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

Maciej Besta¹, Jens Domke², Marcel Schneider¹, Marek Konieczny², Salvatore Di Girolamo¹, Timo Schneider¹, Ankit Singla¹, Torsten Hoefler¹

¹Department of Computer Science, ETH Zurich; ²RIKEN Center for Computational Science (R-CCS) ³Department 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 traditional HPC systems and supercomputers, and for data centers and 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 [137] and related networks such as Clos [58] are the most commonly deployed topologies in data centers and supercomputers today, dominating the landscape of Ethernet clusters [160], [96], [212]. 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 [34].

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 [42], 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) [34], [32] (a variant with diameter 2), Dragonfly (DF) [133] (the “balanced” variant with diameter 3), Jellyfish (JF) [196] (with diameter 3), Xpander (XP) [212] (with diameter \( \leq 3 \)), and HyperX (Hamming graph) (HX) [4] that generalizes Flattened Butterflies (FBF) [132] with diameter 3. As observed [42], “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, 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 [42]).
Simultaneously, these low-diameter topologies offer high diversity of non-minimal paths [42]: 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 [42].

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 schemes such as Equal-Cost Multi-Path (ECMP) [102], 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 [102] or Network Address Aliasing (NAA)) and routing architectures (e.g., PortLand [160] or PARX [69]). 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 cluster computing 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 [70] 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 [212], [96], [216]. Another example is Mellanox, with its Ethernet sales for the 3rd quarter of 2017 being higher than those for InfiniBand [174]. Similar trends are observed in more recent numbers: “Sales of Ethernet adapter products increased 112% year-over-year (...) we are shipping 400 Gbps Ethernet switches” [226]. At the same time, “We saw 27% year-over-year growth in InfiniBand, led by strong demand for our HDR 200 gigabit solutions” [226]. Thus, our analysis can facilitate developing multipath routing in both InfiniBand-based supercomputers but also in a broad landscape of cloud computing infrastructure such as data centers.

In general, we provide the following contributions:

- We provide the first taxonomy of networking architectures and the associated routing protocols, focusing on the offered support for path diversity and multipathing.
- We use our taxonomy to categorize a wide selection of routing designs for data centers and supercomputers.
- We investigate the relationships between support for path diversity and architectural and technological details of different routing protocols.
- We discuss in detail the design of representative protocols.
- We are the first to analyze multipathing schemes related to both supercomputers and the High-Performance Computing (HPC) community (e.g., the InfiniBand stack) and to data centers (e.g., the TCP/IP stack).

**Complementary Analyses** There exist surveys on multipathing [176], [5], [235], [15], [210], [138], [195]. Yet, none focuses on multipathing and path diversity offered by routing protocols in data centers or supercomputers. For example, Lee and Choi describe multipathing in the general Internet and telecommunication networks [135]. Li et al. [138] also focus on the general Internet, covering aspects of multipath transmission related to all TCP/IP stack layers. Similarly, Singh et al. [195] 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 [3], [105]. Next, different works are dedicated to multipathing in sensor networks [15], [5], [176], [235].

Finally, there are analyses of other aspects of data center networking, for example energy efficiency [193], [26], optical interconnects [120], network virtualization [113], [25], overall routing [55], general data center networking with focus on traffic control in TCP [162], low-latency data centers [141], the TCP incast problem [181], bandwidth allocation [56], transport control [229], general data center networking for
Example modern low-diameter topologies

- Slim Fly [2014]: diam. ≤ 2
- Xpander [2016]: diam. ≤ 3
- Dragonfly [2008]: diam. ≤ 3
- Jellyfish [2012]: diam. ≤ 3

The default structure has no well-defined groups (is flat), it can however be arbitrarily modified.

Legend:
- An example of physical packaging
- A certain type of routers
- Routers with many and few ports
- Communicating routers
- Minimal path
- Non-minimal path

Example modern low-diameter topologies

- Slim Fly [2014]: diam. ≤ 2
- Jellyfish [2012]: diam. ≤ 3
- Dragonfly [2008]: diam. ≤ 3

In these topologies, each router connects to a certain number of servers / compute nodes. Routers usually have many servers (endpoints) attached.

