Survey of Reconfigurable Data Center Networks: Enablers, Algorithms, Complexity

Klaus-Tycho Foerster
Faculty of Computer Science
University of Vienna, Austria
klaus-tycho.foerster@univie.ac.at

Stefan Schmid
Faculty of Computer Science
University of Vienna, Austria
stefan_schmid@univie.ac.at

Abstract

Emerging optical technologies introduce opportunities to reconfigure network topologies at runtime. The resulting topological flexibilities can be exploited to design novel demand-aware and self-adjusting networks. This paper provides an overview of the algorithmic problems introduced by this technology, and surveys first solutions.

1 Introduction

Communication networks have become a critical infrastructure of our digital society. This introduces increasingly stringent requirements on the dependability and performance of such networks. At the same time, network traffic is growing explosively, a trend which is likely to continue [71]: next-generation workloads, such as (distributed) machine learning and artificial intelligence will lead to additional workloads headed for the world's data centers. Indeed, network traffic is growing particularly fast in data centers [83], also because many applications generate much internal data center traffic: the traffic staying inside the data center is often much larger than the traffic entering or leaving the data center [71].

Motivated by these trends, the networking community has recently created massive efforts to the design more efficient data center architectures, developing new scheduling [4], load-balancing [40], monitoring [68], and congestion control [27] algorithms, to just name a few. A particularly active and interesting area is the design of new data center architectures, exploring alternatives to current (multi-rooted) fat-tree networks [2]: from alternative fat-trees [58] to fat-free [86] networks, and from trees to hypercubes [43], random graphs [88], or expanders [53]. All these designs aim to make data center networks more efficient, in terms of performance, cost, and cabling. The situation has recently been compared to the early 1980s, when the emergence of new applications led to many

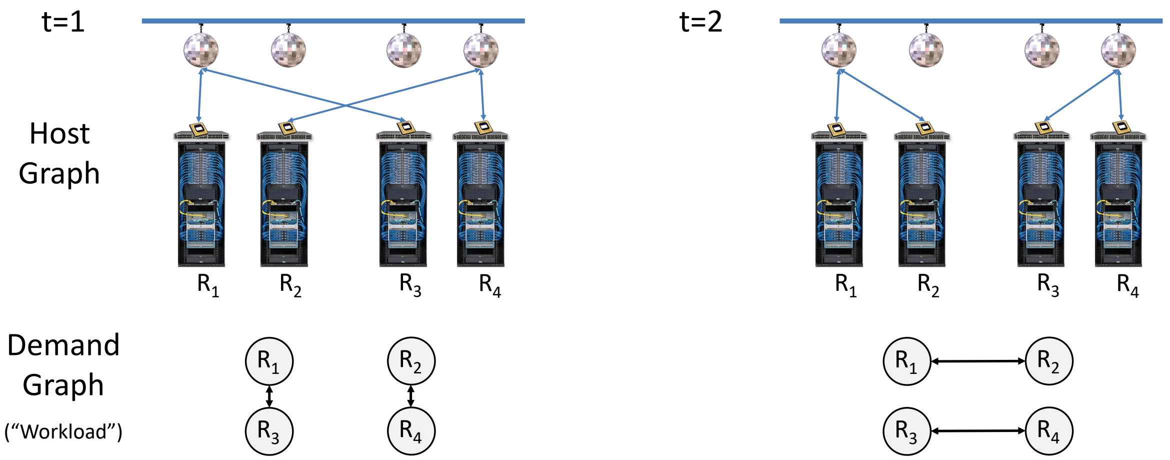

Figure 1: Illustrating example for demand-aware reconfigurable networks. Each node (here: a rack resp. top-of-rack switch) $R_1, \dots, R_4$ can create a direct connection to another node via mirrors or "disco-balls" on the ceiling. Due to hardware constraints, only one such connection can exist for each node per epoch, i.e., the reconfigurable links form a matching. On the left side, at time $t = 1$, there are demands between the nodes $R_1 \leftrightarrow R_3$ and $R_2 \leftrightarrow R_4$, respectively. As such, we create direct connections between them, as seen in the physical network above it. When the demand graph changes at time $t = 2$, the network can adapt to the new workload, by creating direct connections between $R_1 \leftrightarrow R_2$ and $R_3 \leftrightarrow R_4$, respectively. However, when the demand graph no longer forms a matching, or when physical constraints prevent some reconfigurable links, the situation becomes more complex.

proposals for new interconnection network designs for parallel computers [1]. However, all these architectures also have in common that their designs are demand-oblivious and fixed.

Recent technological innovations introduce a radically new dimension to the data center network optimization problem: the possibility to reconfigure the data center topology at runtime. This technology hence has the potential to introduce a paradigm shift: it enables a vision of demand-aware data center networks which (self-)adjust to the their workload (i.e., communication pattern) [12].

While the technology of such reconfigurable networks is evolving at a fast pace, these networks lack theoretical foundations: models, metrics, and algorithms. We have fallen behind the curve.

The objective of this paper is to help bridge this gap, by raising awareness in the theory community about a rich and potentially impactful research area. We first briefly discuss technological enablers and report on motivating empirical studies. Our main focus in this paper then is on the new models and algorithmic challenges introduced by this field. In particular, we will review existing algorithms and complexity results, and highlight future research directions.

The remainder of this article is structured as follows: We first give an overview of the technological enablers in §2.1, along with the empirical motivation for reconfigurable data centers in §2.2, followed by a formal model overview in §3. We then survey different algorithmic approaches for reconfigurable data center topologies in §4, also pointing out opportunities and underlying complexities. A selected (algorithmic) timelime is shown in Table 1. Lastly, we conclude in §5.

