Minimizing Congestion in Data-Center Networks with Unsplittable Flow

Anonymous Authors

Abstract

Unsplittable flow problems have long been motivated by computer networking. Today, the most prominent unsplittable flow problem in networking is the minimum congestion problem in data-center networks. These networks are typically built after *Clos networks*, which have the key property that there is always a routing with congestion at most 1. That is, if the demands are limited by the edge capacities *to* the network, then they can always be routed such that they are not limited by the edge capacities *inside* the network. In this sense, Clos networks acts as a macro-switch connecting all sources to all destinations. However, this property is premised splitting flow arbitrarily, which is unrealistic. Thus, a natural question arises: if *unsplittable* flow is required, then do Clos networks still enjoy the same properties as they do for splittable flow? Or, more specifically: are there always low-congestion routings with unsplittable flow provided that low-congestion routings with splittable flow exist?

In this paper, we present the first non-trivial results on the minimum congestion unsplittable flow problem in data-center networks. First, we show that for some sets of commodities the minimum congestion is at least $3/2$, and, furthermore, that it is NP-hard to approximate the minimum congestion by a factor less than $3/2$. That is, with unsplittable flow, Clos networks *do not* acts as a macro-switch. Second, in our main result, we present a polynomial-time algorithm that guarantees a congestion of at most $9/5$ for any set of commodities, while also providing a $9/5$ approximation of the minimum congestion. Notably, this algorithm breaks through the barrier of 2 for congestion and approximation implicit in existing heuristics. Last, shifting to the online setting, we demonstrate that no online algorithm (deterministic or randomized) can approximate the minimum congestion by a factor less than 2, thus establishing a strict separation between the offline and the online settings.

Contents

1 Introduction

Unsplittable flow problems are a classic family of combinatorial optimization problems. The input to an unsplittable flow problem consists of: a (directed or undirected) graph $G = (V, E)$ with edge capacities $\{c(e)\}_{e \in E}$, $c(e) \in \mathbb{R}^+$, and a set $\mathcal{F}$ of commodities, each commodity $f \in \mathcal{F}$ characterized by a source-destination pair $(s(f), t(f))$ and a demand $d(f)$, $s(f), t(f) \in V$ and $\text{dem}(f) \in \mathbb{R}^+$. A solution is a *single-path routing* for the commodities, that is, an assignment from each demand f to a single $s(f)-t(f)$ path $P(f)$ on which $\text{dem}(f)$ demand is routed. In this paper, we are interested in the minimum congestion problem with unsplittable flow, where the objective is to minimize the maximum congestion on any edge. The *congestion* $\gamma(e)$ of a edge is the maximum factor by which the total demand routed through that edge exceeds the edge capacity, $\gamma(e) \coloneqq \sum_{f \in \mathcal{F}: e \in P(f)} {}^{\text{dem}(f)}/_{c(e)}$. While with splittable flow the minimum congestion problem is just the maximum concurrent flow problem and can be solved in polynomial time [1], with unsplittable flow the problem becomes NP-hard.

Despite being significantly more difficult than its splittable counterpart, the minimum congestion problem with unsplittable flow is well understood. In general graphs, an early and famous application of randomized rounding yields an $O\left({}^{\log n}/_{\log \log n}\right)$-approximation ratio [2] (where $n = |V|$), whereas a more recent and powerful application in the bounded path length setting yields an $O\left({}^{\log l}/_{\log \log l}\right)$-approximation ratio [3] (where l is the maximum length of any source-destination path). Conversely, there are graphs for which the integrality gap matches the $O\left({}^{\log n}/_{\log \log n}\right)$-approximation ratio [4], and it is hard to approximate the minimum congestion by a factor smaller than $\Theta\left({}^{\log n}/_{\log \log n}\right)$ under standard complexity assumptions [5]. In order to break through these lower bounds, the problem has been investigated in several special classes of graphs [6–10] and several special classes of demands [11–13]. For example, constant factor approximations have been shown for ring graphs [6–8], and single-source demands [11, 12].

