Reconfigurable routing in data center networks

David C. Kutner[1 [0000−0003−2979−4513]] and Iain A. Stewart[1 [0000−0002−0752−1971]]

Department of Computer Science, Durham University,
Upper Mountjoy Campus, Stockton Road, Durham DH1 3LE, UK
{david.c.kutner, i.a.stewart}@durham.ac.uk

Abstract. A hybrid network is a static (electronic) network that is augmented with optical switches. The Reconfigurable Routing Problem (RRP) in hybrid networks is the problem of finding settings for the optical switches augmenting a static network so as to achieve optimal delivery of some given workload. The problem has previously been studied in various scenarios with both tractability and NP-hardness results obtained. However, the data center and interconnection networks to which the problem is most relevant are almost always such that the static network is highly structured (and often node-symmetric) whereas all previous results assume that the static network can be arbitrary (which makes existing computational hardness results less technologically relevant and also easier to obtain). In this paper, and for the first time, we prove various intractability results for RRP where the underlying static network is highly structured, for example consisting of a hypercube, and also extend some existing tractability results.

Keywords: algorithms · complexity · reconfigurable topologies · optical circuit switches · software-defined networking.

1 Introduction

The rapid growth of cloud computing applications has induced demand for new technologies to optimize the performance of data center networks dealing with ever-larger workloads. The data center topology design problem (that of finding efficient data center topologies) has been studied extensively and resulted in myriad designs (see, e.g., [5]). Advances in hardware, such as optical switches reconfigurable in milli- to micro-seconds, have enabled the development of reconfigurable topologies (see, e.g., [14]). These topologies can adjust in response to demand (*demand-aware* reconfigurable topologies) or vary configurations over time according to a fixed protocol (*demand-oblivious* reconfigurable topologies; see, e.g., [2]). So-called *hybrid* data center networks are a combination of a static topology consisting of, for example, electrical switches, and a demand-aware reconfigurable topology implemented, for example, with optical circuit switches or free space optics (see, e.g., [4,11,15,19]). An intuitive example of a simple reconfigurable topology is illustrated in Fig. 1.

The hybrid network paradigm combines the robustness guarantees of static networks with the ability of demand-aware reconfigurable networks to serve large

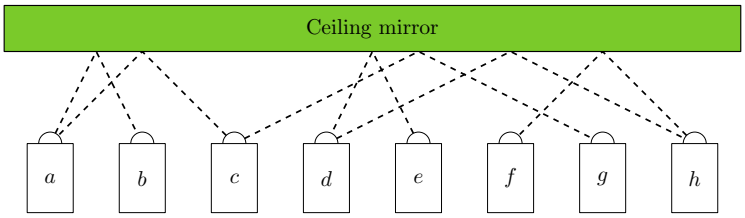

Fig. 1. Basic model of an optical wireless data-center network, as described in [4,15,19]. Practical timescales for reconfiguration vary from milliseconds [15] to microseconds or nanoseconds [4,19].

workloads at very low cost. Consider, for example, the hybrid network shown in Figure 2, and the configuration shown in Figure 3. In the (unaugmented) static network, there are two possible paths along which a message from node b to node d may be routed: $b \to f \to h \to e \to d$ or $b \to f \to e \to d$. In the hybrid network as configured in 3, the path $b \dashrightarrow a \to c \dashrightarrow d$ (among others) is an option*.

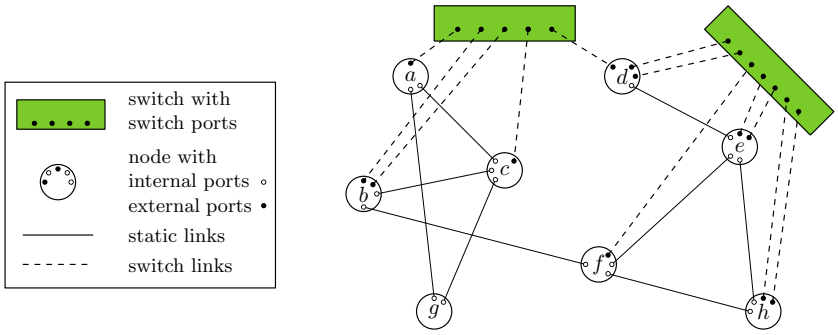

Fig. 2. A hybrid network.

Of particular interest to us is the question of how the reconfigurable (optical) portion of the network should be configured for some demand pattern, formalized by Foerster, Ghobadi and Schmid [9] as the Reconfigurable Routing Problem (RRP): in short, given a hybrid network (consisting of a static network and of some switches) and a workload, we wish to choose a *configuration* (setting of the switches) which results in an optimal delivery of the workload.

Crucially, existing hardness results are only valid when the static network is allowed to be arbitrary, which is almost never the case in practice where interconnection and data center network design is driven by symmetry, high connectivity,

*We denote by $u \dashrightarrow v$ the concatenation of a switch link from u to some switch, of the internal switch connection, and of a switch link to v from that switch.