Example traditional [high-diameter] topologies

- Mesh
- Torus

Diameter in traditional topologies is usually high.

Routers in traditional topologies usually have few attached endpoints.

Mesh: SF groups are not necessarily fully intra-connected.

In a basic DF variant, routers in a group are fully connected. Still, in general, intra-group cabling pattern may vary.

Every "row" and "column" of routers are fully connected.

An example packaging of routers.

Mesh Torus

Servers (endpoints)

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.

clouds [219], congestion management [116], reconfigurable data center networks [77], and transport protocols [200], [175]. We complement all these works, focusing solely on multipath routing in data centers, high-performance clusters, and supercomputers. 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 [172], [83].

2 Fundamental Notions

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

2.1 Network Model and Parameters

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

2.2 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 [42]). 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) [137] and Clos (CL) [58]. 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. 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.

Modern low-diameter networks are already mentioned in Section 1. The most important are Slim Fly (SF) [34], Dragonfly (DF) [133], Jellyfish (JF) [196], Xpander (XP) [212], and HyperX (Hamming graph) (HX) [4]. There are also numerous other variants of these networks, for example Flexfly [220], Galaxyfly [136], Megafly [73], projective (e.g., OSPF [157]), and others [177], [124]. All these networks have different structure and thus different potential for multipathing [42]; 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 [42].

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

### 2.3 Routing Concepts and Related

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

### 2.4 Routing Schemes

We consider routing schemes (designs) that can be loosely grouped into specific protocols (e.g., OSPF [157]), architectures (e.g., PortLand [160]), and general strategies and techniques (e.g., ECMP [102] or spanning trees [167]). 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.

### 2.5 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).

#### 2.5.1 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 [160].

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 [96]. 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).

#### 2.5.2 InfiniBand

The InfiniBand (IB) architecture is a switched fabric design and is intended for high-performance and system area networks (SAN) deployment scales. Up to 49,151 endpoints (physical or virtual), addressed by a 16bit **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. Contrary to Ethernet, IB is lossless: its design prevents any packets from being dropped due to filled buffers.

InfiniBand natively supports Remote Direct Memory Access (RDMA) and atomic operations. The necessary (for high performance) lossless packet forwarding within IB subnets is realized through link-level, credit-based flow control. 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 [148], [74] 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 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, e.g., TCP/IP via IPoIB, whenever the HPC community deemed it necessary.

2.5.3 Other HPC Network Designs

Cray’s Aries [14] 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 [60] 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 [1] is the next-generation Cray network. It implements a dragonfly topology with fully-connected groups. Slingshot can switch two types of traffic: RDMA over Converged Ethernet (RoCE) [108] (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 [19] 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. The PERCS Hub Chip connects the POWER7 chips within a compute node between themselves and with the rest of the network. The Hub Chip participates to the cache coherence protocol and is able to fetch/inject data directly from the processors’ L3 caches. 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 [46] is a local area massively parallel processor network, designed to connect thousands of small compute nodes. A more recent development, Myrinet Express (MX) [82], provides more functionalities in its network interface cards (NICs). Open-MX [88] is a communication layer that offers the MX API on top of the Ethernet hardware. Quadrics’ QsNet [171], [170] integrates local memories of compute nodes into a single global virtual address space. More recently, Intel has introduced OmniPath [45], an architecture for a tight integration of CPU, memory, and storage units. Many of these architectures featured some form of programmable NICs [46], [170].

2.6 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. § 2.5.1. 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. § 2.5.2.
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 § 2.3. “Stack Layer” indicates the location of each routing scheme in the TCP/IP or InfiniBand stack (cf. § 2.5). SP: A given scheme enables using arbitrary shortest paths. NP: A given scheme enables using arbitrary non-minimal paths. ALB: A given scheme offers adaptive load balancing. AT: A given scheme works with an arbitrary topology. OR: A given scheme does not 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.

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. 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 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 § 2.3) 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 [213]).