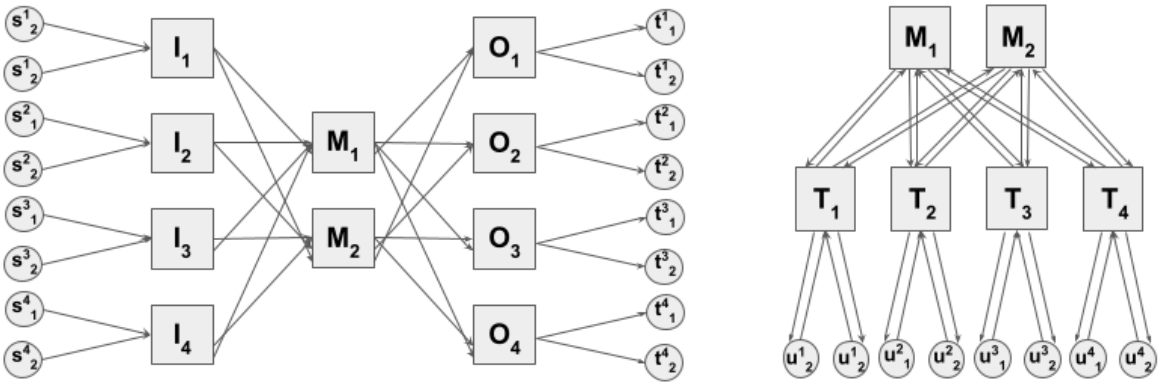
While these are important results, the original motivation for studying unsplittable flow problems comes from problems in computer networking where practical considerations require flow to be unsplittable. These problems included the virtual circuit problem [14, 15], the routing and wavelength problem [16–18], and the wire routing problem in VLSI design [19]. Today, the most relevant unsplittable flow problem in networking is the minimum congestion problem in data-center networks [20–25]. These networks are often built as *Clos networks* with uniform link capacities [25–30], and flow is unsplittable since implementing flow splitting requires extensive modifications to existing transport protocols and remains unverified in large-scale [25, 29–34].

Definition 1.1 ([35, 36]). *A (unfolded) Clos network $C_{N,R}$ is parameterized by positive integers N and R, and designates the directed graph $G = (V, E)$ with the following vertex and edge sets (see Figure 1a).*[1]

- ***Vertex set.*** *The vertices V are partitioned into 5 layers V_1 through V_5. In V_1 (respectively V_5), there are $N \times R$ vertices, and each vertex is a* source (destination) *and is denoted by s_i^j (t_i^j), $i \in [R]$ and $j \in [N]$. The vertices in $V_1 \cup V_5$ are also called* servers. *In V_2 (respectively V_4), there are R vertices, and each vertex called an* input (output) switch *and is denoted by I_i (O_i), $i \in [R]$. The vertices in $V_2 \cup V_4$ are also called Top-of-Rack (ToR) switches. In V_3, there are N vertices, and each vertex called a* middle switch *and is denoted by M_m, $m \in [N]$.*

- ***Edge set.*** *The edges E are from V_i to V_{i+1} for all $i \in \{1, 2, 3, 4\}$. For all sources $s_i^j \in V_1$, there is an edge $s_i^j I_i$, and for all destinations $t_i^j \in V_5$, there is an edge $O_i t_i^j$. There are complete bipartite graphs*

[1] In practice, data-center networks are built as *folded* Clos networks, where we start from an unfolded Clos network and then fold it around the middle switches to get a new directed graph (see Figure 1b). The sources and destinations with the same indexation correspond to the same physical server, and, similarly, the input and output switch with the same indexation correspond to the same physical ToR switch. For ease of exposition, we assume unfolded Clos networks throughout the paper, with all results carrying over from unfolded to folded Clos networks.



(a) *A unfolded Clos network with $R = 4$ input (output) switches, and $N = 2$ middle switches.*

(b) *A folded Clos network with $R = 4$ input (output) switches, and $N = 2$ middle switches.*

Figure 1: *A unfolded and a folded Clos network. Circles symbolize servers, and squares symbolize switches. For easy of exposition, we assume unfolded Clos networks, with all results carrying over to folded Clos networks.*

between V_2 and V_3 and between V_3 and V_4: for all $I_i \in V_2$ and $M_m \in V_3$, there is an edge $I_i M_m$, and for all $M_m \in V_3$ and $O_i \in V_4$, there is an edge $M_m O_i$.

Therefore, N designates the number of servers per ToR switch, which equals the number of middle switches, and R the number of input (output) switches.

The input to the minimum congestion problem in data-center networks consists of: a Clos network with unit edge capacities, and a set of *doubly sub-stochastic* demands, meaning that each source sends at most unit demand and each destination receives at most unit demand, $\sum_{f \in \mathcal{F}: s(k) = s_i^j} \text{dem}(f) \leq 1$ and $\sum_{f \in \mathcal{F}: t(f) = t_i^j} \text{dem}(f) \leq 1$ for all $i \in [R]$ and $j \in [N]$. In other words, the demand leaving a source or entering a destination is limited by the capacity of the edge leaving that source or entering that destination. While this assumption limits the minimum congestion, it is easy to show via a simple scaling argument that it implies no loss of generality concerning its approximation: if there is a ρ-approximation algorithm for the problem when the demands are doubly sub-stochastic, then there is also a ρ-approximation algorithm for when they do not.

Despite its practical importance, the minimum congestion problem in data-center networks is only fully understood for splittable flow: the minimum congestion is at most 1, and can be found in polynomial time by dividing the flow of each demand uniformly over all source-destination paths [23]. This unit congestion carries a special significance: if the minimum congestion is at most 1, then a Clos network emulates a single switch connecting all sources to all destinations, or a *macro-switch* [20, 27, 37–39]. We say that a Clos network emulates a macro-switch if any demands achievable in the macro-switch are also achievable in the network. A macro-switch is a highly desirable abstraction, since it black boxes the inside of a network from a performance modeling standpoint: the demand is limited by the edge capacities leaving and entering the network, not by those inside the network [37, 40–42]. In fact, it was the ability of Clos networks to emulate a macro-switch that motivated their widespread adoption in modern data-centers.