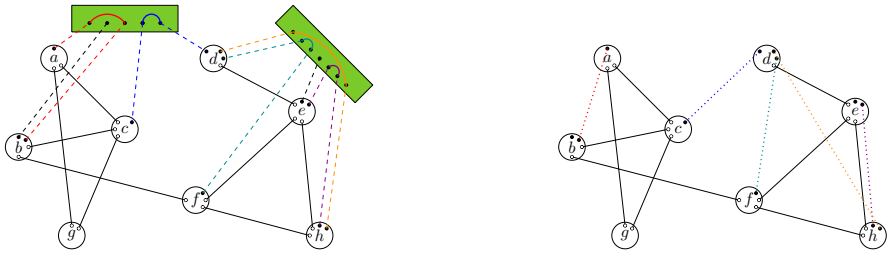

Fig. 3. An augmented network and its abstracted dynamic links.

recursive decomposition, and so forth. For example: the popular switch-centric data center network Fat-Tree [1] is derived from a folded Clos network; the server-centric data center network DCell [13] is recursively-structured whereby at each level, a graph-theoretic matching of servers is imposed; and the server-centric data center network BCube [12] is recursively-structured with a construction based around a generalized hypercube. (It should be noted that there do exist examples of unstructured data center networks, such as Jellyfish [17] and Xpander [18] which utilize the theory of random graphs.) Many (but not all) NP-complete problems become tractable when the input is restricted to the graphs providing the communications fabric for data center networks and other interconnection networks. For example, Hamiltonian paths are often trivial to find in many interconnection networks; indeed, no finite connected vertex-transitive graph *without* a Hamiltonian path is known to exist (the Lovász Conjecture contends there is no such graph - see Section 4 of [16]). This motivates our investigation into how the complexity of RRP changes when we restrict to more structured and realistic networks. The question of the complexity of RRP for specific network topologies was specifically identified as an area for future work in [8].

In this paper, we establish for the first time hardness results for RRP that apply to various specific families of highly structured static networks such as, for example, the hypercubes. Our constructions are (perhaps not surprisingly) of a much more involved nature than has hitherto been the case.

2 Problem Setting

The decision problem Reconfigurable Routing Problem considered in this paper is a proper restriction of that presented in prior work [8,9,10]. In this section, we provide technical detail to fully formalize our version of the problem, but also additionally provide sufficient framing to briefly review existing results and to identify the areas strengthened by our contribution.

We adopt the usual terminology of graph theory though we tend to use 'nodes' and 'links' when speaking about the components of reconfigurable networks and 'nodes' and 'edges' when dealing with (abstract) graphs. We denote the natural numbers by $\mathbb{N}$ (we include $0 \in \mathbb{N}$) and the non-negative rationals by $\mathbb{Q}_+$.

2.1 Hybrid networks, (re)configurations and (segregated) routing

A hybrid network $G(S)$ can be visualized as in Fig. 2, and consists of a static network G and some switches S augmenting it. A *static network* G can be abstracted as an undirected graph $G = (V, E)$ so that each *static link* $(u, v) \in E$ has some fixed *weight* $w \in \mathbb{Q}_+$ (reflecting a transmission cost) and is incident with *internal ports* of two distinct nodes of V. The number of internal ports of some node $v \in V$ is then exactly the degree of v in the abstracted graph G. We denote by S a set of *switches* augmenting the static network G with *switch links* joining *switch ports* of some switch to *external ports* of some of the nodes of V. Every switch link has weight 0 (we say more about switch link weights momentarily). Every switch $s \in S$ has at least two switch ports.

In general, the number of external ports of the nodes of a static network $G = (V, E)$ is variable, as is the number of switch ports of the switches of a hybrid network $G(S)$, and it may be the case that there is more than one switch link between a specific node and a specific switch. We assume that the switch links describe a bijection between the external ports and the switch ports; otherwise, there would be some unused ports, which we can safely ignore.

Given a hybrid network $G(S)$ and a switch $s \in S$, a *switch matching* N_s of s is a set of pairs of switch ports of s so that all switch ports involved are distinct. Each switch matching represents an internal setting of the switch and naturally yields a set of pairs of external ports of nodes where all such ports are distinct; we refer to a set of pairs of external ports obtained in this way as a *node matching* (note that this differs from the standard graph-theoretic notion of a matching). An illustration of a configured hybrid network is shown in Figure 3: on the right side, switch matchings are represented as sets of arcs, and on the left side the corresponding node matching is shown as a set of dotted lines.

A *configuration* N is a set of switch matchings, one for each switch. A configuration straightforwardly encodes the corresponding node matchings. We say that (u, v) is a *dynamic link* in the configuration N (we sometimes write $(u, v) \in N$) if (u, v) appears in any node matching corresponding to N.

We allocate a fixed weight $\mu \in \mathbb{Q}_+$ to each internal port-to-port connection in a switch s. Although a dynamic link is an atomic entity, it can be visualized as consisting of a switch link followed by an internal port-to-port connection in s followed by another switch link. We denote by $G(N)$ the static network G augmented with the dynamic links (each of weight μ) resulting from the configuration N and we call $G(N)$ an *augmented network*. In the augmented network visualized in Figure 3, for example: (a, b) is a dynamic link; (a, c) is a static link; and (e, h) is both a static link a dynamic link. Note that it is possible that an augmented network $G(N)$ is a multigraph.

The concepts defined above are driven by reconfigurable hardware technology such as optical switches, wireless (beamforming) and free-space optics, all of which establish port-to-port connections, i.e., switch matchings. The survey paper [11] provides some detail as regards the relationship between the emergent theoretical models and current opto-electronic technology.