Table 1 Selected timeline of reconfigurable data centers

2009	– *Flyways* [51]: Steerable antennas (narrow beamwidth at 60 GHz [78]) to serve hotspots
2010	– *Helios* [33]/*c-Through* [98, 99]: Hybrid switch architecture, maximum matching (Edmond's algorithm [30]), single-hop reconfigurable connections ($O(10)ms$ reconfiguration time).
	– *Proteus* [21, 89]: k reconfigurable connections per ToR, multi-hop path stitching, multi-hop reconfigurable connections (weighted b-matching [69], edge-exchanges for connectivity [72], wavelength assignment via edge-coloring [67] on multigraphs)
2011	– Extension of *Flyways* [51] to better handle practical concerns such as stability and interference for 60GHz links, along with greedy heuristics for dynamic link placement [45]
2012	– *Mirror Mirror on the ceiling* [106]: 3D-beamforming (60 Ghz wireless), signals bounce off the ceiling
2013	– *Mordia* [31, 32, 77]: Traffic matrix scheduling, matrix decomposition (Birkhoff-von-Neumann (BvN) [18, 97]), fiber ring structure with wavelengths ($O(10)\mu s$ reconfiguration time)
	– *SplayNets* [6, 76, 82]: Fine-grained and online reconfigurations in the spirit of self-adjusting datastructures (all links are reconfigurable), aiming to strike a balance between short route lengths and reconfiguration costs
2014	– *REACToR* [56]: Buffer burst of packets at end-hosts until circuit provisioned, employs [77]
	– *Firefly* [14] Combination of Free Space Optics and Galvo/switchable mirrors (small fan-out)
2015	– *Solstice* [57]: Greedy perfect matching based hybrid scheduling heuristic that outperforms BvN [77]
	– Designs for optical switches with a reconfiguration latency of $O(10)ns$ [3]
2016	– *ProjecToR* [39]: Distributed Free Space Optics with digital micromirrors (high fan-out) [38] (Stable Matching [26]), goal of (starvation-free) low latency
	– *Eclipse* [95, 96]: $(1 - 1/e^{(1-\varepsilon)})$-approximation for throughput in traffic matrix scheduling (single-hop reconfigurable connections, hybrid switch architecture), outperforms heuristics in [57]
2017	– *DAN* [7, 8, 11, 12]: Demand-aware networks based on reconfigurable links only and optimized for a demand snapshot, to minimized average route length and/or minimize load
	– *MegaSwitch* [23]: Non-blocking circuits over multiple fiber rings (stacking rings in [77] doesn't suffice)
	– *Rotornet* [63]: Oblivious cyclical reconfiguration w. selector switches [64] (Valiant load balancing [94])
	– *Tale of Two Topologies* [105]: Convert locally between Clos [24] topology and random graphs [87, 88]
2018	– *DeepConf* [81]/*xWeaver* [102]: Machine learning approaches for topology reconfiguration
2019	– Complexity classifications for weighted average path lengths in reconfigurable topologies [34, 35, 36]
	– *ReNet* [13] and *Push-Down-Trees* [9] providing statically and dynamically optimal reconfigurations
	– *DisSplayNets* [75]: fully decentralized *SplayNets*
	– *Opera* [60]: Maintaining expander-based topologies under (oblivious) reconfiguration

2 Technological Enablers and Empirical Motivation

Reconfigurable data center networks are enabled by emerging technologies and motivated by the rich structure often observed in communication patterns. In the following, we briefly review these technological enablers and the empirical motivations.

2.1 Technological Enablers

Traditionally, data center networks are based on electrical elements[1], and optimized toward static properties such as scalability, construction cost, capacity, cabling complexity, latency, and/or robustness [103]. The introduction of *optical circuit switching* introduces an opportunity to save cost and improve performance through *reconfigurations*. To this end, it is important that a reconfigurable network is (1) *agile*, i.e., provides low *reconfiguration time* and (2) supports a high *fan-out*, a large number of different nodes can be connected at any given time as well as over time. Most emerging reconfigurable data center topologies leveraging optical circuit switching are *hybrid* in the sense that they combine optical circuit switching with conventional electrical packet switching. As we discuss later, reconfigurable topologies can also be implemented by wireless technologies and free-space optics, but also via electric solutions [5, 20]. For example, optical circuit switching is used to dynamically interconnect only (a subset of) top-of-rack switches and reserved for elephant flows. An additional conventional topology is commonly used for the remaining traffic (e.g., mice flows). Mice flows can also be routed along the reconfigurable topology, e.g., under extremely fast reconfiguration times [32] or when the reconfigurable topology always forms a connected multi-hop network [21], simulating a static topology by priority queuing [60].[2]

Different reconfigurable networks differ by the type of reconfigurable devices used. We refer to [19, 39, 46, 65, 92, 103] for a further overview, as detailing all technologies, such as e.g. thermo-optic switches [54], goes far beyond the scope of this paper. The three most common technologies allowing to dynamically change the capacity between pairs of ToRs[3] are:

1. **Optical Circuit Switches (OCS):** Examples employing OCSes include *Helios* [33], *REACToR* [56], *Solstice* [57], *Mordia* [77], *OSA* [21], *c-Through* [99], among others. To provide some intuition, an OCS can be understood as "*a Layer 0 switch — it operates directly on light beams without decoding any packets*" [33]. There are multiple technological enablers in this setting. Micro-Electro-Mechanical System (MEMS) switches employ small mirrors controlled by motors, typically reconfiguring in $O(10)ms$ in a $N \times N$ crossbar 3D-MEMS setting. While these mirrors can be redesigned to react in smaller timescales, there is a tradeoff regarding the number of ports, see [62] for a discussion on the scaling limits. A Wavelength-Selective Switch (WSS) is usually a $1 \times N$ switch that distributes the incoming wavelengths over the N output ports, with an extra bypass port for the remaining wavelengths. Such WSSes can be built with faster 2D-MEMS, at the cost of having smaller port counts. A related enabler is Wavelength Division Multiplexing (WDM), where typically 40 or more wavelengths can be used on a single fiber [21]. For example the *Mordia* [77] prototype achieves a $O(10)\mu s$ reconfiguration time with 24 ports. However, such switches can also be combined and extended, see e.g. [23, 3.1] for a discussion on how to reach $O(100k)$ ports. Another idea to increase the port count while remaining at a $O(10)\mu s$ reconfiguration time is to reduce the number of different combinations the switch can serve [64]. WSSes can also be realized without 2D-MEMS, see e.g. *WaveCube* [22] with a reconfiguration time of ∼$10ms$. Lastly, technology with a reconfiguration time in the order of nanoseconds is also emerging, e.g. [3], we refer to the various references in [92] for further technology proposals and details.

[1] However, "*optical fibers are gaining the momentum with high data rates, low transmission loss, and low power consumption*" [103]. Such optical fibers do not necessarily imply a reconfigurable network, they can also be used as fixed links.

[2] Notwithstanding, "*Supporting low-latency traffic without a hybrid network is a subject of ongoing investigation*" [65].

[3] In one of the earlier papers, extra links between pairs of ToRs were called *flyways* [51].

2. **60 GHz Wireless:** 60 GHz technology can support short range (1-10 meters), high-bandwidth wireless links in an unlicensed band. Examples include *Flyways* [45, 51] and the 3D-beamforming approach by Zhou et al. [106]. In order to extend the "line-of-sight" and reduce interference, mirrors can be used, e.g., on the ceiling [106]. Regarding the reconfiguration time, *Flyways* [45, §3.1] mentions that antennas can be steered in $O(100)\mu$s, but ignores such overheads in their simulations. Ranges from 0.01 to 1 second are reported in [106, §4.1]. In order to overcome mechanical steering delays, antenna arrays can be used, trading in a lower fan-out and greater interference for smaller delays in the range of $O(10)ns$, see [106, §6] for a discussion.

3. **Free-Space Optics (FSO):** FSO is another emerging technology which usually relies on *lasers* to enable high-capacity communication between nodes (e.g., top-of-rack switches) through the "free space". In general, FSO is expected to offer lower interference and higher bandwidth over long ranges than 60 GHz communication. Similar to wireless links, it is e.g. known that the mechanical steering of FSO links (e.g., using vertical and rotational motion) [79] or the use of Galvo/switchable mirrors [14] results in relatively high switching times (e.g. ~$20ms$ in *FireFly* [14]) compared to the Digital Micromirror Devices (DMD) used e.g., in *ProjecToR* [39] $(7-12\mu s)$; DMD-based solutions such as *ProjecToR* also increase the fan-out and can *scale* better than e.g. MEMS-based solutions [61].

