279–288.

[65] R. Garcia et al. “LSOM: A link state protocol over mac addresses for metropolitan backbones using optical ethernet switches”. In: *IEEE NCA*. 2003.

[66] P. Geoffray. “Myrinet express (MX): Is your interconnect smart?” In: *IEEE HPC Asia*. 2004.

[67] R. Gerstenberger et al. “Enabling Highly-scalable Remote Memory Access Programming with MPI-3 One Sided”. In: *Proc. of ACM/IEEE Supercomputing*. 2013.

[68] S. Ghorbani et al. “DRILL: Micro load balancing for low-latency data center networks”. In: *ACM SIGCOMM*. 2017.

[69] L. Gianinazzi et al. “Communication-avoiding parallel minimum cuts and connected components”. In: *ACM SIGPLAN Notices*. Vol. 53. 1. ACM. 2018, pp. 219–232.

[70] B. Goglin. “Design and implementation of Open-MX: High-performance message passing over generic Ethernet hardware”. In: *IEEE IPDPS*. 2008.

[71] I. Gojmerac, T. Ziegler, and P. Reichl. “Adaptive multipath routing based on local distribution of link load information”. In: *Springer QofIS*. 2003.

[72] A. Greenberg et al. “Towards a next generation data center architecture: scalability and commoditization”. In: *ACM PRESTO*. 2008.

[73] A. Greenberg et al. “VL2: a scalable and flexible data center network”. In: *ACM SIGCOMM computer communication review* 39.4 (2009), pp. 51–62.

[74] GSIC, Tokyo Institute of Technology. *TSUBAME2.5 Hardware and Software*. Nov. 2013.

[75] GSIC, Tokyo Institute of Technology. *TSUBAME3.0 Hardware and Software Specifications*. July 2017.

[76] C. Guo et al. “BCube: a high performance, server-centric network architecture for modular data centers”. In: *ACM SIGCOMM CCR* 39.4 (2009), pp. 63–74.

[77] C. Guo et al. “Dcell: a scalable and fault-tolerant network structure for data centers”. In: *ACM SIGCOMM Computer Communication Review*. Vol. 38. 4. ACM. 2008, pp. 75–86.

[78] M. Handley et al. “Re-architecting datacenter networks and stacks for low latency and high performance”. In: *ACM SIGCOMM*. 2017.

[79] K. He et al. “Presto: Edge-based Load Balancing for Fast Datacenter Networks”. In: *ACM SIGCOMM*. 2015.

[80] T. Hoefler, T. Schneider, and A. Lumsdaine. “Optimized Routing for Large-Scale InfiniBand Networks”. In: *IEEE HOTI*. 2009.

[81] T. Hoefler et al. “sPIN: High-performance streaming Processing in the Network”. In: *ACM/IEEE Supercomputing*. 2017.

[82] C. Hopps. *RFC 2992: Analysis of an Equal-Cost Multi-Path Algorithm*. 2000.

[83] S. Hu et al. “Explicit path control in commodity data centers: Design and applications”. In: *IEEE/ACM Transactions on Networking* 24.5 (2016), pp. 2768–2781.

[84] X. Huang and Y. Fang. “Performance study of node-disjoint multipath routing in vehicular ad hoc networks”. In: *IEEE Transactions on Vehicular Technology* (2009).

[85] IEEE. *IEEE 802.1ah standard*. http://www.ieee802.org/1/pages/802.1ah.html, 2008.

[86] Infiniband Trade Association and others. *RoCEv2*. 2014.

[87] InfiniBand® Trade Association. *InfiniBandTM Architecture Specification Volume 1 Release 1.3 (General Specifications)*. Mar. 2015.

[88] A. Iwata et al. “Global open Ethernet (GOE) system and its performance evaluation”. In: *IEEE Journal on Selected Areas in Communications* 22.8 (2004), pp. 1432–1442.