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 [16, 202].

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. § 2.2) 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.

Typically, the lookup table is implemented using longest-prefix matching, which allows entries with an identical address prefix and identical output port to be compressed into one table slot. This method is especially well suited to hierarchically organized networks. In general, longest-prefix matching is not required: it is feasible and common to keep uncompressed routing tables, e.g., in Ethernet routing.

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 [66]. 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 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 [117]. 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 [117], or with packet encapsulation (IP-in-IP or MAC-in-MAC) [91], [107], [90]. Source routing can also be achieved to some degree with Multiprotocol Label Switching (MPLS) [183], 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 [183], [225]).

4.3 Minimal Routing Protocols
A common approach to path selection is to only use minimal paths: Paths that are no longer than the shortest path between their endpoints. Minimal paths are preferable 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.

To construct shortest-path routing tables, a variation of the Floyd-Warshall all-pairs shortest path algorithm [76] can be used. Here, besides the shortest-path distance for all router pairs, one also records the out-edge at a given router (i.e., the output port) for the first step of a shortest path to any other router. Other schemes are also applicable, for example an algorithm by Suurballe and Tarjan for finding shortest pairs of edge-disjoint paths [205].

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 [102] routing is an extension of simple destination-based 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 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 [57]: 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 [201] or FastSwitching [236]). Yet, random assignments do not imply uniform load balancing in general, and more advanced schemes such as Weighted Cost Multipathing (WCMP) [230], [233] 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) [198] or Table-based Hashing with Reassignments (THR) [57].

4.5 Spanning Trees (ST)
Another approach to 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 [158]. 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 [202].

4.6 Packet Spraying
A fundamental concept for load balancing is per-packet load balancing. In the basic variant, random packet spraying [65], 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 [65], [96], for example by spraying not single packets but series of packets, such as flowlets [216] or flowcells [98]. 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 [233].

4.7 Virtual LANs (VLANs)
Virtual LANs (VLANs) [143] 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 link in the network. VLANs – in the context of multipathing – primarily address path selection.

4.8 Simple IP Routing
We explicitly distinguish a class of established IP routing protocols, such as OSPF [157] or IS-IS [164]. 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 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. Mapping between identifiers and addresses can be stored in a distributed hash table (DHT) maintained by switches [131] or hosts, or it can be provided by a directory service (e.g., using DNS) [90]. This approach enables more scalable routing [72]. Importantly, it may facilitate multipathing by – for example – maintaining multiple virtual topologies defined by different mappings in DHTs [114].

4.10 Valiant Load Balancing (VLB)
To facilitate non-minimal 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 [213], where this additional information is an arbitrary intermediate router that can be selected at the source endpoint. The routing is divided into two parts: first, the packet is minimally routed to Rs; then, it is minimally routed to the actual destination. VLB has aspects of source routing, namely the choice of Rs 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 path selection (by selecting the intermediate node randomly). This also provides simple, oblivious load balancing.

4.11 Universal Globally-Adaptive Load Balanced (UGAL)
Universal Globally-Adaptive Load balanced (UGAL) [133] is an extension of VLB that enables more advantageous routing decisions in the context of load balancing. 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 [173] or port virtualization via LID mask control (LMC) in InfiniBand [109, 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) [218] or for load balancing the traffic (enhancing performance) [69]. 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 — 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 [93], [222]. Another common scheme is multi-plane topologies, where nodes are connected to a set of disjoint topologies, either similar [92] or different [151]. This is used to increase path diversity and available throughput. However, this increase 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, 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 into 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 2We 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.
### Related to Ethernet and TCP/IP (loosely coupled small clusters and general networks):