2.2 Empirical Motivation

There exist several empirical studies demonstrating the potential of reconfigurable networks. For the following discussion, we classify reconfigurable networks along the following dimension:

- **Demand-Oblivious Networks:** Demand-oblivious reconfigurable networks such as *Rotornet* [63] and *Opera* [60] change the topology according to a *fixed* schedule which is *independent* of the workload.
- **Demand-Aware Networks:** The topology of demand-aware networks is adjusted to optimally serve the current traffic pattern.

A key advantage of demand-oblivious reconfigurable networks is their simplicity and high-degree of decentralization: empirical studies show that solutions like *Rotornet* [63] and *Opera* [60] can provide high capacity (emulating a "complete graph") without sacrificing scalability, at the price of a small latency cost.

Demand-aware networks, in contrast, can also leverage *structure in the demand* (i.e., traffic pattern or workload): the usefulness of demand-aware and self-adjusting networks hence depends on the amount of exploitable structure there is in the demand.[4]

Existing measurement studies show that real-world communication patterns are indeed often far from all-to-all: depending on the application mix in the data center [80], traffic patterns feature spatial and temporal structure, i.e., are sparse and only a small fraction of all possible source-destination pairs are involved in intensive communications at any time [80]. For example, empirical studies in data centers found that in a given time interval, a high percentage of rack pairs does not exchange any traffic at all, and that less than 1% of the rack pairs account for 80% of the total

[4] We note that the problem of actually obtaining the exact current demands is not trivial, unless they are e.g. provided by a central scheduling system. Notwithstanding, the situation is not as grim as it may seem, as even "*a presumed lack of advance knowledge of flow sizes is not necessarily prohibitive for highly efficient scheduling*" [28]. We refer to e.g. [77, §3.4] for an introductory overview of demand estimation.

traffic [39]. Data center traffic further exhibits regionality and some stability [21]: only a few ToRs are hot and most of their traffic goes to a few other ToRs [51, 52], over 90% bytes flow in elephant flows [42], traffic at ToRs exhibits an ON/OFF pattern [16], 60% ToRs see less than 20% change in traffic volume for between 1.6-2.2 seconds [17], and a production DCN traffic shows stability even on a hourly time scale [108]. Such empirical studies hence indicate that demand-aware reconfigurable networking technology may lower costs without affecting performance [14].

3 Models and Taxonomy

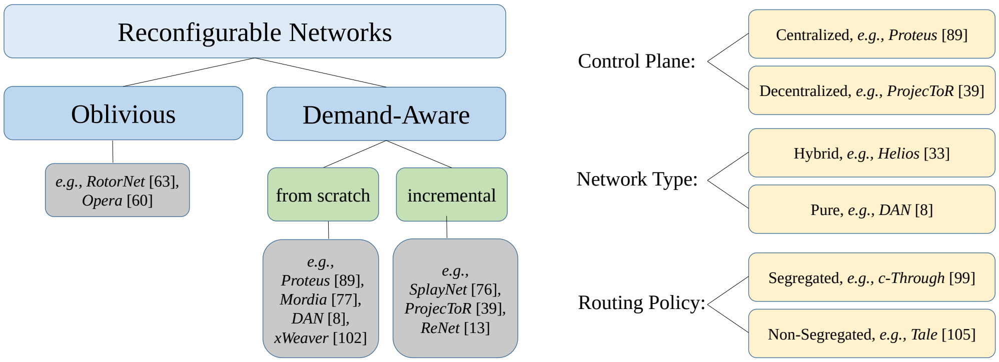
Reconfigurable Networks
Oblivious
Demand-Aware
e.g., RotorNet [63], *Opera* [60]
from scratch
incremental
e.g., Proteus [89], *Mordia* [77], *DAN* [8], *xWeaver* [102]
e.g., SplayNet [76], *ProjecToR* [39], *ReNet* [13]
Control Plane:
Centralized, *e.g., Proteus* [89]
Decentralized, *e.g., ProjecToR* [39]
Network Type:
Hybrid, *e.g., Helios* [33]
Pure, *e.g., DAN* [8]
Routing Policy:
Segregated, *e.g., c-Through* [99]
Non-Segregated, *e.g., Tale* [105]

Figure 2: Taxonomy overview for reconfigurable data centers, with a selected example for each category. On the left side, we classify the different types of reconfigurable networks and give examples. On the right side, we list additional dimensions along which existing models differ: related to the control plane, whether the physical network is hybrid or not, related to routing, etc. We omit the dimension of different objective functions for clarity.

Essentially, reconfigurable data center networks pose the following graph theoretic problem: how to augment the topology with additional edges, s.t. some desired objective function is optimized? However, unlike in classic graph augmentation [25, 66, 73, 74], these edges cannot be added[5] arbitrarily. Rather, the edge additions are subject to real-world constraints such as degree (e.g. a node only has one physical receiver) or connectivity (e.g. distance or objects blocking the signal path). In some shared mediums, coloring algorithms for different wavelengths also come into play.

It's a Match(ing)! In the most basic case, the network nodes are connected to a reconfigurable (e.g. optical circuit) switch, which creates direct physical connections between the nodes. However, these connections can only be formed between two nodes each time, i.e., more formally, the output is a matching (Fig. 1). In turn, matching links are added to the topology, augmenting the network to better serve the demands. While such an augmentation should take the fixed static topology into account, there are also various proposals that completely omit non-reconfigurable links.

Moreover, reconfigurable networks can offer great flexibility beyond a single reconfigurable switch. 1) Multiple such switches can be added, also interconnecting them, where each switch might only connect a subset of nodes, 2) nodes can also be connected multiple times to a single switch, allowing for multi-matchings, 3) reconfigurable links can be uni- or bidirectional, depending on technology and policy, and 4) the reconfigurable switches might only allow a subset of all possi-

[5] In some settings, it can also be interesting to study the *removal* of edges, see e.g. [107, §5.1].

ble matchings, to increase the reconfiguration time. Nonetheless, the fundamental problem setting remains unchanged: Given a static graph e.g., $G = (V, E)$, how to augment this graph with some matchings (subject to various constraints) s.t. some objective function is optimized?

Lastly, there is a multitude of technologies that implements the above settings, most notably optical circuit switching, wireless (beamforming), and free-space optics, but also shared mediums such as fiber ring structures and completely electric solutions. In some settings, e.g., fiber rings and wireless, it is also possibly to dynamically create one-to-many connections. If the transmission medium is shared, additional constraints can come into play, such as interference between signals (channels for non-beamforming wireless) and different colors for different wavelengths on fiber.

Routing Part I: From Snapshots to Ongoing Reconfiguration. Routing the data itself adds another layer of complexity to reconfigurable networks, as the control plane has to distribute the new routing paths that are enabled by the topology reconfiguration. An orthogonal approach is to periodically schedule the topology changes ahead of time, s.t. a calendar is available to all nodes (i.e., being demand-oblivious). In such a setting, it can also be advantageous if some traffic can be buffered, in order to send it when a fitting topology appears.