2.2 Routing in hybrid networks

Consider again the example shown in Figure 3. In the configuration shown, a message M from c to node e may be routed:

1. via static links only, along the path $\varphi_1 := c \to b \to f \to e$ with weight $3w$, or
2. via dynamic links only, along the path $\varphi_2 := c \dashrightarrow d \dashrightarrow h \dashrightarrow e$ with weight 3μ, or
3. via a combination of static and dynamic links, along the path $\varphi_3 := c \dashrightarrow d \to e$ with weight $\mu + w$.

Depending on the value of μ, any of the paths may minimize the cost to route M: if $\mu \geq 2w$ then φ_1 is optimal; if $\mu \leq \frac{w}{2}$ then φ_2 is optimal; and if $\mu \in [\frac{w}{2}, 2w]$ then φ_3 is an optimal. We may wish to bound the number of alternations allowed between optic and static links in any path a message takes; we capture this hardware requirement via a *segregation parameter* $\sigma \in \mathbb{N} \cup \{\infty\}$, as introduced in [10], that is the number of alternations between static and dynamic links. In the fully segregated case, $\sigma = 0$: messages may be routed either by static links only (as in φ_1) or by dynamic links only (as in φ_2). In the non-segregated case, $\sigma = \infty$ and there is no restriction on the number of alternations, so any path is admitted. Note φ_3 is admitted as a valid path to route M if and only if $\sigma \geq 1$.

Networks are expected to route many messages (of varying sizes) optimally at the same time. Given a hybrid network $G(S)$ we represent the set of all demands we must optimize for as a *workload* (*matrix*) D with entries $\{D[u, v] \in \mathbb{Q}_+ : u, v \in V\}$ providing the intended pairwise *node-to-node workloads* (each $D[u, u]$ is necessarily 0).

Given a configuration N and $u, v \in V$ for which $D[u, v] > 0$, we route the corresponding workload via a path in $G(N)$ from u to v in $G(N)$ so that this chosen *flow-path* $\varphi(u, v)$ has *workload cost* $D[u, v] \times wt_{G(N)}(\varphi(u, v))$, where the *weight* $wt_{G(N)}(\varphi(u, v))$ is the sum of the weights of the links of the flow-path $\varphi(u, v)$ (if $G(N)$ has both a static link (x, y) and a dynamic link (x, y) then we need to say which we are using in $\varphi(u, v)$). The *total workload cost* (of D under N) is defined as

$$\sum_{u,v \in V, D[u,v]>0} D[u, v] \times wt_{G(N)}(\varphi(u, v)).$$

Our aim will be to find a configuration N in some hybrid network $G(S)$ and flow-paths in $G(N)$ for which the total workload cost of some workload matrix D is minimized. In an unrestricted scenario, we would choose any flow-path $\varphi(u, v)$ to be a flow-path of minimum weight from u to v in $G(N)$, the weight of which we denote by $wt_{G(N)}(u, v)$. When $\sigma \neq \infty$ we must also ensure the flow-path has at most σ alternations. We also have the analogous concepts $wt_G(\varphi(u, v))$ and $wt_G(u, v)$ where we work entirely in the static network G. Note that we often describe D by a weighted digraph, which we usually call D', so that the node set is V and there is an edge (u, v) of weight $w > 0$ if, and only if, $D[u, v] = w$. We also refer to some $D[u, v] > 0$ as a *demand* (from u to v).

2.3 The Reconfigurable Routing Problem

We are now in a position to introduce our protagonist:

Reconfigurable Routing Problem (σ) (RRP(σ))
Input: $(G, S, \mu, w, D, \kappa)$: D is a workload matrix for the hybrid network $G(S)$ with static (resp. dynamic) links all of weight w (resp. μ).
Question: Does $G(S)$ admit some configuration N such that the total workload cost of D under N (where the number of alternations for any path is bounded by σ) is at most κ?

As previously alluded to, this setting is more expressive than we require for most of this paper, and more restrictive than the exact formalism considered in prior work [8,9,10]: in those works, w and μ are sometimes allowed to be functions of their endpoints rather than fixed constants. This provides much more expressivity; notably, their model loses no power when it is restricted to inputs where G is a complete graph and there is only one switch, since it is possible to simulate any other instance by assigning prohibitively large weights to any static edges and any pair of switch ports which should not be usable.

We now turn to the "realistic" networks we mentioned in our introduction. Henceforth unless otherwise specified, static link weights are all equal (and normalized to 1) and dynamic link weights are always some fixed constant $\mu \in \mathbb{Q}_+$. Also, there is a single switch and all nodes are connected to it with identical hardware. This is both practically relevant and intuitively realistic; see e.g. Fig. 1. Then the set of switches S of the hybrid network consists of just one switch, which is fully described by the number of switch links each node in the hybrid network has, which we call Δ_S. This is closely related to the maximum reconfigurable degree Δ_R from [10], which is an upper bound on the number of external ports per node. The resulting restriction of RRP can be formalized as follows:

Δ_S-switched RRP (σ)
Input: (G, μ, D, κ): D is a workload matrix for the hybrid network $G(S)$ with static (resp. dynamic) links all have weight 1 (resp. μ) (where S consists of a single switch that every node in G is connected to exactly Δ_S times).
Question: Does $G(S)$ admit some configuration N such that the total workload cost of D under N (where the number of alternations for any path is bounded by σ) is at most κ?