| Routing Scheme | Stack Layer | Features of routing schemes | Scheme used | Additional remarks and clarifications |
|----------------|-------------|-----------------------------|------------|---------------------------------------|
| OSPF-OMP (OMP) [217] | L3 | CSPF | OSPF | Cisco's enhancement of OSPF to the multipathing setting. Packets from the same flow are forwarded using the same path. |
| MPA [159] | L3 | CSPF | OSPF | MPA only focuses on algorithms for generating routing paths. |
| SmartBridge [182] | L2 | ST | ST | SmartBridges improves ST; packets are sent between hosts using the shortest possible path in the network. |
| MSTP [16], [63] | L2 | ST+VLAN | ST | Shortest paths are offered only within spanning trees. STAR improves ST; frames are forwarded over alternate paths that are shorter than their corresponding ST path. |
| STAR [146] | L2 | ST | ST | STAR supports mesh networks also in MAN. LSA manages state of links. |
| LSOM [80] | L2 | — | LSOM supports mesh networks also in MAN. LSA manages state of links. |
| AMP [89] | L3 | ECMP, OMP | AMP | Extends ECMP and OSPF-OMP. |
| RBBridges [168] | L2 | — | — | — |
| THR [57] | L3 | ECMP | Table-based Hashing with Reassignments (THR) extends ECMP. It selectively reassigns some active flows based on load sharing statistics. |
| GOE [110] | L2 | ST+VLAN | ST | Shortest paths are offered only within spanning trees. One spanning tree per VLAN is used. Focus on resilience. |
| Viking [190] | L2 | ST+VLAN | ST | Shortest paths are offered only within spanning trees. One spanning tree per VLAN is used. "∗" Viking uses elaborate load balancing, but it is static. Routing in ISP, path are computed offline. |
| TeXCP [121] | L3 | — | — | Load balancing selects paths based on congestion and failures. The scheme focuses on generating paths and on adaptive load balancing. |
| CTMP [198] | L3 | — | — | It extends ECMP. "∗" Path generation is agnostic to the layer. 
| SEATTLE [131] | L2 | L3 | MSP | — |
| SPB [13], TRILL [207] | L2 | L3 | MSP | Multipathing used only for resilience. 
| Ethernet on Air [184] | L2 | L3 | MSP | Multipathing could be enabled by using multiple virtual networks over the same underlying physical topology. |
| VIRO [114] | L2–L3 | L3 | MSP | Not all shortest paths are enabled. "∗" Multipathing only for resilience. |
| MLAG [203], MC-LAG [203] | L3 | L3 | MSP | — |

### Related to Ethernet and TCP/IP (data centers, supercomputers):

| Routing Scheme | Stack Layer | Features of routing schemes | Scheme used | Additional remarks and clarifications |
|----------------|-------------|-----------------------------|------------|---------------------------------------|
| DCell [95] | L2–L3 | L3 | VLB, SR | DCell comes with a specific topology that consists of layers of routers. VLB is used in groups of edge routers. "∗" VLB is used only between border and access routers. |
| Monsoon [91] | L2, L3 | L2–L3 | — | — |
| Work by Al-Fares et al. [7] | L2 | L2–L3 | — | — |
| PortalNet [160] | L2 | CSPF | ECMP | — |
| MOOSE [187] | L2 | ECMP | OSPF-OMP** | — |
| BCube [94] | L2–L3 | L2–L3 | L3 | Only a brief discussion on augmenting the frame format for multipathing. 
| VL2 [90] | L3 | L3 | MSP | Only mentioned as a possible mechanism for multipathing in MOOSE. 
| SPAIN [158] | L2 | L3 | MSP | — |
| Work by Linden et al. [215] | L3 | CSPF | MSP | — |
| Work by Suchara et al. [204] | L3 | CSPF | MSP | — |
| PAST [202] | L2 | CSPF | MSP | — |
| Shadow MACs [2] | L2–L3 | L3 | MSP | — |
| WCMC for DC [233] | L3 | L3 | MSP | — |
| Source routing for flexible DC fabric [117] | L3 | L3 | MSP | — |
| XPath [103] | L3 | L3 | MSP | — |
| Adaptive load balancing* | L3 | L3 | MSP | — |
| ECMP-VLB [123] | L3 | L3 | MSP | — |
| FatPaths [42] | L2–L3 | L3 | MSP | — |