To be more demand-aware, a first idea is to optimize the networks for snapshots, i.e., the network topology is reconfigured only once for some demands or communication pattern. Such a reconfiguration can then be scheduled at periodic intervals, to adapt the network to current settings. Improved hardware for faster reconfiguration motivated the idea of changing the topology multiple times for a single snapshot, coined traffic matrix scheduling. Under traffic matrix scheduling, the logically centralized control plane can bundle the upcoming changes, instead of multiple interactions. Moreover, e.g. the throughput can be improved over a single reconfiguration.

For better scaling and faster response times, the control plane can also be distributed, where the nodes run some distributed protocol to arrange for the routing (and ideally the topology reconfiguration as well). Beyond snapshots, one can also consider request-based models, where the topology is adapted online for each incoming (transfer) request.

Routing Part II: Routing Policies. Whereas multi-hop routing is standard in most computer networks, the situation is not so clear in reconfigurable networks. Enforcing that reconfigurable connections may only be used along a single hop has the advantage that there are no dependencies on other links, i.e., links can be reconfigured independently, as long as the matching is valid. Along the same lines, policies restricted to single-hop connections are easier to analyze and optimize. Notwithstanding, enforcing single-hop connections is usually an artificial constraint, which degrades the maximum theoretical system performance by limiting the routing options. On the other hand, the system performance also suffers if topology and routing changes take unreasonably long to compute and distribute.

Optimization Objectives. Ideally, a networked system should be optimized for a multitude of objectives, in particular throughput maximization (each edge has some capacity), low latency respectively weighted average path lengths (edges have weights), and optimized flow completion times, as well as starvation and fairness issues. In practice, most approaches primarily optimize their algorithms for a single goal, implicitly optimizing the other objectives as well.

4 Algorithmic Opportunities and Complexity

We now investigate different classes of algorithms used to reconfigure data center networks in various approaches. Besides (intractable) mixed integer programs and (greedy) heuristics, we can roughly categorize most proposed algorithms into those motivated by matchings (§4.1) and those inspired by datastructures and coding (§4.2). Further approaches are briefly discussed in §4.3.

4.1 Matching Algorithms

As reconfiguring network topologies, at its core, is a matching problem (except for contention in shared mediums), it is not surprising that a variety of matching algorithms are studied in the literature.

Maximum Matching Algorithms. The motivation behind maximum matching algorithms, as employed by, e.g., *Helios* [33] and *c-Through* [99], is twofold. First, matching algorithms are usually fast, e.g. Edmond's algorithm [30]. Second, if the network may only be reconfigured once for a snapshot and the only objective is to maximize single-hop throughput along reconfigurable links, then the throughput is optimal: every possible link can be assigned a benefit, where Edmond's algorithm [30] optimizes the sum of benefits. However, when accounting for the static topology as well, the performance deteriorates, and such settings are mostly unstudied from a theoretical perspective. Multiple reconfigurations for a single snapshot are discussed in the next paragraph.

Proteus [89] and its extension *OSA* [21] employ an analogous idea for multiple reconfigurable connections per node by employing weighted b-matching [69] algorithms.[6] In order to create a connected graph for multi-hop routing, *OSA* [21] then leverages edge-exchange [72] operations. Notwithstanding, no theoretical guarantees for the performance of multi-hop routing are provided.

For minimizing the average weighted path length, maximum (b-)matching algorithms can be directly adapted to yield optimal results, for the case of segregated routing policies (single-hop reconfigurable xor multi-hop static) [35, 36]. Non-segregated routing policies (joint routing on reconfigurable and static parts, along multiple hops) turn the problem NP-hard again [35, 36], only heuristics without approximation guarantees are known so far [14, §5.1], [34].

Traffic Matrix Scheduling. Motivated by faster hardware reconfiguration times, *Mordia* [77] proposed to reconfigure the network multiple times for a single (traffic demand) snapshot. To this end, the traffic demand matrix is scaled into a bandwidth allocation matrix, which represents the fraction of bandwidth every possible matching edge should be allocated in an ideal schedule. Next, the allocation matrix is decomposed into a schedule, employing a computationally efficient [41] Birkhoff-von-Neumann decomposition, resulting in $O(n^2)$ reconfigurations and durations. However, *Mordia* [77] did not account for the static part of the network (hybrid architectures) and also did not minimize the number of reconfigurations, each still inducing some delay.

Solstice [57] takes hybrid architectures and time constraints into account (which turn the problem NP-hard [55]) but does not provide theoretical guarantees for their greedy scheduling heuristic. A further literature overview for different reconfiguration delays is found in [57, §6] as well.

Eclipse [96] can provide an $(1 - 1/e^{(1-\varepsilon)})$-approximation for throughput in the hybrid switch architecture with reconfiguration delay, but only for direct routing along single-hop reconfigurable connections. The indirect routing (multi-hop) case remains open w.r.t. provable guarantees.

Stable Matching Algorithms. A significant downside of the previously in this section discussed algorithms is that they rely on centralized computation and coordination. Stable matching algorithms have the benefit that it is easier to implement them in a distributed fashion, also allowing for solutions that are incremental, i.e., to keep the current topology state in mind. *ProjecToR* [39] employs (Gale-Shapley [37]) stable matching algorithms with the goal of minimizing (the ℓ_2 norm

[6] Regarding the quality of such solutions, please see the paragraph on stable matching algorithms below.

of) latencies, accounting for starvation via aging of requests. In their setting, reconfigurable connections are single-hop, with multiple senders and receivers for a subset of nodes, and control packets are exchanged over the static topology. Even though their algorithm performs online computations, they can achieve a constant-factor approximation for latencies.

Instantaneous throughput optimization in this setting would be computationally easy (under centralized computation), if each node can match to each other node at most once. When optimizing for path lengths, we can assume this constraint, as parallel edges do not decrease distances. For throughput however, the situation is unclear, and it is not known if the problem is NP-hard [26].

Oblivious Matching Schedules. An orthogonal approach to centralized versus distributed reconfiguration is to be completely oblivious (cf also Figure 2) to current demands and pre-compute a *fixed* periodic scheduling for topology reconfigurations. One idea could be to cycle through all possible sets of matchings, which however would take an unreasonably long time. Instead, *Rotornet* [63] proposes to just use a small set of matchings, s.t. connectivity between endpoints is guaranteed in a matching cycle. In this setting, the reconfiguration time of switches can also be improved, as the switches only need to implement a small subset of all possible matchings. In case of uniform (delay-tolerant) traffic, such single-hop forwarding can saturate the network's bisection bandwidth [63]. Still, for skewed traffic matrices, many direct connections will remain underutilized. To this end, *Rotornet* [63] uses Valiant load balancing [94] and multiple reconfigurable switches to perform distributed multi-hop routing, coupled with buffering of such indirect traffic.

In the context of purely static topologies, building the network topology itself can also be understood as an oblivious reconfiguration—just only once. As common networking technology has a small number of ports (often identical throughout the data center), *Jellyfish* [88] proposes to use random regular graphs to obtain good throughput, leveraging low average path length and the ability for route traffic through underutilized network parts for skewed traffic matrices. The deterministic version of this idea relies on d-regular expander graphs, which are only a logarithmic throughput factor away from d-regular graphs built for specific traffic matrices [93]. A proposal how to realize such expanders as a fixed data center topology was presented in [53].