However, with unsplittable flow (which is the more realistic setting), the minimum congestion problem is poorly understood. For *permutation demands*, meaning that each source sends flow to a single destination and each destination receives flow from a single source, the minimum congestion is still at most 1 (corresponding to an edge-disjoint routing), and can still be found in polynomial time [36, 43]. In particular, routing permutation demands with congestion at most 1 equates to edge coloring a bipartite graph with bounded degree into as many colors, which is ensured by Kőnig's edge-coloring theorem [44].

In contrast, for general *doubly sub-stochastic demands*, no non-trivial result is known. Of course, the application of randomized rounding to minimizing congestion in general graphs [2] guarantees that the minimum congestion is at most $O\left(\log n/\log\log n\right)$ and can be approximated by a factor of $O\left(\log n/\log\log n\right)$. Moreover, since every source-destination path in a Clos network has length 4, the application of the same technique in the bounded path length setting improves these factors to $O(1)$, for some large constant. Finally, several heuristics for routing in Clos networks have been studied in simulation and deployed in practice [20,21,23,25,45,46]. While these heuristics further improve upon the previous factors, it is easy to show that they only ensure that the minimum congestion is at most 2 and can be approximated by a factor of 2. In summary, despite the fact that the main motivation for unsplittable flow is in networking, and that the most relevant networking problem for unsplittable flow is minimizing congestion in Clos-based data-center networks, very little is known about unsplittable flow in this special class of graphs.

In light of this mismatch, we ask the following fundamental questions concerning the existence and efficient computation of minimum congestion routings with unsplittable flow in Clos networks:

Does unsplittable flow have the same properties as splittable flow in Clos networks? In particular:

(Q1) *Is the minimum congestion with unsplittable flow at most* 1 *for all sets of doubly sub-stochastic commodities? If not, how close to* 1 *is it?*

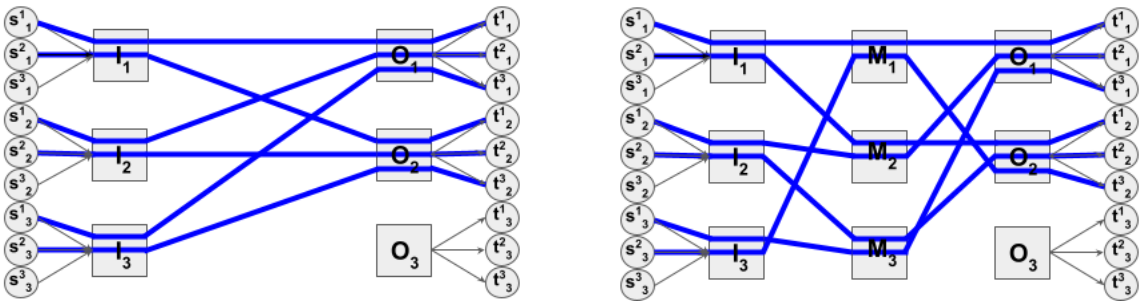
(Q2) *Is the minimum congestion with unsplittable flow computable in polynomial-time? If not, how well can it be approximated?*

2 Summary of Results and Techniques

We present the first non-trivial results on the minimum congestion problem with unsplittable flows in data-center networks. In all results, we assume commodities are doubly sub-stochastic. First, we answer the questions $\mathbf{Q}_1$ and $\mathbf{Q}_2$ in the negative: unsplittable flow does not act the same as splittable flow in Clos networks. In particular, we prove that there are sets of commodities for which the minimum congestion is at least $3/2$, and, furthermore, that it is NP-hard to approximate the minimum congestion by a factor smaller than $3/2$. Second, as our main result, we give a new algorithm that returns a routing with congestion at most $9/5$ for all sets of commodities while also approximating the minimum congestion by a factor at most $9/5$. In other words, we show that, while Clos networks fail to emulate a macro-switch (thus contradicting the original motivation for their widespread deployment), there are approximation algorithms that at least achieve relatively low congestion. Notably, our algorithm achieve lower (worst-case) congestion than either the current theoretical state of the art or the current heuristics used in practice.

A related setting that is relevant in practice is the *online* setting, where the commodities are revealed one at a time and each commodity must be routed irrevocable before receiving the next one. In this setting, we prove that no online (deterministic or randomized) algorithm can have competitive ratio less than 2. Combined with our offline upper bound, this implies a strict separation between the offline and the online settings: while in the offline our new algorithm approximates the minimum congestion by a factor less than $9/5$, in the online no algorithm can approximate it by a factor less than 2.