[89] A. IWATE. “Global Optical Ethernet Architecture As Cost-Effective Scalable VPN solution”. In: *NFOEC*. 2002.

[90] R. Jain and S. Paul. “Network virtualization and software defined networking for cloud computing: a survey”. In: *IEEE Communications Magazine* 51.11 (2013), pp. 24–31.
[91] S. Jain et al. “Viro: A scalable, robust and namespace independent virtual id routing for future networks”. In: *IEEE INFOCOM*. 2011.
[92] R. P. Joglekar and P. Game. “Managing congestion in data center network using congestion notification algorithms”. In: *IRJET* (2016).
[93] S. A. Jyothi, M. Dong, and P. Godfrey. “Towards a flexible data center fabric with source routing”. In: *ACM SOSR*. 2015.
[94] S. A. Jyothi et al. “Measuring and understanding throughput of network topologies”. In: *ACM/IEEE Supercomputing*. 2016.
[95] C. Kachris and I. Tomkos. “A survey on optical interconnects for data centers”. In: *IEEE Communications Surveys & Tutorials* 14.4 (2012), pp. 1021–1036.
[96] S. Kandula et al. “Walking the tightrope: Responsive yet stable traffic engineering”. In: *ACM SIGCOMM CCR*. Vol. 35. 4. ACM. 2005, pp. 253–264.
[97] S. Kassing et al. “Beyond fat-trees without antennae, mirrors, and disco-balls”. In: *ACM SIGCOMM*. 2017, pp. 281–294.
[98] G. Kathareios et al. “Cost-effective diameter-two topologies: Analysis and evaluation”. In: *ACM/IEEE Supercomputing*. 2015.
[99] C. Kim, M. Caesar, and J. Rexford. “Floodless in seattle: a scalable ethernet architecture for large enterprises”. In: *ACM SIGCOMM*. 2008, pp. 3–14.
[100] J. Kim, W. J. Dally, and D. Abts. “Flattened butterfly: a cost-efficient topology for high-radix networks”. In: *ACM SIGARCH Comp. Arch. News*. 2007.
[101] J. Kim et al. “Technology-driven, highly-scalable dragonfly topology”. In: *2008 International Symposium on Computer Architecture*. IEEE. 2008, pp. 77–88.
[102] G. Kwasniewski et al. “Red-blue pebbling revisited: near optimal parallel matrix-matrix multiplication”. In: *ACM/IEEE Supercomputing*. ACM. 2019, p. 24.
[103] G. M. Lee and J. Choi. “A survey of multipath routing for traffic engineering”. In: *Information and Communications University, Korea* (2002).
[104] F. Lei et al. “Galaxyfly: A novel family of flexible-radix low-diameter topologies for large-scales interconnection networks”. In: *ACM ICS*. 2016.
[105] C. E. Leiserson et al. “The Network Architecture of the Connection Machine CM-5”. In: *J. Parallel Distrib. Comput.* 33.2 (1996), pp. 145–158.
[106] M. Li et al. “Multipath transmission for the internet: A survey”. In: *IEEE Communications Surveys & Tutorials* 18.4 (2016), pp. 2887–2925.
[107] S. Liu, H. Xu, and Z. Cai. “Low latency datacenter networking: A short survey”. In: *arXiv preprint arXiv:1312.3455* (2013).
[108] LS Committee and others. “IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks”. In: *IEEE Std* 802 (2006).
[109] K.-S. Lui, W. C. Lee, and K. Nahrstedt. “STAR: a transparent spanning tree bridge protocol with alternate routing”. In: *ACM SIGCOMM CCR* 32.3 (2002), pp. 33–46.
[110] O. Lysne and T. Skeie. “Load Balancing of Irregular System Area Networks Through Multiple Roots”. In: *CIC*. CSREA Press, 2001.
[111] G. Maglione-Mathey et al. “Scalable deadlock-free deterministic minimal-path routing engine for infiniband-based dragonfly networks”. In: *IEEE TPDS* (2017).
[112] S. Matsuoka. *A64fx and Fugaku: A Game Changing, HPC / AI Optimized Arm CPU for Exascale*. San Diego, CA, Sept. 2019.
[113] N. McKeown et al. “OpenFlow: enabling innovation in campus networks”. In: *ACM SIGCOMM Computer Communication Review* 38.2 (2008), pp. 69–74.
[114] Mellanox Technologies. *How To Configure Adaptive Routing and SHIELD (New)*. Nov. 2019.
[115] Mellanox Technologies. *Mellanox OFED for Linux User Manual Rev. 2.0-3.0.0*. Aug. 2013.
[116] J. Moy. *OSPF version 2*. Tech. rep. 1997.
[117] J. Mudigonda et al. “SPAIN: COTS Data-Center Ethernet for Multipathing over Arbitrary Topologies.” In: *NSDI*. 2010, pp. 265–280.