These insights for random and expander graphs also found their way into reconfigurable data centers shortly after. In a *Tale of Two Topologies* [105], the topology is reconfigured to locally convert between Clos and random graphs, though in a demand-aware fashion. Moreover, *Opera* [60] extends the ideas of *Rotornet* [63] by maintain expander graphs in its periodic reconfigurations. Even though the reconfiguration scheduling of *Opera* is deterministic and oblivious, the precomputation of the topology layouts is in its current form still randomized.

4.2 Datastructure and Coding Approaches

Another approach to design demand-aware reconfigurable networks leverages an interesting connection to datastructures and coding. Existing datastructure- and coding-based approaches fall into two categories (according to our more general taxonomy in Figure 2):

- **Fixed demand-aware networks:** These networks (e.g., [7, 8, 11, 12]) are optimized toward a given snapshot of the demand. The objective is usually to minimize the expected path length [7, 8, 11, 12] but there also examples additionally minimizing the network load [11] or resilience [7].

- **Self-adjusting demand-aware networks:** These networks (e.g., [6, 12, 75, 76, 82]) adjust in a fine-grained manner, trying to react *quickly and locally* two new communication requests.

The objective is to strike an optimal tradeoff between the benefits of reconfigurations (e.g., shorter routes) and their costs (e.g., reconfiguration latency, energy, packet reorderings, etc.)

The main observation underlying these approaches is that designing an "optimal" reconfigurable network for a *single* source is related to the design of Binary Search Trees (BSTs) or Huffman coding. We briefly elaborate on this connection by discussing the fixed and the self-adjusting problems in turn:

- *Fixed:* If the distribution of which keys are accessed more frequently in a BST is known, it is possible to optimize the BST towards its demand: i.e., to compute a *biased* BST. Similarly, if the frequency distribution of which letters need to be communicated is known, it is possible to optimize the encoding of those letter, e.g., using a Huffman tree: the expected number of to-be-communicated bits is reduced.

- *Self-adjusting:* Even if the demand is not known ahead of time, it is possible to adjust a binary search tree or an encoding *in an online manner*, e.g., using *splay trees* (in case of BSTs) or *dynamic Huffman trees* (in case of coding).

The idea is then to organize the communication partners (i.e., the destinations) of a *given* communication source in either a static binary search or Huffman tree (if the demand is known), or in a dynamic tree (if the demand is not known or if the distribution changes over time). The former approach is used in *DANs* [8, 11, 12], the latter approach is used in *SplayNets* [6, 12, 75, 76, 82], in *Push-Down-Trees* [10], and in *ReNets* [13]. The tree optimized for a single source is sometimes called the *ego-tree* [11], and the approach relies on combining these ego-trees of the different sources into a network, while keeping the resulting node degree constant and preserving distances (i.e., low distortion). One exception to this general approach is *rDAN* [7], which relies on Shannon-Fano-Elias coding and a continuous-discrete approach [70].

Besides algorithms to design demand-aware reconfigurable networks, the connection to datastructures and coding also provides *metrics* for demand-aware networks. Recall that in demand-oblivious BSTs, the average access cost is $O(\log n)$, where n is the size of the BST; similarly, worst-case coding requires $O(\log n)$ bits per transmitted letter. In contrast, the expected lookup cost in *biased* BSTs or the expected number of bits per symbol in a Huffman tree, is proportional to the *entropy* of the demand, which can be much lower. Similarly, it has recently been shown that (a variant of) the entropy of the demand is also a useful metric for the performance achievable by a reconfigurable network.

4.3 Further Approaches

The observation that traffic demands feature much structure naturally leads to the question whether this structure could also be exploited using machine learning. To just give two examples, *xWeaver* [102] and *DeepConf* [81] use neural networks to provide traffic-driven topology adaptation. Another approach is takens by Kalmbach et al. [50], who aim to strike a balance between topology optimization and "keeping flexibilities", leveraging the concept of *empowerment* (e.g., known from robotics).

In the context of shared mediums (e.g., non-beamformed wireless broadcast, fiber[7] (rings)), contention and interference of signals can be avoided by using different channels and wavelengths. The algorithmic challenge is then to find (optimal) edge-colorings on multi-graphs, an NP-hard

[7] In the context of data center proposals, shared fiber is the more common medium, e.g., in [21, 23, 77].

problem for which fast heuristics exist [67]. However, on specialized topologies optimal solutions can be found in polynomial time, e.g., in *WaveCube* [22]. Shared mediums also have the benefit that it is easier to distribute data in a one-to-many setting [100]. For example on fiber rings, all nodes on the ring can intercept the signal [23, §3.1]. One-to-many paradigms[8] such as multicast can also be implemented in other technologies, using e.g., optical splitters for optical circuit switches or half-reflection mirrors for free-space optics [15, 90, 91, 104]. The challenges posed by one-to-one transfers carry over to this setting, in particular results with provable (approximation) guarantees on reconfigurable multi-hop routing would be of interest.

5 Conclusion and Future Work

We presented a survey of emerging reconfigurable data centers, discussing technological enablers and empirical studies in the literature, and with an emphasis on the algorithmic problems such data centers introduce. While we deliberately omitted many details, especially on the technological enablers of demand-aware networks, we hope that our paper can serve the community as a motivation to dig deeper in the referenced literature, and contribute toward more refined models and improved algorithmic techniques, accordingly. Finally, we also note that reconfigurable networks are not only arising in data centers, but for example also in the wide-area [29, 44, 48, 49, 59, 84, 85], introducing further interesting avenues for research.

References

[1] A. Akella, T. Benson, B. Chandrasekaran, C. Huang, B. Maggs, and D. Maltz. A universal approach to data center network design. In *Proceedings of the 2015 International Conference on Distributed Computing and Networking*, page 41. ACM, 2015.

[2] M. Al-Fares, A. Loukissas, and A. Vahdat. A scalable, commodity data center network architecture. In *SIGCOMM*. ACM, 2008.

[3] D. Alistarh, H. Ballani, P. Costa, A. Funnell, J. Benjamin, P. M. Watts, and B. Thomsen. A high-radix, low-latency optical switch for data centers. *Computer Communication Review*, 45(5):367–368, 2015.

[4] M. Alizadeh, S. Yang, M. Sharif, S. Katti, N. McKeown, B. Prabhakar, and S. Shenker. pfabric: Minimal near-optimal datacenter transport. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, New York, NY, USA, 2013. ACM.

[5] Arista. White Paper: A layman's guide to Layer 1 Switching. https://www.arista.com/assets/data/pdf/Whitepapers/Laymans-Guide-White-Paper.pdf, Dec. 2018.

[6] C. Avin, B. Haeupler, Z. Lotker, C. Scheideler, and S. Schmid. Locally self-adjusting tree networks. In *Proc. 27th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, May 2013.

[7] C. Avin, A. Hercules, A. Loukas, and S. Schmid. rdan: Toward robust demand-aware network designs. In *Information Processing Letters (IPL)*, 2018.