While our results concern Clos networks with five layers, which are the simplest and most studied members of this special class of graphs and thus the natural starting point for this investigation, hyper-scale data-centers are often built using Clos networks with seven or more layers. Therefore, we close by briefly discussing how our results carry over to Clos networks with an arbitrary number of layers.



(a) *The funnel gadget without routing.* ***(b)*** *A minimum congestion routing for the funnel gadget.*

Figure 2: *The funnel gadget underlying the lower bounds in the offline setting in a Clos network with $N = 3$ middle switches. Lines symbolize commodities from source to destination.*

2.1 Offline: Lower Bounds on Congestion and Approximation

As discussed in Section 1, a fundamental property of Clos networks is that, if demands are *integral* (and thus form a permutation due to the assumption of double sub-stochastic demands), then the minimum congestion is at most 1 and a minimum congestion routing can be found in polynomial time [36, 43]. In contrast, we show that as soon as the demands become half-integral, the minimum congestion can be $3/2$ and a minimum congestion routing cannot be found in polynomial time.

Theorem 2.1. *In a Clos network, there is a set of commodities with demand 1 or $1/2$ for which the minimum congestion is $3/2$.*

Theorem 2.2. *For a set of commodities with demand 1 or $1/2$ in a Clos network, deciding whether the minimum congestion is at most 1 is NP-complete.*

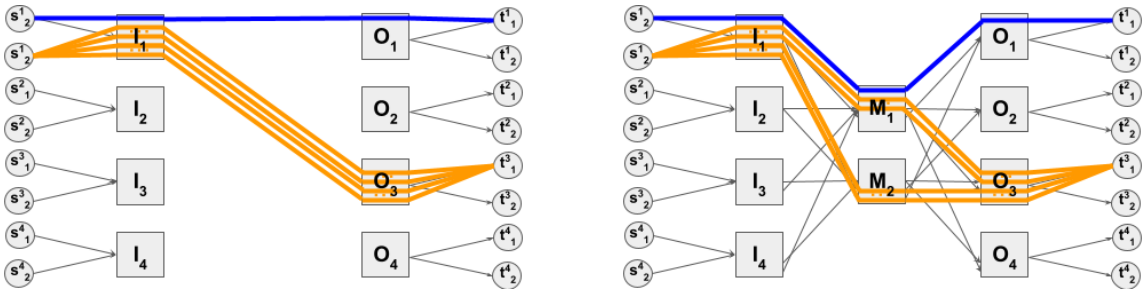
The latter theorem implies a hardness of approximation result.

Corollary 2.3. *No polynomial-time algorithm can approximate the minimum congestion in a Clos network by a factor less than $3/2$ unless $P = NP$.*

The key idea for proving both theorems is a set of commodities with flow 1 that we call a *funnel gadget*. In a Clos network with N middle switches, the *funnel gadget* funnels commodities from N input switches to $N-1$ output switches; specifically, it maps each of $N-1$ commodities leaving each input switch to a different output switch (see Figure 2a). The funnel gadget has the property that, in any routing with congestion 1, the middle switch to which no commodity is assigned is different for each input switch; consequently, any additional commodities leaving each of the input switches are assigned to different middle switches (see Figure 2b). In the first theorem, the funnel gadget is used to construct a new set of commodities for which no routing with congestion 1 exists by adding commodities with flow $1/2$ from each of the input switches to an extra output switch such that all middle switches are blocked. In the second theorem, the funnel gadget is used to establish a reduction from the 3-edge coloring problem.

2.2 (Main Result) Offline: Upper Bound via a New Routing Algorithm

As also discussed in Section 1, the best known upper bound on the minimum congestion and its approximation by a polynomial-time algorithm is 2, and it is obtained from the analysis of existing heuristics. As our main result, we design a new algorithm that provides the first non-trivial upper bounds on congestion and approximation, breaking through the barrier of 2 implicit in prior work.



(a) *The commodities without routing. The blue commodity has demand 1, whereas the orange commodities have demand ϵ, for some small $\epsilon > 0$.*

(b) *The routing returned by the Melen-Turner algorithm for the commodities. The orange commodities are divided uniformly divided over the middle switches.*

Figure 3: *A worst-case instance for the Melen-Turner algorithm in a Clos network with $N{=}2$ middle switches.*

Theorem 2.4. *There is a polynomial-time algorithm that in a Clos network returns a routing with congestion at most $9/5$ for all sets of commodities, and approximates the minimum congestion by a factor of at most $9/5$.*