### Related to InfiniBand and other traditionally HPC-related designs (data centers, supercomputers):

| Scheme used | Additional remarks and clarifications |
|-------------|---------------------------------------|
| Shortest path schemes* | — |
| MUD [148], [74] | — |
| LASH-TOR [197] | — |
| Multi-Routing [161] | — |
| Adaptive Routing [154] | — |
| SAR [66] | — |
| PARX [69] | — |

**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. § 2.5). 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 not offer a given feature. ~: A given scheme offers a feature in a limited way. |
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) [16], [63] 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) [57] extends ECMP to a simple form of load balancing; it selectively reassigns some active flows based on load sharing statistics. Global Open Ethernet (GOE) [110], [111] provides virtual private network (VPN) services in metro-area networks (MANs) using Ethernet. Its routing protocol, per-destination multipathing through link aggregation.

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, 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 based on Ethernet & TCP/IP, and 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. § 2.2). 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 [91] 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) [107] so that one may forward a frame via an intermediate switch.

PortLand [160] uses 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, PortLand relies on switches that support MAC-in-MAC tunneling (encapsulation) [107] so that one may forward a frame via an intermediate switch.  

3While 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.
Al-Fares et al. [7] 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 [90] 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 [52], offers round-robin outline them in Section 6 and coarsely summarize them in Table 2. An example design, DRB [52], offers round-robin networks. It (1) divides physical links into layers that form cyclic 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.

### 5.2.2 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 [42]. 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 [158] is an L2 architecture that focuses on using commodity off-the-shelf switches. To enable multipath routing 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 multipath routing 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 [187] addresses the limited scalability of Ethernet; it simply relies on orthogonal designs such as OSPF-OMP for multipathing.

PAST [202] 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 destination 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. [117] discuss encoding arbitrary paths in a data center with OpenFlow to enable flexible fabric, XPath [103] 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. [215] 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 [123] that, as the name suggests, combines the advantages of ECMP and VLB. Finally, FatPaths [42] targets general low-diameter networks. It (1) divides physical links into layers that form cyclic 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 [95], [94]. 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 [153], SSSP [101], Deadlock-Free SSSP (DFSSSP) [68], and a DFSSSP variant called Nue [67].

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

Numerous variations of Multi-Up*/Down* routing have been proposed, e.g., [148], [74], 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 et al. [74] 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, Lynse et al. [148] 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 [197] 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 § 2.2. In preliminary experiments, researchers have tried if the use of different routing algorithms on similar network planes can have an observable performance gain [161]. 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 [154]. 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 [66]. Using NAA with LMC = 1, SAR employs a primary set of shortest paths, calculated with a modified Dijkstra SSSP routing [68], 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 [69], 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 [133]. 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 [171], [170] 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 [45] offers two mechanisms for multipathing between any two endpoints: different paths in the fabric, 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 [46] or Open-MX [88] do not disclose details on their support for multipathing. However, Myrinet does use source routing and works on arbitrary topologies.

6 RELATED ASPECTS OF NETWORKING
Congestion control & load balancing are strongly related to the transport layer (L4). This area was extensively covered in surveys, covering overall networking [150], [50], [180], [147], [112], [221], mobile or ad hoc environments [142], [194], [189], [75], [232], [61], [84], [149], and more recent
cloud and data center networks [223, [192], [206], [231], [224], [20], [85], [191], [162], [181], [229], [219], [116], [77], [200], [175]. Thus, we do not focus on these aspects of networking and we only mention them whenever necessary. However, as they are related to many considered routing schemes, we cite respective works as a reference point for the reader. Many schemes for load balancing and congestion control were proposed in recent years [145], [28], [165], [163], [169], [140], [155], [53], [11], [99], [234], [96], [178], [23], [12], [214], [144], [106], [156], [115], [24], [9], [48], [49], [104], [125], [81], [228]. Such adaptive load balancing can be implemented using flows [59], [179], [188], [211], [29], [233], [8], [119], [102], flowcells (fixed-sized packet series) [98], flowlets (variable-size packet series) [127], [10], [216], [126], [122], and single packets [227], [96], [65], [52], [169], [227], [178], [86]. In data centers, load balancing most 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 [78], [79].