[8] C. Avin, K. Mondal, and S. Schmid. Demand-aware network designs of bounded degree. In *DISC*, 2017.

[9] C. Avin, K. Mondal, and S. Schmid. Push-down trees: Optimal self-adjusting complete trees. In *arXiv*, 2018.

[10] C. Avin, K. Mondal, and S. Schmid. Push-down trees: Optimal self-adjusting complete trees. *CoRR*, abs/1807.04613v1, 2018.

[8] Conceptually similar challenges also appear in the context of coflows [47, 101].

[11] C. Avin, K. Mondal, and S. Schmid. Demand-aware network design with minimal congestion and route lengths. In *Proc. IEE INFOCOM*, 2019.

[12] C. Avin and S. Schmid. Toward demand-aware networking: A theory for self-adjusting networks. In *ACM SIGCOMM Computer Communication Review (CCR)*, 2018.

[13] C. Avin and S. Schmid. Renets: Toward statically optimal self-adjusting networks. *CoRR*, arXiv:1904.03263, 2019.

[14] N. H. Azimi, Z. A. Qazi, H. Gupta, V. Sekar, S. R. Das, J. P. Longtin, H. Shah, and A. Tanwer. Firefly: a reconfigurable wireless data center fabric using free-space optics. In *SIGCOMM*. ACM, 2014.

[15] J. Bao, D. Dong, B. Zhao, Z. Luo, C. Wu, and Z. Gong. Flycast: Free-space optics accelerating multicast communications in physical layer. *Computer Communication Review*, 45(5):97–98, 2015.

[16] T. Benson, A. Akella, and D. A. Maltz. Network traffic characteristics of data centers in the wild. In *Proc. ACM SIGCOMM Conference on Internet Measurement (IMC)*, pages 267–280. ACM, 2010.

[17] T. Benson, A. Anand, A. Akella, and M. Zhang. Understanding data center traffic characteristics. In *Proc. 1st ACM Workshop on Research on Enterprise Networking (WREN)*, pages 65–72. ACM, 2009.

[18] G. Birkhoff. Tres observaciones sobre el algebra lineal. *Univ. Nac. Tucuman, Ser. A*, 5:147–154, 1946.

[19] A. Celik, B. Shihada, and M. Alouini. Wireless data center networks: Advances, challenges, and opportunities. *CoRR*, abs/1811.11717, 2018.

[20] A. Chatzieleftheriou, S. Legtchenko, H. Williams, and A. I. T. Rowstron. Larry: Practical network reconfigurability in the data center. In *NSDI*, pages 141–156. USENIX Association, 2018.

[21] K. Chen, A. Singla, A. Singh, K. Ramachandran, L. Xu, Y. Zhang, X. Wen, and Y. Chen. OSA: an optical switching architecture for data center networks with unprecedented flexibility. *IEEE/ACM Trans. Netw.*, 22(2):498–511, 2014.

[22] K. Chen, X. Wen, X. Ma, Y. Chen, Y. Xia, C. Hu, Q. Dong, and Y. Liu. Toward A scalable, fault-tolerant, high-performance optical data center architecture. *IEEE/ACM Trans. Netw.*, 25(4):2281–2294, 2017.

[23] L. Chen, K. Chen, Z. Zhu, M. Yu, G. Porter, C. Qiao, and S. Zhong. Enabling wide-spread communications on optical fabric with megaswitch. In *NSDI*, pages 577–593. USENIX, 2017.

[24] C. Clos. A study of non-blocking switching networks. *The Bell System Technical Journal*, 32(2):406–424, March 1953.

[25] E. D. Demaine and M. Zadimoghaddam. Minimizing the diameter of a network using shortcut edges. In *SWAT*, 2010.

[26] N. Devanur, J. Kulkarni, G. Ranade, M. Ghobadi, R. Mahajan, and A. Phanishayee. Stable matching algorithm for an agile reconfigurable data center interconnect (MSR-TR-2016-1140). Technical report, Microsoft Research, June 2016.

[27] M. Dong, Q. Li, D. Zarchy, P. B. Godfrey, and M. Schapira. PCC: Re-architecting Congestion Control for Consistent High Performance. NSDI'15, 2015.

[28] V. Dukic, S. A. Jyothi, B. Karlas, M. Owaida, C. Zhang, and A. Singla. Is advance knowledge of flow sizes a plausible assumption? In *NSDI*, pages 565–580. USENIX Association, 2019.

[29] R. Durairajan, P. Barford, J. Sommers, and W. Willinger. Greyfiber: A system for providing flexible access to wide-area connectivity. *CoRR*, abs/1807.05242, 2018.

[30] J. Edmonds. Paths, trees and flowers. *Canad. J. Math*, (17):449–467, 1965.

[31] N. Farrington, A. Forencich, G. Porter, P. . Sun, J. E. Ford, Y. Fainman, G. C. Papen, and A. Vahdat. A multiport microsecond optical circuit switch for data center networking. *IEEE Photonics Technology Letters*, 25(16):1589–1592, Aug 2013.

[32] N. Farrington, G. Porter, Y. Fainman, G. Papen, and A. Vahdat. Hunting mice with microsecond circuit switches. In *HotNets*, pages 115–120. ACM, 2012.

[33] N. Farrington, G. Porter, S. Radhakrishnan, H. H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat. Helios: a hybrid electrical/optical switch architecture for modular data centers. In *SIGCOMM*. ACM, 2010.

[34] T. Fenz, K.-T. Foerster, S. Schmid, and A. Villedieu. Efficient non-segregated routing for reconfigurable demand-aware networks. In *18th IFIP Networking Conference (IFIP Networking)*, May 2019.

[35] K.-T. Foerster, M. Ghobadi, and S. Schmid. Characterizing the algorithmic complexity of reconfigurable data center architectures. In *ANCS*. IEEE/ACM, 2018.

[36] K.-T. Foerster, M. Pacut, and S. Schmid. On the complexity of non-segregated routing in reconfigurable data center architectures. *ACM SIGCOMM Computer Communication Review (CCR)*, 2019.

[37] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[38] M. Ghobadi, R. Mahajan, A. Phanishayee, P.-A. Blanche, H. Rastegarfar, M. Glick, and D. Kilper. Design of mirror assembly for an agile reconfigurable data center interconnect (MSR-TR-2016-1139). Technical report, June 2016.

[39] M. Ghobadi, R. Mahajan, A. Phanishayee, N. R. Devanur, J. Kulkarni, G. Ranade, P. Blanche, H. Rastegarfar, M. Glick, and D. C. Kilper. Projector: Agile reconfigurable data center interconnect. In *SIGCOMM*. ACM, 2016.

[40] S. Ghorbani, Z. Yang, P. Godfrey, Y. Ganjali, and A. Firoozshahian. Drill: Micro load balancing for low-latency data center networks. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 225–238. ACM, 2017.

[41] A. Goel, M. Kapralov, and S. Khanna. Perfect matchings in o(nlog n) time in regular bipartite graphs. *SIAM J. Comput.*, 42(3):1392–1404, 2013.