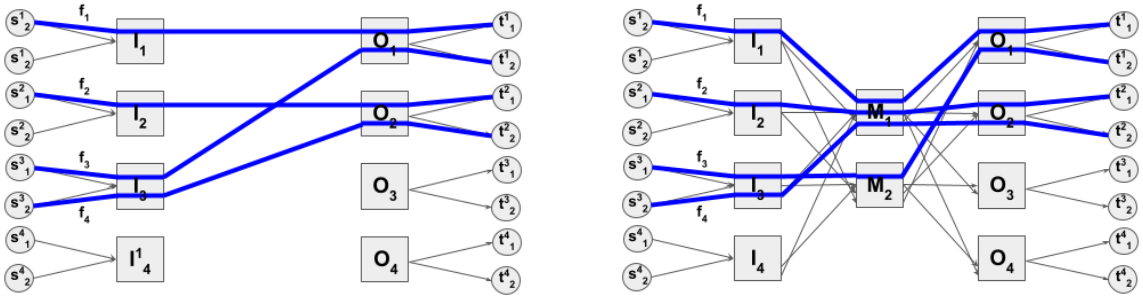
To achieve this property, our algorithm combines the following two heuristics in a new way.

(1) The *Melen-Turner algorithm* [47] yields a routing for the demands satisfying the following property: for every ToR switch, the demands incident to that switch between the heaviest and the N-th heaviest are assigned to different middle switches, those between the $N{+}1$-th heaviest and the $2N$-th heaviest are assigned to different middle switches, and so on. The algorithm attains this property by first considering a new Clos network where there are multiple copies of each ToR switch, and mapping the commodities to copies of its switches such that, for every ToR switch, the commodities incident to that switch between the heaviest and the N-th heaviest are mapped to the first copy, those between the $N{+}1$-th heaviest and the $2N$-th heaviest are mapped to the second copy, and so on. Critically, the demands in the new network form a permutation. Then, the algorithm finds a edge-disjoint routing for the commodities in the new network, which equates to a routing in the original network with the desired property.

(2) The *Sorted-Greedy algorithm* [20] sorts the commodities in decreasing order of their demands and assigns each demand to a path of minimum congestion; the congestion of a path is the maximum demand flow routed on any of its edges.

On the one hand, the pitfall of the Melen-Turner algorithm is that it evenly splits low demand commodities over all middle switches without accounting for the presence of high demand commodities. In Figure 3, there is a single commodity with demand 1, and multiple commodities with demand ϵ, for some arbitrary small $\epsilon > 0$, such that their total demand is $N{-}1$. Clearly, there is a routing with congestion 1. However, the algorithm uniformly distributes the commodities with demand ϵ over the middle switches, such that their total demand traversing each edge is $N{-}1/N$. Consequently, the congestion of the left edge traversed by the commodity with demand 1 is $1 + N{-}1/N$, which approaches 2 as N grows large.

On the other hand, the pitfall of the Sorted-Greedy algorithm is that it fails to route high demand commodities according to a edge-disjoint routing. In the Figure 4, there are four commodities with demand 1. Consider the commodities in increasing order of their indices, and tie-breaking among the middle switches in favor of that with the lowest index. Clearly, there is again a routing with congestion 1. However, the algorithm assigns commodities f_1 and f_2 to the first middle switch, and the commodity f_3 to the second middle switch. Consequently, the congestion of the right edge traversed by the commodity f_4 is 2.



(a) *The commodities without routing. The index of each commodity is annotated next to its source. All commodities for the commodities. have demand* 1.

(b) *The routing returned by the Sorted-Greedy algorithm*

Figure 4: *A worst-case instance for the Sorted-Greedy algorithm in a Clos network with* $N=2$ *middle switches.*

These examples suggest that the high demand commodities should be distributed as uniformly as possible over the middle switches, as in the Melen-Turner algorithm, but the low demand commodities should be routed on paths that in the aftermath have low congestion, as in the Sorted-Greedy algorithm. Therefore, the key idea of our algorithm is that, by carefully interpolating between these two algorithms, we can mitigate their pitfalls, and thus arrive at a better bound than they achieve individually.

Our algorithm routes a set of commodities in two phases bridged by a threshold on the congestion of each phase. In the first phase, a subset of the commodities is routed via the Melen-Turner algorithm with congestion at most the threshold. In the second phase, the remaining commodities are routed via the Sorted-Greedy algorithm without increasing the congestion beyond the threshold. By setting the threshold to $9/5$, the algorithm returns a routing with congestion at most $9/5$.