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 [212], [124], [118], [196], [123], [32], [139], [130], [97], [128], [209], [73], [129], [22], [208], [6]. Such works could be used 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 [42] or the PARX routing [69]. 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 [27], linear algebra computations [38], [39], [199], [134], graph processing [35], [32], [41], [44], [37], [87], [31], [40], and other distributed workloads [36], [33], [83] and algorithms [186], [185].

Finally, we expect the growing importance of various schemes enabling programmable routing and transport [18], [54]. Here, one line of research will probably heavily depend on OpenFlow [152] and, especially, P4 [47]. It is also interesting to investigate how to use FPGAs [30], [43], [62] or “smart NICs” [64], [100], [54] 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 principles. 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] Slingshot: The Interconnect for the Exascale Era - Cray Inc. https://www.cray.com/sites/default/files/Slingshot-The-Interconnect-for-the-Exascale-Era.pdf.
[2] K. Agarwal, C. Dixon, E. Rozner, and J. B. Carter. Shadow macs: scalable label-switching for commodity ethernet. In HotSDN’14, pages 157–162, 2014.
[3] S. Aggarwal and P. Mittal. Performance evaluation of single path and multipath regarding bandwidth and delay. Intl. J. Comp. App., 145(9), 2016.
[4] J. H. Ahn, N. Binkert, A. Davis, M. McLaren, and R. S. Schreiber. HyperX: topology, routing, and packaging of efficient large-scale networks. In ACM/IEEE Supercomputing, page 41, 2009.
[5] H. D. E. Al-Ariki and M. S. Swamy. A survey and analysis of multipath routing protocols in wireless multimedia sensor networks. Wireless Networks, 23(6), 2017.
[6] F. Al Faisal, M. H. Rahman, and Y. Inoguchi. A new power efficient high performance interconnection network for many-core processors. Journal of Parallel and Distributed Computing, 101:92–102, 2017.
[7] M. Al-Fares, A. Loukissas, and A. Vahdat. A scalable, commodity data center network architecture. In ACM SIGCOMM, pages 63–74, 2008.
[8] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat. Hedera: Dynamic flow scheduling for data center networks. In NSDI, volume 10, pages 19–19, 2010.
[9] M. Alasmar, G. Parisis, and J. Crowcroft. Polyraptors: embracing path and data redundancy in data centres for efficient data transport. In Proceedings of the ACM SIGCOMM 2018 Conference on Posters and Demos, pages 69–71. ACM, 2018.
[10] M. Alizadeh, T. Edsall, S. Dharmapurikar, R. Vaidyanathan, K. Chu, A. Fingerhut, F. Matus, R. Pan, N. Yadav, G. Varghese, et al. CONGA: Distributed congestion-aware load balancing for datacenters. In Proceedings of the 2014 ACM conference on SIGCOMM, pages 503–514. ACM, 2014.
[11] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan. Data center TCP (DCTCP). ACM SIGCOMM computer communication review, 41(4):63–74, 2011.
[211] F. P. Tso, G. Hamilton, R. Weber, C. Perkins, and D. P. Pezaros. Longer is better: Exploiting path diversity in data center networks. In IEEE 33rd International Conference on Distributed Computing Systems, ICDCS, pages 430–439, 2013.

[212] A. Valadarsky, M. Dinitz, and M. Schapira. Xpander: Unveiling the secrets of high-performance datacenters. In ACM HotNets, 2015.

[213] L. Valiant. A scheme for fast parallel communication. SIAM journal on computing, 11(2):350–361, 1982.

[214] B. Vamanan, J. Hasan, and T. Vijaykumar. Deadline-aware datacenter TCP (D2TCP). ACM SIGCOMM Computer Communication Review, 42(4):115–126, 2012.

[215] S. Van der Linden, G. Detal, and O. Bonaventure. Revisiting next-hop selection in multipath networks. In ACM SIGCOMM CCR, volume 41, 2011.