3 Results

Table 1 shows a summary of hardness results from previous work as well as our three main intractability results. In general terms, we obtain NP-completeness

<table>
<tr><th>Result</th><th>$|S|$</th><th>Δ_R</th><th>σ</th><th>D</th><th>link weights</th><th>notes</th></tr>
<tr><td>[8], Theorem 1</td><td>$\Theta(n)$ (or 1 †)</td><td>$\Theta(n)$</td><td rowspan="2">any $\sigma \geq 2$</td><td rowspan="2">sparse, all values 0 or 1</td><td>variable; $w \in [1, 100n^2]$ $\mu \in [1, 100n^2]$</td><td>Showed inapprox. within $\Omega(\log n)$</td></tr>
<tr><td>[9], Lemma 1</td><td>$\Theta(n)$</td><td>1</td><td rowspan="3">fixed; $w = \mu = 1$</td><td>All switches have 3 ports.</td></tr>
<tr><td>[9], Theorem 2</td><td rowspan="5">1</td><td rowspan="2">2</td><td rowspan="3">any $\sigma \geq 0$</td><td>dense, values in poly(n)</td><td>G has $\Theta(n)$ components</td></tr>
<tr><td>[10], Theorems 4.1, 4.2</td><td>G is empty; there are no static links</td></tr>
<tr><td>Theorem 1</td><td rowspan="3">sparse, values in poly(n)</td><td>fixed; $w = 1$ $\mu \in \Theta(\frac{1}{\text{poly}(n)})$</td><td rowspan="2">$G \in \mathcal{H}$, where $\mathcal{H}$ is any polynomial family of networks (incl. hypercubes, grids, cycles).</td></tr>
<tr><td>Theorem 2</td><td>3</td><td>fixed; $w = 1$ $\mu \in \Theta(\frac{1}{\log(n)})$</td></tr>
<tr><td>Theorem 4</td><td>1</td><td>$\sigma = 3$</td><td>fixed; $w = 1$ any $\mu \in (0, 1)$</td><td>G is a hypercube</td></tr>
</table>

Table 1. Settings for some pre-existing hardness results for RRP. $|S|$ is the number of switches; Δ_R is the maximum number of external ports per node; σ is the segregation parameter; D is the workload matrix; n denotes the number of nodes in the instance.

for 2-SWITCHED RRP and 3-SWITCHED RRP on any fixed class of static networks of practical interest (defined more fully below) and for any value of σ. We then restrict our focus (and associated parameters) to the case where the static network is a hypercube when we establish the NP-completeness of 1-SWITCHED RRP($\sigma = 3$) in this setting; we conjecture that a similar construction can be used to establish hardness when $\sigma > 3$. We also, in Theorem 3, show that 1-SWITCHED RRP($\sigma = 0$) is solvable in polynomial time. The cases when $\sigma \in \{1, 2\}$ remain interesting open problems.

As is standard in NP-hardness proofs, we reduce from known NP-complete problems to instances of RRP; the challenge is that, due to the expansive scope of our theorems, we lose several "degrees of freedom" which are used for encoding hard instances in, e.g., [7,8,9]. Specifically, we may not make use of varying static or dynamic link weights to prohibit certain connections, nor encode any features of the input instance in the topology of the hybrid network $G(S)$. For example, in Lemma 1 [9], many small switches with two feasible configurations each are

† By using variable μ with prohibitively large weights, it is possible to simulate many switches with just one.

used to encode a truth assignment, and in Theorem 1 of [8] "bad" links are given weights of order $\Theta(n^2)$. Neither of these mechanisms can be leveraged to obtain hardness in our setting; in this sense, our hardness results are strictly stronger and also harder to obtain than those from [7,8,9]. We are constrained to choose a *size* for the network G, and then to encode the input instance in the demand matrix D.

Our first two results hold for a wide class of graph families, which may be of broader interest for the study of computational hardness in network problems. Rather than allowing arbitrary static networks in instances of RRP, we wish to force any such static network to come from a fixed family of networks where a *family of networks* $\mathcal{H}$ is an infinite sequence of networks $\{H_i : i \geq 0\}$ so that the size $|H_i|$ of any H_i is less than the size of H_{i+1}. However, we wish to control the sequence of network sizes. Consequently, we define a *polynomial family of networks* as being a family of networks $\mathcal{H} = \{H_i : i \geq 0\}$ where there exists a polynomial $p_{\mathcal{H}}(x)$ so that $|H_{i+1}| = p_{\mathcal{H}}(|H_i|)$, for each $i \geq 0$[‡]. Note that given any $n \geq 0$, we can determine in time polynomial in n the smallest i such that $n \leq |H_i|$. As an example of a polynomial family of networks, consider the hypercubes; here, the polynomial $p_{\mathcal{H}}(x) = 2x$. Other examples include independent sets, complete graphs, cycles, complete binary trees and square grids, among many others. The sweeping generality of having a single construction which holds for any polynomial family $\mathcal{H}$ poses a challenge in our proofs of Theorems 1 and 2; we require that our constructed network $H(S)$ behaves identically when H is a connected (or even complete) graph and, at the opposite extreme, when H is disconnected (or even independent). For reasons of length, full proofs are deferred to the full version of this paper; we provide our construction for Theorem 1 in full, along with a sketch of the proof.