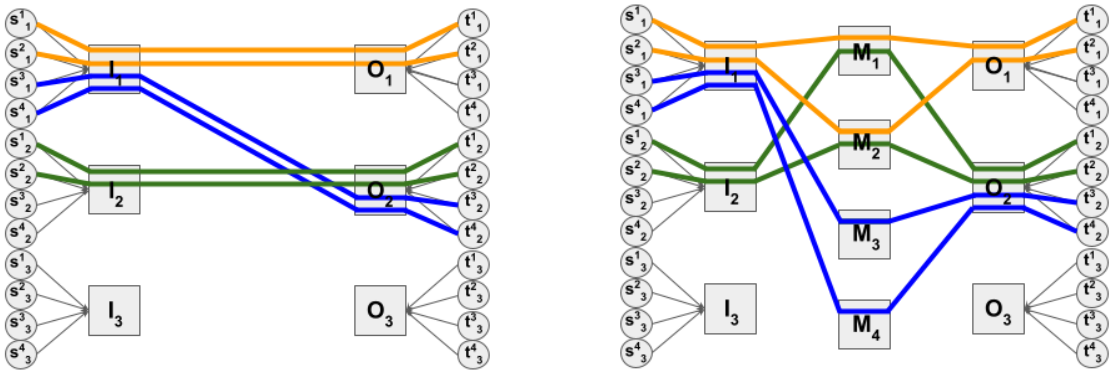
As defined, our algorithm returns a routing with congestion at most $9/5$, but fails to approximate the minimum congestion by a factor of $9/5$. The reason is that the minimum congestion is not known a priori: if we knew the minimum congestion, then the algorithm could be easily fixed by setting the threshold to $9/5$ times the minimum congestion. Standard guess-and-double techniques would allow us to "guess" the minimum congestion accurately enough to achieve a $9/5 + \epsilon$ approximation in $\text{poly}(n, \log 1/\epsilon)$ time. However, it is possible to modify our algorithm to achieve a true $9/5$-approximation via a more nuanced bridging of the two phases. The key idea is, in the first phase, to route via the Melen-Turner algorithm a subset of the commodities, which critically includes all those with demand at least $1/3$ times the minimum congestion (despite not knowing the minimum congestion), with congestion at most $9/5$ times a suitable lower bound on the minimum congestion (rather than its unknown value). Consequently, in the second phase, it possible to route the remaining commodities via the Sorted-Greedy algorithm without increasing congestion beyond $9/5$ times the minimum congestion.

2.3 Online: Lower Bounds on Congestion and Approximation

In the offline setting, we have repeatedly mentioned that a fundamental property of Clos networks is the existence of a routing with congestion 1 for all sets of commodities with integral demands [43]. On the contrary, in the online setting, we establish that no deterministic algorithm can ensure this property. Furthermore, we prove that randomizing the routing choices does not help.

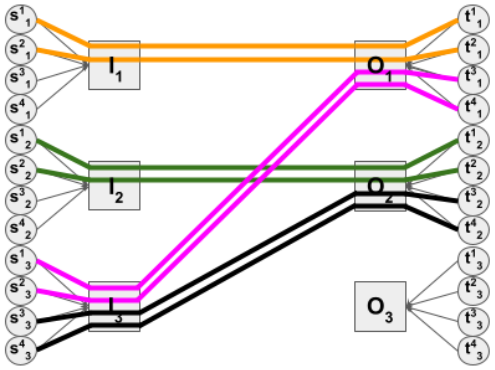
Theorem 2.5. *For every online algorithm (deterministic or randomized), there is a sequence of commodities with demand* 1 *for which the congestion of the algorithm is at least* 2.

This theorem implies a lower bound on the competitive ratio of any online algorithm.

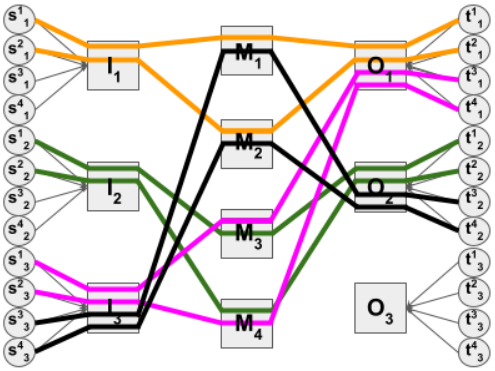


(a) *Sequence X.*

(b) *A minimum congestion routing for X.*



(c) *Sequence Y.*



(d) *A minimum congestion routing for Y.*

Figure 5: *The two sequences of commodities underlying the lower bound in the online setting in a Clos network with 3 middle switches. The sequences are denoted by $X = (X_1, X_2)$ and $Y = (Y_1, Y_2)$, with $X_1 = Y_1$ and $X_2 \neq Y_2$, where subsequence X_1 consists of two commodities (I_1, O_1) (in orange) and two commodities (I_2, O_2) (in green), X_2 of two commodities (I_1, O_2) (in blue), and Y_2 of two commodities (I_3, O_1) (in purple) and two commodities (I_3, O_2) (in black).*

Corollary 2.6. *No online algorithm (deterministic or randomized) can have competitive ratio less than* 2.

The proof of the theorem is divided into two parts, the former showing the result for deterministic algorithms, and the latter generalizing it for randomized algorithms. The first part designs two sequences of commodities that agree in their prefixes but disagree in their suffixes (see Figure 5). The key property of the sequences is that in a routing with congestion 1 the routing of their common prefix differs; thus, any deterministic algorithm that returns a routing with congestion 1 for one sequence returns a routing with congestion 2 for the other. The second part designs 2^S supersequences consisting of S independent sequences, each corresponding to one of the previous sequences, for some S linear in the number R of input (output) switches. Therefore, from the key property of the sequences, any deterministic algorithm returns a routing with congestion 2 for at least $2^S - 1$ supersequences; thus, the expected congestion of an optimal deterministic algorithm when supersequences are chosen uniformly at random over the 2^S supersequences is at least $2 - 1/2^s$. Then, the theorem follows from the application of Yao's Minimax Principle [48, 49].