[118] P. Narvaez, K.-Y. Siu, and H.-Y. Tzeng. “Efficient algorithms for multi-path link-state routing”. In: (1999).
[119] S. Neuwirth et al. “Scalable communication architecture for network-attached accelerators”. In: *IEEE HPCA*. 2015.
[120] R. Niranjan Mysore et al. “Portland: a scalable fault-tolerant layer 2 data center network fabric”. In: *ACM SIGCOMM CCR* 39.4 (2009), pp. 39–50.
[121] A. Nomura et al. “Performance Evaluation of Multi-rail InfiniBand Network in TSUBAME2.0 (in Japanese)”. In: *IPSJ SIG Technical Report* 2012 (2012).
[122] M. Noormohammadpour and C. S. Raghavendra. “Datacenter traffic control: Understanding techniques and tradeoffs”. In: *IEEE Comm. Surveys & Tutorials* (2017).
[123] D. Oran. *OSI IS-IS intra-domain routing protocol*. Tech. rep. 1990.
[124] R. Peñaranda et al. “The k-ary n-direct s-indirect family of topologies for large-scale interconnection networks”. In: *The Journal of Supercomputing* (2016).
[125] I. Pepelnjak. *EIGRP network design solutions*. Cisco press, 1999.
[126] R. Perlman. “An algorithm for distributed computation of a spanningtree in an extended lan”. In: *ACM SIGCOMM CCR*. Vol. 15. 4. ACM. 1985, pp. 44–53.
[127] R. Perlman. “Rbridges: transparent routing”. In: *IEEE INFOCOM*. 2004.
[128] F. Petrini et al. “Performance evaluation of the quadrics interconnection network”. In: *Cluster Computing* (2003).
[129] F. Petrini et al. “The Quadrics network: High-performance clustering technology”. In: *IEEE Micro* (2002).
[130] G. F. Pfister. “An introduction to the infiniband architecture”. In: *High Performance Mass Storage and Parallel I/O* 42 (2001), pp. 617–632.
[131] J.-M. Pittet. *IP and ARP over HIPPI-6400 (GSN)*. RFC 2835. May 2000.
[132] T. N. Platform. *THE TUG OF WAR BETWEEN INFINIBAND AND ETHERNET*. https://www.nextplatform.com/2017/10/30/tug-war-infiniband-ethernet/.
[133] M. Polese et al. “A survey on recent advances in transport layer protocols”. In: *IEEE Communications Surveys & Tutorials* 21.4 (2019), pp. 3584–3608.
[134] M. Radi et al. “Multipath routing in wireless sensor networks: survey and research challenges”. In: *Sensors* 12.1 (2012), pp. 650–685.
[135] M. S. Rahman et al. “Load-balanced slim fly networks”. In: *Proceedings of the 47th International Conference on Parallel Processing*. 2018, pp. 1–10.
[136] Y. Ren et al. “A survey on TCP Incast in data center networks”. In: *International Journal of Communication Systems* 27.8 (2014), pp. 1160–1172.
[137] T. L. Rodeheffer, C. A. Thekkath, and D. C. Anderson. “SmartBridge: A scalable bridge architecture”. In: *ACM SIGCOMM CCR* 30.4 (2000), pp. 205–216.
[138] E. Rosen, A. Viswanathan, R. Callon, et al. “Multiprotocol label switching architecture”. In: (2001). RFC 3031, January.
[139] D. Sampath, S. Agarwal, and J. Garcia-Luna-Aceves. “‘Ethernet on AIR’: Scalable Routing in very Large Ethernet-Based Networks”. In: *IEEE ICDCS*. 2010.
[140] P. Schmid, M. Besta, and T. Hoefler. “High-performance distributed RMA locks”. In: *ACM HPDC*. 2016, pp. 19–30.
[141] H. Schweizer, M. Besta, and T. Hoefler. “Evaluating the cost of atomic operations on modern architectures”. In: *IEEE PACT*. 2015, pp. 445–456.
[142] M. Scott, A. Moore, and J. Crowcroft. “Addressing the Scalability of Ethernet with MOOSE”. In: *Proc. DC CAVES Workshop*. 2009.
[143] S. Sen et al. “Scalable, optimal flow routing in datacenters via local link balancing”. In: *CoNEXT*. 2013.
[144] S. Sharma et al. “Viking: A multi-spanning-tree Ethernet architecture for metropolitan area and cluster networks”. In: *IEEE INFOCOM*. 2004.
[145] J. Shuja et al. “Survey of techniques and architectures for designing energy-efficient data centers”. In: *IEEE Systems Journal* 10.2 (2014), pp. 507–519.
[146] S. K. Singh, T. Das, and A. Jukan. “A survey on internet multipath routing and provisioning”. In: *IEEE Communications Surveys & Tutorials* 17.4 (2015), pp. 2157–2175.