Theorem 1. *For any polynomial family of networks $\mathcal{H} = \{H_i : i \geq 0\}$, the problem 2-SWITCHED RRP restricted to instances (H, μ, D, κ) satisfying:*

- *$H \in \mathcal{H}$ has size n*
- *the workload matrix D is sparse and all values in it are polynomial in n*
- *$\mu \in \Theta(\frac{1}{n})$ is fixed for all dynamic links*

is NP-complete.

Proof. 2-SWITCHED RRP is straightforwardly in **NP** as a subproblem of RRP. We describe a polynomial-time reduction from the problem 3-MIN-BISECTION, which is known to be **NP**-complete [3] and is defined as follows:

[‡]Technically, we insist that there exists a polynomial Turing machine $\mathcal{M}$ which computes H_{i+1} on input H_i, for each $i \geq 0$, but this definition obfuscates the utility of this description.

3-Min-Bisection
Input: $(G = (V, E), k)$: G is a 3-regular graph on n vertices and $k \in \mathbb{N}$.
Question: Is there a partition of V into two disjoint subsets A and B, of equal size, so that the set of edges incident with both a node in A and a node in B has size at most k? Or: does G have *bisection width* at most k?

Note that any 3-regular graph necessarily has an even number of nodes, and that we may assume that $k \leq \frac{n}{3} + 46$ as it was proven in [6] that every 3-regular graph has bisection width at most $\frac{n}{3} + 46$. Given an arbitrary instance $(G = (V, E), k)$ of size n of 3-Min-Bisection, we now build our instance (H, μ, D, κ).

We describe our workload matrix D via the weighted digraph $D' = (V', E')$, which has a directed edge (u, v) with weight w if, and only if, there is a node-to-node workload of w from u to v. Let $\bar{n}$ be the size of the network H_i where i is the smallest integer such that $n + 6n^2 + 2 \leq |H_i|$ and set $H = H_i$.

The node set V' is taken as a disjoint copy of the node set V of G, which we also refer to as V, together with the set of nodes $V_c = \{x_i, y_i : -\frac{L}{2} \leq i \leq \frac{L}{2}\}$, where $L = 3n^2$ (recall, n is even), and another set of nodes U of size $\bar{n} - (n + 6n^2 + 2)$; so, $|V'| = \bar{n}$. We call every node of V_c a *chain-node*. For ease of presentation, we denote the chain-nodes $x_{\frac{L}{2}}$ and $x_{-\frac{L}{2}}$ by x^+ and x^-, respectively, and we define the chain-nodes y^+ and y^- analogously. The (directed) edge set E' consists of $E_\alpha \cup E_\beta \cup E_1$ where:

- the set of *chain-edges* $E_\alpha = \{(x_i, x_{i+1}), (y_i, y_{i+1}) : 0 \leq i < \frac{L}{2}\} \cup \{(x_i, x_{i-1}), (y_i, y_{i-1}) : -\frac{L}{2} < i \leq 0\}$
- the set of *star-edges* $E_\beta = \{(x_0, v) : v \in V\} \cup \{(y_0, v) : v \in V\}$
- the set of *unit-edges* E_1 which is a copy of the edges E of G, but on our (copied) node set V and so that every edge is replaced by a directed edge of arbitrary orientation.

Note that the nodes of U are all isolated in D' and that $|V'| = \bar{n}$ (the nodes of U will play no role in the following construction). The workloads on the edges of E' are α, β or 1 depending upon whether the edge is a chain-edge from E_α, a star-edge from E_β or a unit-edge from E_1, respectively, where we define $\alpha = 24n^6$ and $\beta = 6n^3$. If the directed edge (u, v) has weight α (resp. β, 1) in D' then we say that (u, v) is an α-demand (resp. β-demand, 1-demand). The digraph D' can be visualized as in Fig 4.

As stated earlier, our static network H is the network $H_i \in \mathcal{H}$ where $|H_i| = \bar{n}$. We refer to the node set of H as V' also and we refer to the subset of nodes within V' corresponding to V as V also. Since we are in the 2-switched setting, we have one switch s with $2|V'|$ ports so that every node of H is adjacent, via switch links, to exactly two ports of the switch. Hence, our switch set is $S = \{s\}$ and our hybrid network is $H(S)$. It is important to note that for any configuration N, any node of $H(N)$ can be adjacent to at most 2 other nodes via dynamic links (as $\Delta_S = 2$).

As can be seen, we have the graph $G = (V, E)$, the digraph $D' = (V', E')$ and the hybrid network $H(S)$ with node set V'. Although G, D' and $H(S)$ are