These lower bounds are complemented by the fact that the *Unsorted Greedy algorithm* yields an upper bound of 3 on congestion and approximation, as it easy to show.

2.4 Clos networks with arbitrary number of layers

The construction of a Clos network can be generalized to an arbitrary number of layers. At a high level, a Clos network with $2K + 3$ layers, henceforth called a K-Clos network, $K>1$, corresponds to a Clos network with 5 layers where each middle switch is replaced by a Clos network with $2K + 1$ layers. (See Appendix A for a formal definition.) It is not hard to see that our lower bounds of $3/2$ on congestion and approximation in 1-Clos networks carry over to K-Clos networks. Intuitively, the reason is that at best each of the $(K-1)$-Clos networks connecting input to output switches in a K-Clos network acts as a middle switch of a 1-Clos network. On the other hand, our upper bounds of $9/5$ do not carry over to K-Clos networks. This is somewhat less intuitive, but the reason is that the natural generalization of our algorithm would yield a routing where the demands that form the input to the $(K-1)$-Clos networks are no longer doubly sub-stochastic.

Similarly to 1-Clos networks, there are no non-trivial upper bounds on congestion and approximation known for K-Clos networks. Since every source-destination path in a K-Clos network has length $2K+2$, the application of randomized rounding in the bounded path length setting [3] yields upper bounds of $O(\log K/\log\log K)$. In practice, however, data-center operators typically want to deploy heuristics that, despite possibly having worst performance guarantees, are simpler to implement and understand. We show that it is possible to get the best of both worlds by establishing that the natural generalization of the Melen-Turner algorithm [47] yields $O(\log K)$ congestion and approximation in K-Clos networks.

Proposition 2.7. *The natural generalization of the Melen-Turner algorithm in a K-Clos network returns a routing with congestion at most $O(\log K)$ while approximating the minimum congestion by a factor of at most $O(\log K)$ for all sets of commodities.*

Proving either super-constant lower bounds or designing an algorithm that achieves a constant-factor approximation in K-Clos networks is an attractive open question that is left to future work.

2.5 Roadmap

In Section 3, we review related work. In Section 4, we present a formal description of the minimum congestion routing problem with unsplittable flow in data-center networks. In Section 5, we design and analyze the new routing algorithm. In Sections 6 and §7 we prove the lower bounds on congestion and approximation in the offline and online settings, respectively. In Section 8, we conclude the paper and discuss open questions. In Appendix A, we analyze the natural generalization of the Melen-Turner algorithm to Clos networks with an arbitrary number of layers.

3 Related Work

Minimizing congestion in Clos networks. Chiesa, Kindler, and Schapira [23] also investigate the minimum congestion problem with unsplittable flows in Clos networks. However, their setting differs from ours in that they assume that the graph modeling a data-center network is the *undirected* graph obtained from a folded Clos network (see Figure 1b) by replacing the two edges with uniform capacity between pairs of vertices in consecutive layers with a single edge with twice the capacity. Since in almost every wired network deployed data transmission is independent between the two directions of a link [50–53], our model is the more accurate representation of a data-center network.

As a result of their modeling choice, their results are more pessimistic than ours. First, they show that with integer demands the question of deciding whether there is a routing with congestion at most 1 is NP-complete, and thus conclude that it is NP-hard to approximate a minimum congestion routing by a factor less than 2; in our model, it is known that this problem is in P [43], and our new algorithm shows

that the minimum congestion can be approximated by a factor of $9/5$. Second, they give a local-search algorithm that, while there is a commodity and a path such that re-assigning that commodity to that path decreases the congestion of the routing, re-assigns the commodity; in our model, it can be shown that their algorithm approximates a minimum congestion routing by a tight factor of 3.

Multirate rearrangeability in Clos networks. The multirate rearrangeability problem with unsplittable flow in Clos networks is the closest problem to the minimum congestion problem. The setting consists of: a Clos network whose number of servers per ToR and of input (output) switches is *fixed*, but whose number of middle switches is *variable*; and a set of commodities with doubly sub-stochastic demands. The goal is to find a routing with congestion at most 1 while minimizing the number of middle switches used.