[42] A. G. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta. VL2: a scalable and flexible data center network. In *SIGCOMM*, pages 51–62. ACM, 2009.

[43] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu. Bcube: a high performance, server-centric network architecture for modular data centers. In *SIGCOMM*, pages 63–74. ACM, 2009.

[44] M. Hall, V. Chidambaram, and R. Durairajan. vFiber: Virtualizing Unused Optical Fibers (Extended Abstract). In *NSDI*, 2018.

[45] D. Halperin, S. Kandula, J. Padhye, P. Bahl, and D. Wetherall. Augmenting data center networks with multi-gigabit wireless links. In *SIGCOMM*. ACM, 2011.

[46] A. S. Hamza, J. S. Deogun, and D. R. Alexander. Wireless communication in data centers: A survey. *IEEE Communications Surveys and Tutorials*, 18(3):1572–1595, 2016.

[47] X. S. Huang, X. S. Sun, and T. S. E. Ng. Sunflow: Efficient optical circuit scheduling for coflows. In *CoNEXT*, pages 297–311. ACM, 2016.

[48] S. Jia, X. Jin, G. Ghasemiesfeh, J. Ding, and J. Gao. Competitive analysis for online scheduling in software-defined optical wan. In *Proc. IEEE INFOCOM*, 2017.

[49] X. Jin, Y. Li, D. Wei, S. Li, J. Gao, L. Xu, G. Li, W. Xu, and J. Rexford. Optimizing bulk transfers with software-defined optical wan. In *Proc. ACM SIGCOMM*, 2016.

[50] P. Kalmbach, J. Zerwas, P. Babarczi, A. Blenk, W. Kellerer, and S. Schmid. Empowering self-driving networks. In *Proc. ACM SIGCOMM 2018 Workshop on Self-Driving Networks (SDN)*, 2018.

[51] S. Kandula, J. Padhye, and P. Bahl. Flyways to de-congest data center networks. In *HotNets*. ACM SIGCOMM, 2009.

[52] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken. The nature of data center traffic: measurements & analysis. In *Proc. 9th ACM SIGCOMM Conference on Internet Measurement (IMC)*, pages 202–208. ACM, 2009.

[53] S. Kassing, A. Valadarsky, G. Shahaf, M. Schapira, and A. Singla. Beyond fat-trees without antennae, mirrors, and disco-balls. In *SIGCOMM*, pages 281–294. ACM, 2017.

[54] S. Kim, D. Cha, Q. Pei, and K. Geary. Polymer optical waveguide switch using thermo-optic total-internal-reflection and strain-effect. *IEEE Photonics Technology Letters*, 22(4):197–199, Feb 2010.

[55] X. Li and M. Hamdi. On scheduling optical packet switches with reconfiguration delay. *IEEE Journal on Selected Areas in Communications*, 21(7):1156–1164, 2003.

[56] H. Liu, F. Lu, A. Forencich, R. Kapoor, M. Tewari, G. M. Voelker, G. Papen, A. C. Snoeren, and G. Porter. Circuit switching under the radar with reactor. In *NSDI*. USENIX, 2014.

[57] H. Liu, M. K. Mukerjee, C. Li, N. Feltman, G. Papen, S. Savage, S. Seshan, G. M. Voelker, D. G. Andersen, M. Kaminsky, G. Porter, and A. C. Snoeren. Scheduling techniques for hybrid circuit/packet networks. In *CoNEXT*, pages 41:1–41:13. ACM, 2015.

[58] V. Liu, D. Halperin, A. Krishnamurthy, and T. E. Anderson. F10: A fault-tolerant engineered network. In *NSDI*. USENIX, 2013.

[59] L. Luo, K.-T. Foerster, S. Schmid, and H. Yu. DaRTree: Deadline-aware Multicast Transfers in Reconfigurable Wide-Area Networks. In *27th IEEE/ACM International Symposium on Quality of Service (IWQoS 2019)*, 2019.

[60] W. M. Mellette, R. Das, Y. Guo, R. McGuinness, A. C. Snoeren, and G. Porter. Expanding across time to deliver bandwidth efficiency and low latency. *CoRR*, abs/1903.12307, 2019.

[61] W. M. Mellette and J. E. Ford. Scaling limits of free-space tilt mirror mems switches for data center networks. In *Optical Fiber Communication Conference*, pages M2B–1. Optical Society of America, 2015.

[62] W. M. Mellette and J. E. Ford. Scaling limits of mems beam-steering switches for data center networks. *Journal of Lightwave Technology*, 33(15):3308–3318, Aug 2015.

[63] W. M. Mellette, R. McGuinness, A. Roy, A. Forencich, G. Papen, A. C. Snoeren, and G. Porter. Rotornet: A scalable, low-complexity, optical datacenter network. In *SIGCOMM*. ACM, 2017.

[64] W. M. Mellette, G. M. Schuster, G. Porter, G. Papen, and J. E. Ford. A scalable, partially configurable optical switch for data center networks. *Journal of Lightwave Technology*, 35(2):136–144, Jan 2017.

[65] W. M. Mellette, A. C. Snoeren, and G. Porter. Toward optical switching in the data center (invited paper). In *Proc. HPSR*, 2018.

[66] A. Meyerson and B. Tagiku. Minimizing average shortest path distances via shortcut edge addition. In *Proc. APPROX/RANDOM*, pages 272–285, Berlin, Heidelberg, 2009.

[67] J. Misra and D. Gries. A constructive proof of vizing's theorem. *Inf. Process. Lett.*, 41(3):131–133, 1992.

[68] M. Moshref, M. Yu, R. Govindan, and A. Vahdat. Trumpet: Timely and precise triggers in data centers. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 129–143. ACM, 2016.

[69] M. Müller-Hannemann and A. Schwartz. Implementing weighted b-matching algorithms: Insights from a computational study. *ACM Journal of Experimental Algorithmics*, 5:8, 2000.

[70] M. Naor and U. Wieder. Novel architectures for p2p applications: the continuous-discrete approach. *ACM Transactions on Algorithms (TALG)*, 3(3):34, 2007.

[71] M. Noormohammadpour and C. S. Raghavendra. Datacenter traffic control: Understanding techniques and trade-offs. *IEEE Communications Surveys & Tutorials*, 2017.

[72] K. Obraczka and P. Danzig. Finding low-diameter, low edge-cost, networks. *Univ. Southern California Technical Report*, 1997.

[73] M. Papagelis, F. Bonchi, and A. Gionis. Suggesting ghost edges for a smaller world. In *Proc. 20th ACM International Conference on Information and Knowledge Management*, pages 2305–2308, 2011.

[74] N. Parotsidis, E. Pitoura, and P. Tsaparas. Selecting shortcuts for a smaller world. In *Proc. SIAM International Conference on Data Mining*, pages 28–36. SIAM, 2015.

[75] B. Peres, O. A. de Oliveira Souza, O. Goussevskaia, C. Avin, and S. Schmid. Distributed self-adjusting tree networks. In *INFOCOM*. IEEE, 2019.