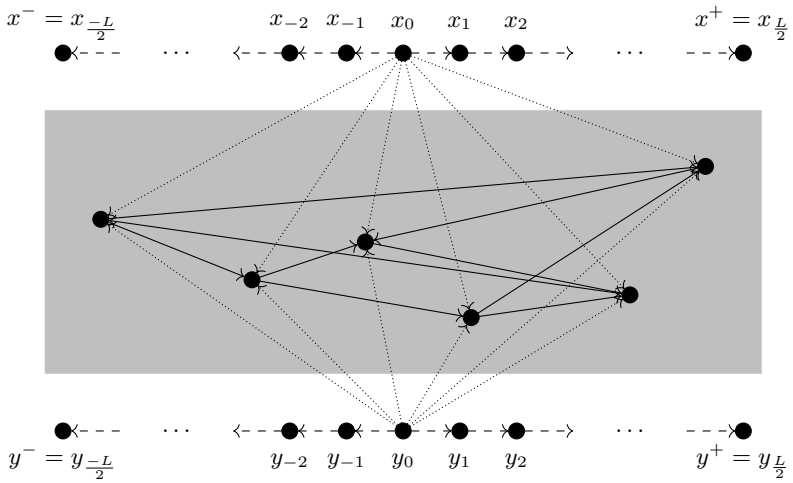

Fig. 4. The digraph D'. The nodes of V are in the grey rectangle, the nodes of V_c appear along the top and the bottom and the dashed (resp. dotted, solid) directed edges depict the chain-edges (resp. star-edges, unit-edges). The nodes of U are omitted.

disjoint in terms of node sets, we do not distinguish between, say, the node set V of G and the subset of nodes V of H.

We set the weight of any dynamic link as $\mu = \frac{1}{2L} = \frac{1}{6n^2}$ and the bound κ for the total workload cost as $\kappa = \kappa_\alpha + \kappa_\beta + \kappa_1$ where:

- $\kappa_\alpha = 24n^6$
- $\kappa_\beta = 3n^4 + \frac{n^3}{2} + n^2$
- $\kappa_1 = \frac{k}{2} + \frac{1}{8} - \frac{1}{4n} + \frac{k}{3n^2}$.

The values of κ_α, κ_β and κ_1 have the following significance.

- Suppose that for every chain-edge (u, v) of E_α, N contains the dynamic link joining u and v in $H(N)$ and the α-demand (u, v) is routed by the flow-path $u \dashrightarrow v$. Then the total workload cost of flow-paths serving α-demands is $2L\alpha\mu = 24n^6 = \kappa_\alpha$.
- Further, suppose that the dynamic links incident with nodes of V in $H(N)$ are chosen so that we have a path of dynamic links p_A from x^+ to either y^- or y^+, involving the subset of nodes $A \subseteq V$, and a path of dynamic links p_B from x^- to y^+ or y^-, respectively, involving the subset of nodes $B \subseteq V$, so that both p_A and p_B have length $\frac{n}{2} + 1$. That is, we choose the dynamic links so that they form a cycle C (of length $n + 2L + 2$) in $H(N)$ covering exactly the nodes of V and V_c. Suppose that for any star-edge (x_0, v) (resp. (y_0, v)) of E_β, we choose the flow-path in $H(N)$ serving this star-edge as consisting entirely of dynamic links resulting from the shortest path in our cycle C from x_0 to v (resp. y_0 to v). The total workload cost of flow-paths

corresponding to the star-edges is

$$4\mu\beta\sum_{i=1}^{\frac{n}{2}}(\frac{L}{2}+i) = 3n^4 + \frac{n^3}{2} + n^2 = \kappa_\beta.$$

– Further, suppose we choose the flow-path in $H(N)$ serving the 1-demand (u, v) (in E_1) to be a path of dynamic links within the cycle C of shortest length. If u and v both lie on p_A or both lie on p_B then the workload cost of this flow-path is at most $\mu(\frac{n}{2}-1) = \frac{1}{6n}(\frac{1}{2}-\frac{1}{n})$, and if one of u and v lies on p_A with the other node lying on p_B then the workload cost of this flow-path is at most $\mu(\frac{n}{2}+L+1) = \frac{1}{2}+\frac{1}{12n}+\frac{1}{6n^2}$. If the width of the bisection of G formed by A and B is at most k then the total workload cost of flow-paths corresponding to the unit-edges is at most

$$(\frac{3n}{2}-k)\mu(\frac{n}{2}-1) + k\mu(\frac{n}{2}+L+1) = \frac{k}{2}+\frac{1}{8}-\frac{1}{4n}+\frac{k}{3n^2} = \kappa_1.$$

From above, we immediately obtain that if (G, k) is a yes-instance of 3-Min-Bisection then (H, μ, D, κ) is a yes-instance of 2-switched RRP.

It remains to show that if (H, μ, D, κ) is a yes-instance of 2-switched RRP, then (G, k) is a yes-instance of 3-Min-Bisection; this is much more technical. For reasons of length, we provide only a flavor of the full proof here. Our first step is to show that all chain-edges are realized as dynamic links (i.e. $E_\alpha \subseteq N$, abusing notation slightly), then that the set of dynamic links N forms a cycle C covering exactly the nodes of $V \cup V_c$. From there, we obtain that deleting V_c from C produces two paths on exactly $\frac{n}{2}$ nodes, and hence encodes a bisection A, B of G (the function of the β-demands is to force $|A| = |B|$). Lastly, applying our choice of κ_1 we obtain that there are at most k edges in G between A and B. So, if (H, μ, D, κ) is a yes-instance of 2-switched RRP then (G, k) is a yes-instance of 3-Min-Bisection. Our result follows as (H, μ, D, κ) can be constructed from (G, k) in time polynomial in n. □

This result significantly strengthens Theorems 4.1 and 4.2 from [10]: there, RRP($\Delta_R \geq 2, \sigma = 0$) is shown to be NP-complete when the static network is an independent set, and the proof does not enable us to restrict the workload matrix D meaningfully. The main weakness of Theorem 1 is its reliance on μ being a polynomial factor smaller than any static link weight. This is actually related to the fact that a connected 2-regular network, as is $G(N)$ when G is an independent set and $\Delta_S = 2$, has diameter linear in the number of nodes n. A network of maximum degree 3, on the other hand, may have diameter logarithmic in n (e.g., a complete binary tree has this property) and we indeed show NP-completeness of RRP($\Delta_S = 3$) when $\mu = \Theta(\frac{1}{\log n})$.