[147] A. Singla et al. “Jellyfish: Networking data centers randomly”. In: *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI)* (2012).
[148] T. Skeie et al. “LASH-TOR: A Generic Transition-Oriented Routing Algorithm”. In: *ICPADS*. IEEE Computer Society, 2004, p. 595.
[149] *Slingshot: The Interconnect for the Exascale Era - Cray Inc.* https://www.cray.com/sites/default/files/Slingshot-The-Interconnect-for-the-Exascale-Era.pdf.
[150] S. Sohn, B. L. Mark, and J. T. Brassil. “Congestion-triggered multipath routing based on shortest path information”. In: *IEEE ICCCN*. 2006.
[151] E. Solomonik et al. “Scaling betweenness centrality using communication-efficient sparse matrix multiplication”. In: *ACM/IEEE Supercomputing*. 2017, p. 47.
[152] P. Sreekumari and J.-i. Jung. “Transport protocols for data center networks: a survey of issues, solutions and challenges”. In: *Photonic Network Comm.* 31.1 (2016).
[153] A. Sridharan et al. “Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks”. In: *IEEE/ACM TON* 13.2 (2005), pp. 234–247.
[154] B. Stephens et al. “PAST: Scalable Ethernet for data centers”. In: *ACM CoNEXT*. 2012.
[155] K. Subramanian. *Multi-chassis link aggregation on network devices*. US Patent 8,761,005. June 2014.
[156] M. Suchara et al. “Network architecture for joint failure recovery and traffic engineering”. In: *ACM SIGMETRICS*. 2011.
[157] J. Touch and R. Perlman. *Transparent interconnection of lots of links (TRILL): Problem and applicability statement*. Tech. rep. 2009.
[158] J. Tsai and T. Moors. “A review of multipath routing protocols: From wireless ad hoc to mesh networks”. In: *ACoRN workshop on wireless multihop networking*. 2006.
[159] A. Valadarsky, M. Dinitz, and M. Schapira. “Xpander: Unveiling the Secrets of High-Performance Datacenters”. In: *ACM HotNets*. 2015.
[160] L. Valiant. “A scheme for fast parallel communication”. In: *SIAM journal on computing* 11.2 (1982), pp. 350–361.
[161] S. Van der Linden, G. Detal, and O. Bonaventure. “Revisiting next-hop selection in multipath networks”. In: *ACM SIGCOMM CCR*. Vol. 41. 4. 2011.
[162] E. Vanini et al. “Let It Flow: Resilient Asymmetric Load Balancing with Flowlet Switching”. In: *NSDI*. 2017, pp. 407–420.
[163] C. Villamizar. “OSPF optimized multipath (OSPF-OMP)”. In: (1999).
[164] A. Vishnu et al. “Automatic Path Migration over InfiniBand: Early Experiences”. In: *IEEE IPDPS*. 2007.
[165] B. Wang et al. “A survey on data center networking for cloud computing”. In: *Computer Networks* 91 (2015), pp. 528–547.
[166] K. Wen et al. “Flexfly: Enabling a reconfigurable dragonfly through silicon photonics”. In: *ACM/IEEE Supercomputing*. 2016.
[167] N. Wolfe et al. “Preliminary Performance Analysis of Multi-rail Fat-tree Networks”. In: *IEEE/ACM CCGrid*. 2017.
[168] C. Xu, J. Zhao, and G.-M. Muntean. “Congestion control design for multipath transport protocols: A survey”. In: *IEEE communications surveys & tutorials* (2016).
[169] X. Xu et al. “Unified Source Routing Instructions using MPLS Label Stack”. In: *IETF MPLS Working Group draft, Internet Eng. Task Force* (2017).
[170] Yahho Finance. *Mellanox Delivers Record First Quarter 2020 Financial Results*. https://finance.yahoo.com/news/mellanox-delivers-record-first-quarter-200500726.htm.
[171] P. Yébenes et al. “Improving non-minimal and adaptive routing algorithms in slim fly networks”. In: *IEEE HOTI*. 2017.
[172] J. Zhang, F. Ren, and C. Lin. “Survey on transport control in data center networks”. In: *IEEE Network* 27.4 (2013), pp. 22–26.
[173] J. Zhang et al. “Optimizing network performance using weighted multipath routing”. In: *IEEE ICCCN*. 2012.
[174] J. Zhou et al. “WCMP: Weighted cost multipathing for improved fairness in data centers”. In: *ACm EuroSys*. 2014.
[175] S. M. Zin et al. “Survey of secure multipath routing protocols for WSNs”. In: *Journal of Network and Computer Applications* 55 (2015), pp. 123–153.
[176] A. Zinin and I. Cisco. *Routing: Packet Forwarding and Intra-domain Routing Protocols*. 2002.