[76] B. Peres, O. Goussevskaia, S. Schmid, and C. Avin. Concurrent self-adjusting distributed tree networks. In *Proc. International Symposium on Distributed Computing (DISC)*, 2017.

[77] G. Porter, R. D. Strong, N. Farrington, A. Forencich, P. Sun, T. Rosing, Y. Fainman, G. Papen, and A. Vahdat. Integrating microsecond circuit switching into the data center. 2013.

[78] K. Ramachandran, R. Kokku, R. Mahindra, and S. Rangarajan. 60 ghz data-center networking: Wireless → worry less? *NEC Research Paper*, 2008.

[79] N. A. Riza and P. J. Marraccini. Power smart in-door optical wireless link applications. In *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 327–332. IEEE, 2012.

[80] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren. Inside the social network's (datacenter) network. In *ACM SIGCOMM Computer Communication Review*, volume 45. ACM, 2015.

[81] S. Salman, C. Streiffer, H. Chen, T. Benson, and A. Kadav. Deepconf: Automating data center network topologies management with machine learning. In *Proceedings of the 2018 Workshop on Network Meets AI & ML*, NetAI'18, pages 8–14, New York, NY, USA, 2018. ACM.

[82] S. Schmid, C. Avin, C. Scheideler, M. Borokhovich, B. Haeupler, and Z. Lotker. Splaynet: Towards locally self-adjusting networks. *IEEE/ACM Trans. Netw.*, 24(3):1421–1433, 2016.

[83] A. Singh, J. Ong, A. Agarwal, G. Anderson, A. Armistead, R. Bannon, S. Boving, G. Desai, B. Felderman, P. Germano, et al. Jupiter rising: A decade of clos topologies and centralized control in google's datacenter network. *ACM SIGCOMM Computer Communication Review (CCR)*, 45(4):183–197, 2015.

[84] R. Singh, M. Ghobadi, K. Foerster, M. Filer, and P. Gill. Run, walk, crawl: Towards dynamic link capacities. In *HotNets*. ACM, 2017.

[85] R. Singh, M. Ghobadi, K.-T. Foerster, M. Filer, and P. Gill. Radwan: Rate adaptive wide area network. In *SIGCOMM*. ACM, 2018.

[86] A. Singla. Fat-free topologies. In *HotNets*, pages 64–70. ACM, 2016.

[87] A. Singla, C. Hong, L. Popa, and P. B. Godfrey. Jellyfish: Networking data centers, randomly. In *HotCloud*. USENIX Association, 2011.

[88] A. Singla, C. Hong, L. Popa, and P. B. Godfrey. Jellyfish: Networking data centers randomly. In *NSDI*. USENIX, 2012.

[89] A. Singla, A. Singh, K. Ramachandran, L. Xu, and Y. Zhang. Proteus: a topology malleable data center network. In *HotNets*. ACM, 2010.

[90] X. S. Sun and T. S. E. Ng. When creek meets river: Exploiting high-bandwidth circuit switch in scheduling multicast data. In *ICNP*, pages 1–6. IEEE Computer Society, 2017.

[91] X. S. Sun, Y. Xia, S. Dzinamarira, X. S. Huang, D. Wu, and T. S. E. Ng. Republic: Data multicast meets hybrid rack-level interconnections in data center. In *ICNP*, pages 77–87. IEEE Computer Society, 2018.

[92] F. Testa and L. Pavesi, editors. *Optical Switching in Next Generation Data Centers*. Springer, 2018.

[93] A. Valadarsky, G. Shahaf, M. Dinitz, and M. Schapira. Xpander: Towards optimal-performance datacenters. In *CoNEXT*, pages 205–219. ACM, 2016.

[94] L. G. Valiant. A scheme for fast parallel communication. *SIAM J. Comput.*, 11(2):350–361, 1982.

[95] S. B. Venkatakrishnan, M. Alizadeh, and P. Viswanath. Costly circuits, submodular schedules and approximate carathéodory theorems. In *SIGMETRICS*, pages 75–88. ACM, 2016.

[96] S. B. Venkatakrishnan, M. Alizadeh, and P. Viswanath. Costly circuits, submodular schedules and approximate carathéodory theorems. *Queueing Syst.*, 88(3-4):311–347, 2018.

[97] J. Von Neumann. A certain zero-sum two-person game equivalent to the optimal assignment problem. *Contributions to the Theory of Games*, 2(0):5–12, 1953.

[98] G. Wang, D. G. Andersen, M. Kaminsky, M. Kozuch, T. S. E. Ng, K. Papagiannaki, M. Glick, and L. B. Mummert. Your data center is a router: The case for reconfigurable optical circuit switched paths. In *HotNets*. ACM SIGCOMM, 2009.

[99] G. Wang, D. G. Andersen, M. Kaminsky, K. Papagiannaki, T. S. E. Ng, M. Kozuch, and M. P. Ryan. c-through: part-time optics in data centers. In *SIGCOMM*, pages 327–338. ACM, 2010.

[100] H. Wang, Y. Xia, K. Bergman, T. S. E. Ng, S. Sahu, and K. Sripanidkulchai. Rethinking the physical layer of data center networks of the next decade: using optics to enable efficient *-cast connectivity. *Computer Communication Review*, 43(3):52–58, 2013.

[101] H. Wang, X. Yu, H. Xu, J. Fan, C. Qiao, and L. Huang. Integrating coflow and circuit scheduling for optical networks. *IEEE Transactions on Parallel and Distributed Systems*, 2019.

[102] M. Wang, Y. Cui, S. Xiao, X. Wang, D. Yang, K. Chen, and J. Zhu. Neural network meets DCN: traffic-driven topology adaptation with deep learning. *POMACS*, 2(2):26:1–26:25, 2018.

[103] W. Xia, P. Zhao, Y. Wen, and H. Xie. A survey on data center networking (DCN): infrastructure and operations. *IEEE Communications Surveys and Tutorials*, 19(1):640–656, 2017.

[104] Y. Xia, T. S. E. Ng, and X. S. Sun. Blast: Accelerating high-performance data analytics applications by optical multicast. In *INFOCOM*, pages 1930–1938. IEEE, 2015.

[105] Y. Xia, X. S. Sun, S. Dzinamarira, D. Wu, X. S. Huang, and T. S. Eugene Ng. A tale of two topologies: Exploring convertible data center network architectures with flat-tree. In *SIGCOMM*. ACM, 2017.

[106] X. Zhou, Z. Zhang, Y. Zhu, Y. Li, S. Kumar, A. Vahdat, B. Y. Zhao, and H. Zheng. Mirror mirror on the ceiling: flexible wireless links for data centers. In *SIGCOMM*. ACM, 2012.

[107] D. Zhuo, M. Ghobadi, R. Mahajan, K.-T. Foerster, A. Krishnamurthy, and T. E. Anderson. Understanding and mitigating packet corruption in data center networks. In *SIGCOMM*. ACM, 2017.

[108] S. Zou, X. Wen, K. Chen, S. Huang, Y. Chen, Y. Liu, Y. Xia, and C. Hu. Virtualknotter: Online virtual machine shuffling for congestion resolving in virtualized datacenter. *Computer Networks*, 67:141–153, 2014.