[216] E. Vanini, R. Pan, M. Alizadeh, P. Taheri, and T. Edsall. Let it flow: Resilient asymmetric load balancing with flowlet switching. In NSDI, pages 407–420, 2017.

[217] C. Villamizar. OSPF optimized multipath (OSPF-OMP). 1999.

[218] A. Vishnu, A. Mamidala, S. Narravula, and D. Panda. Automatic Path Migration over InfiniBand: Early Experiences. In IEEE IPDPS, 2007.

[219] B. Wang, Z. Qi, R. Ma, H. Guan, and A. V. Vasilakos. A survey on data center networking for cloud computing. Computer Networks, 91:528–547, 2015.

[220] K. Wen, P. Samadi, S. Rumley, C. P. Chen, Y. Shen, M. Bahadori, K. Bergman, and J. Wilke. Flexly: Enabling a reconfigurable dragonfly through silicon photonics. In ACM/IEEE Supercomputing, 2016.

[221] J. Widmer, R. Denda, and M. Mauve. A survey on tcp-friendly congestion control. IEEE network, 15(3):28–37, 2001.

[222] N. Wolfe, M. Mubarak, N. Jain, J. Domke, A. Bhaete, C. D. Carothers, and R. B. Ross. Preliminary Performance Analysis of Multi-rail Fat-tree Networks. In IEEE/ACM CCGrid, 2017.

[223] C. Xu, J. Zhao, and G.-M. Muntean. Congestion control design for multipath transport protocols: A survey. IEEE communications surveys & tutorials, 2016.

[224] M. Xu, W. Tian, and R. Buyya. A survey on load balancing algorithms for virtual machines placement in cloud computing. Concurrency and Computation: Practice and Experience, 29(12):e4123, 2017.

[225] X. Xu et al. Unified source routing instructions using mpls label stack. IETF MPLS Working Group draft, Internet Eng. Task Force, 2017.

[226] Yahbo Finance. Mellanox Delivers Record First Quarter 2020 Financial Results. https://finance.yahoo.com/news/mellanox-delivers-record-first-quarter-200500726.htm.

[227] D. Zats, T. Das, P. Mohan, D. Borthakur, and R. H. Katz. Detail: reducing the flow completion time tail in datacenter networks. In ACM SIGCOMM, pages 139–150, 2012.

[228] H. Zhang, J. Zhang, W. Bai, K. Chen, and M. Chowdhury. Resilient datacenter load balancing in the wild. In Proceedings of the Conference of the ACM Special Interest Group on Data Communication, pages 253–266, 2017.

[229] J. Zhang, F. Ren, and C. Lin. Survey on transport control in data center networks. IEEE Network, 27(4):22–26, 2013.

[230] J. Zhang, K. Xi, L. Zhang, and H. J. Chao. Optimizing network performance using weighted multipath routing. In IEEE ICCCN, 2012.

[231] J. Zhang, F. R. Yu, S. Wang, T. Huang, Z. Liu, and Y. Liu. Load balancing in data center networks: A survey. IEEE Communications Surveys & Tutorials, 20(3):2324–2352, 2018.

[232] J. Zhao, L. Wang, S. Li, X. Liu, Z. Yuan, and Z. Gao. A survey of congestion control mechanisms in wireless sensor networks. In 2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pages 719–722. IEEE, 2010.

[233] J. Zhou, M. Tewari, M. Zhu, A. Kabbani, L. Poutievski, A. Singh, and A. Vahdat. Wcmp: Weighted cost multipathing for improved fairness in data centers. In ACM EuroSys, 2014.

[234] D. Zhuo, Q. Zhang, V. Liu, A. Krishnamurthy, and T. Anderson. Rack-level congestion control. In Proceedings of the 15th ACM Workshop on Hot Topics in Networks, pages 148–154. ACM, 2016.

[235] S. M. Zin, N. B. Anuar, M. L. M. Kiah, and I. Ahmedy. Survey of secure multipath routing protocols for wsns. Journal of Network and Computer Applications, 55:123–153, 2015.

[236] A. Zinin and I. Cisco. Routing: Packet forwarding and intradomain routing protocols, 2002.