Maciej Besta is a PhD student at ETH Zurich. His research focuses on high-performance networking and large-scale irregular graph processing.

Jens Domke is a postdoctoral researcher of the High Performance Big Data Research Team at the RIKEN Center for Computational Science, Japan. His research focus is on interconnects, topologies, and routing for HPC systems, as well as performance evaluation and optimization of parallel applications.

Marcel Schneider is a researcher at CERN, Geneva. His research as a MSc student at ETH Zurich focused on simulation of large-scale low diameter networks.

Timo Schneider is a researcher at ETH Zurich. He works on high-performance networking and heterogeneous computing.

Marek Konieczny is a researcher of Computer Science at AGH University of Science and Technology. His research in the Distributed Systems Research Group focuses on networked and data-intensive systems.

Salvatore Di Girolamo is a PhD student at ETH Zurich. He works on high-performance networking, focusing on network offloading.

Ankit Singla is an Assistant Professor at ETH Zurich, where he leads the Network Design & Architecture Lab. His research aims at understanding and improving large modern-day networks, such as the Internet and data center networks.

Torsten Hoefler is a Professor at ETH Zurich, where he leads the Scalable Parallel Computing Lab. His research aims at understanding performance of parallel computing systems ranging from parallel computer architecture through parallel programming to parallel algorithms.