Theorem 2. *For any polynomial family of networks $\mathcal{H} = \{H_i : i \geq 0\}$, the problem 3-switched RRP restricted to instances (H, μ, D, κ) satisfying:*

– *$H \in \mathcal{H}$ has size n*

- *the workload matrix D is sparse and all values in it are polynomial in n*
- $\mu \in \Theta(\frac{1}{\log(n)})$

is NP-complete.

These results led us to consider the problem 2-SWITCHED RRP($\sigma = k$) for $k \geq 0$. By extending Theorem 3.1 from [10] (which establishes tractability in the case where $\sigma = 0$ and only paths using at most one dynamic link are admitted), we show that this restriction entails tractability when either $\sigma = 0$ or the static network is a complete graph, in contrast with our NP-completeness results.

Theorem 3. 1-*SWITCHED RRP($\sigma = 0$) is in* **P**.

Proof. Since each node is connected to the switch exactly once, no vertex is incident to two dynamic links under any configuration N, and hence no flow-path consists of two or more dynamic links. That is, the constraint on the number of dynamic links per flow-path is implicit in this setting, and tractability follows from Theorem 3.1 from [10]. □

Corollary 1. 1-*SWITCHED RRP($\sigma = k$) restricted to instances where the static network G is a complete graph is in* **P**, *for any $k \in \mathbb{N} \cup \{\infty\}$.*

Proof. If G is a complete graph (with all edge weights equal) then without loss under any configuration N, each demand $D[u, v]$ is routed via the flow-path $\varphi(u, v)$ of minimum weight, which is either a single static link from u to v with unit weight, or a single dynamic link from u to v with weight μ. It follows that setting $\sigma = 0$ introduces no new constraints, and then by Theorem 3 we have tractability. □

Corollary 1 rules out the possibility that 1-SWITCHED RRP might be NP-complete for any polynomial graph family $\mathcal{H}$ (since such a claim would extend to the family of complete graphs) unless **P** equals **NP**. This leaves open the practically relevant case where $\Delta_S = 1$ and $\sigma > 0$ for *specific* topologies. We consequently consider the scenario where the static network is a hypercube and the segregation parameter $\sigma = 3$.

Theorem 4. *For any fixed $\mu \in (0, 1)$, the problem 1-SWITCHED RRP($\sigma = 3$) restricted to instances (H, μ, D, κ) satisfying:*

- *$H \in \mathcal{Q}$, where $\mathcal{Q} := \{Q_d | d \in \mathbb{N}\}$ is the family of hypercubes*
- *the workload matrix D is sparse and all values in it are polynomial in n*

is NP-complete.

We emphasize the relevance of the hypercube as a prototypical model of interconnection networks (see, e.g., [12]) and the fact that we obtain hardness here for any choice of fixed dynamic link weight μ.

4 Discussion and Future Work

Taken together, our results comprehensively establish the computational hardness of RRP in practically relevant settings. We establish that the problem remains intractable in several cases where the demand matrix is sparse, the hybrid network is highly structured (in fact node-symmetric) and the weights of links depend only on their medium. Furthermore, in Theorems 1, 2, and 4, the instrument used to "express" NP-completeness is the demand matrix D. In the real world, the computational workload for the network is generally expected to vary significantly with time, unlike the network's hardware, which (in addition to its structural properties already discussed) does not rapidly change. Our results are in this sense closely relevant to the hardness of the real world reconfigurable routing problem.

We take this opportunity to identify some specific questions we have left open, as well as several more general avenues for future work in this area. First, it would be interesting to study the restriction of the problem to cases where Δ_S is greater than 1 and μ is a fixed constant. Results in this setting would "bridge the gap" between Theorems 1 and 2, and Theorem 4. Analogously, there is a gap for 1-Switched RRP on hypercubes between $\sigma = 0$ (which is solvable in polynomial time) and $\sigma = 3$ (which is an intractable case). The complexity of the problem with $\sigma = 1$ and $\sigma = 2$ remains open for hypercubes (note that results for arbitrary networks do exist when $\sigma = 2$, as shown in Table 1).

Secondly, the present work considers only exact computation. In [8] the authors establish inapproximability within $\Omega(\log n)$ for RRP in a more permissive setting (making use of variable link weights). However, the empty solution (there are no dynamic links and all demands are routed through the static network only) is a $\frac{\log n}{\mu}$-approximation for Δ_S-Switched RRP on hypercubes. (This follows straightforwardly from hypercubes having logarithmic diameter.) It would be interesting to see what (in)approximability results can be derived in our model with fixed link weights, with and without restrictions to realistic topologies.

Lastly, parameterized algorithms may provide more fine-grained insights into the computational complexity of reconfigurable routing. Our Theorems 1 and 2 establish that structural parameters of the static network, such as treewidth, are insufficient to yield fixed-parameter tractable (fpt) algorithms (unless P=NP). However, it would be interesting to see whether it is possible to obtain an fpt algorithm by additionally parameterizing by the sum of the demand matrix D; some structural parameters for the digraph representation of the demands, D'; the dynamic link weight μ; or a combination of these.

References

1. Mohammad Al-Fares, Alexander Loukissas, and Amin Vahdat. A scalable, commodity data center network architecture. *ACM SIGCOMM Comput. Commun. Rev.*, 38:63–74, 2008.
2. Chan Avin and Stefan Schmid. Toward demand-aware networking: a theory for self-adjusting networks. *ACM SIGCOMM Comput. Commun. Rev.*, 48:31–40, 2019.

3. Piotr Berman and Marek Karpinski. Approximation hardness of bounded degree MIN-CSP and MIN-BISECTION. In *Proc. of 29th Int. Colloq. on Automata, Languages and Programming (ICALP)*, pages 623–632, 2002.
4. Charidimos Chaintoutis, Behnam Shariati, Adonis Bogris, Paul V. Dijk, Chris G. H. Roeloffzen, Jerome Bourderionnet, Ioannis Tomkos, and Dimitris Syvridis. Free space intra-datacenter interconnects based on 2d optical beam steering enabled by photonic integrated circuits. *Photonics*, 5(3), 2018.
5. Tao Chen, Xiaofeng Gao, and Chen Guihai. The features, hardware, and architectures of data center networks: a survey. *J. Parallel Distrib. Comput.*, 96:45–74, 2016.
6. L.H. Clark and R.C. Entringer. The bisection width of cubic graphs. *Bull. Austral. Math. Soc.*, pages 389–396, 1988.
7. Thomas Fenz, Klaus-Tycho Foerster, Stefan Schmid, and Anaïs Villedieu. Efficient non-segregated routing for reconfigurable demand-aware networks. In *Proc. of IFIP Networking Conf.*, pages 1–9. IEEE Press, 2019.
8. Thomas Fenz, Klaus-Tycho Foerster, Stefan Schmid, and Anaïs Villedieu. Efficient non-segregated routing for reconfigurable demand-aware networks. *Comput. Commun.*, 164:138–147, 2020.
9. Klaus-Tycho Foerster, Manya Ghobadi, and Stefan Schmid. Characterizing the algorithmic complexity of reconfigurable data center architectures. In *Proc. of Symp. on Architectures for Networking and Communications Systems (ANCS)*, pages 89–96. ACM Press, 2018.
10. Klaus-Tycho Foerster, Maciej Pacut, and Stefan Schmid. On the complexity of non-segregated routing in reconfigurable data center architectures. *ACM SIGCOMM Comput. Commun. Rev.*, 49:2–81, 2019.
11. Klaus-Tycho Foerster and Stefan Schmid. Survey of reconfigurable data center networks: enablers, algorithms, complexity. *ACM SIGACT News*, 50:62–79, 2019.
12. Chuanxiong Guo, Guohan Lu, Dan Li, Haitao Wu, Xuan Zhang, Yunfeng Shi, Chen Tian, Yongguang Zhang, and Songwu Lu. BCube: A high performance, server-centric network architecture for modular data centers. *ACM SIGCOMM Comput. Commun. Rev.*, 39:63–74, 2009.
13. Chuanxiong Guo, Haitao Wu, Kun Tan, Lei Shi, Yongguang Zhang, and Songwu Lu. DCell: a scalable and fault-tolerant network structure for data centers. In *Proc. of ACM SIGCOMM Conf. on Data Communication*, pages 75–86, 2008.
14. Matthew Nance Hall, Kalsu-Tycho Foerster, Stefan Schmid, and Ramakrishnan Durairajan. A survey of reconfigurable optical networks. *Opt. Switch. Netw.*, 41:100621, 2021.
15. Navid Hamedazimi, Zafar Qazi, Himanshu Gupta, Vyas Sekar, Samir R. Das, Jon P. Longtin, Himanshu Shah, and Ashish Tanwer. Firefly: a reconfigurable wireless data center fabric using free-space optics. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, page 319–330, New York, NY, USA, 2014. Association for Computing Machinery.
16. Igor Pak and Radoš Radoičić. Hamiltonian paths in cayley graphs. *Discrete Mathematics*, 309(17):5501–5508, 2009. Generalisations of de Bruijn Cycles and Gray Codes/Graph Asymmetries/Hamiltonicity Problem for Vertex-Transitive (Cayley) Graphs.
17. Ankit Singla, Chi-Yao Hong, Lucian Popa, and P. Brighten Godfrey. Jellyfish: networking data centers randomly. In *Proc. of 9th USENIX Conf. on Networked Systems Design and Implementation*, pages 225–238, 2012.

18. Asaf Valadarsky, Gal Shahaf, Michael Dinitz, and Michael Schapira. Xpander: towards optimal-performance datacenters. In *Proc. of 12th Int. Conf. on Emerging Networking Experiments and Technologies*, pages 205–219, 2016.
19. Shaojuan Zhang, Xuwei Xue, Eduward Tangdiongga, and Nicola Calabretta. Low-latency optical wireless data-center networks using nanoseconds semiconductor-based wavelength selectors and arrayed waveguide grating router. *Photonics*, 9(3), 2022.