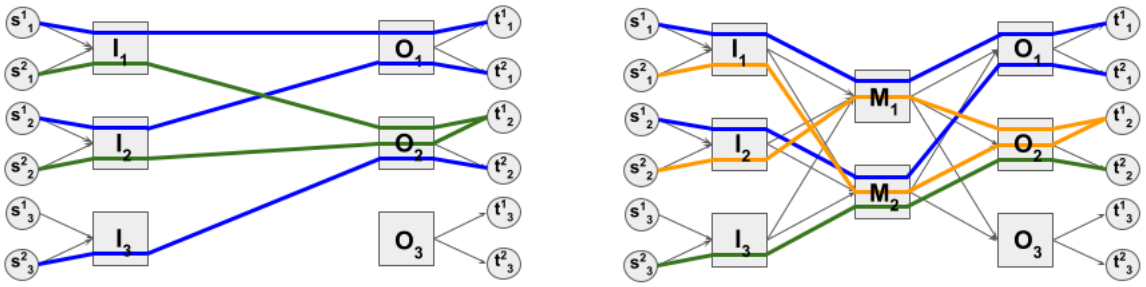
The best known algorithm is given by Khan and Singh [54], and establishes that it is possible to route every set of commodities with congestion at most 1 using at most $\lceil 20/9\, N \rceil$ middle switches, where N is the number of sources per input switch. The algorithm fixes $\lceil 20/9\, N \rceil$ middle switches, and routes a set of commodities in two phases. The first phase considers a subset of the commodities with demand at least $1/10$, and finds an edge-disjoint routing for them. The second phase sorts the commodities in decreasing order of demands, and assigns each commodity to an arbitrary middle switch for which the congestion does not exceed 1. Our algorithm draws inspiration from the two-phase approach of this algorithm. The best known lower bound is given by Ngo and Vu [55], and shows that there are sets of commodities for which every routing with congestion 1 uses at least $\lceil 5/4\, N \rceil$ middle switches. While the underlying construction yields a lower bound of $6/5$ on worst-case congestion, we improve this bound to $3/2$.

4 Problem Setting

We formalize the minimizing congestion routing problem with unsplittable flow in data-center networks. The input to the minimum congestion routing problem in a data-center network is: (1) A Clos network $C_{N,R}$ with unit edge capacities. (2) A set $\mathcal{F}$ of commodities. Each commodity f maps to a source-destination server pair, and a positive *demand* $\mathrm{dem}(f)$. If clear from the context, then a commodity may be identified by its input-output switch pair. Given a commodity $f \in \mathcal{F}$, let $i(f)$ and $s(f)$ be the input switch and the source server that f leaves, respectively, and $j(f)$ and $t(f)$ be the output switch and the destination server that f enters, respectively. The demands satisfy the following assumption: the aggregate demand over all commodities leaving a source and entering a destination is at most 1 (so that if commodities were arbitrary splittable there would be a routing with congestion 1 for every set of commodities, and Clos networks could replicate a macro-switch):

$$\sum_{\substack{f\in\mathcal{F}: i(f)=i \\ s(f)=k}} \mathrm{dem}(f) \le 1 \quad \text{and} \quad \sum_{\substack{f\in\mathcal{F}: j(f)=i \\ t(f)=k}} \mathrm{dem}(f) \le 1 \text{ for all } i \in [R], k \in [N].$$

A solution to the problem is a *routing* r for the set $\mathcal{F}$ of commodities, which is an assignment from each commodity f to a source-destination path $r(f)$; in fact, since there is a bijection between source-destination paths and middle switches, a routing is an assignment to a middle switch. An optimal solution to the problem minimizes over all routings the congestion of each routing; we call it a *minimum congestion routing*. The congestion of a routing is the maximum congestion over all edges between ToR and middle switches, where the congestion of a edge is the total demand over all commodities traversing the edge. Letting $\mathcal{R}$ denote the set of all routings, the congestion of a minimum congestion routing, denoted by



(a) *The commodities without routing.* ***(b)*** *A minimum congestion routing in the Clos network.*

Figure 6: *A set of commodities in the Clos network $C_{2,3}$.*

$OPT(\mathcal{F})$, writes:

$$OPT(\mathcal{F}) \coloneqq \min_{r \in \mathcal{R}} \max_{\substack{i \in [R] \\ m \in [N]}} \max \Bigg\{ \overbrace{\sum_{\substack{f \in \mathcal{F}: i(f)=i \\ r(f)=m}} \mathrm{dem}(f)}^{\text{congestion of } I_i M_m}, \overbrace{\sum_{\substack{f \in \mathcal{F}: j(f)=i \\ r(f)=m}} \mathrm{dem}(f)}^{\text{congestion of } M_m O_j} \Bigg\}}^{\text{congestion of } r}$$

If clear from the context, then the argument $\mathcal{F}$ is omitted.

The minimum congestion routing problem is illustrated in Figure 6. In Figure 6a, we show a set of commodities in the Clos network $C_{2,3}$ composed of three types of commodities, each identified with a different color in the figure:

- *Type 1 (in blue):* There is one commodity (s_1^1, t_1^1) and one commodity (s_2^1, t_1^2), both with demand 1.
- *Type 2 (in orange):* There is one commodity (s_1^2, t_2^1) and one commodity (s_2^2, t_2^1), both with demand $1/2$
- *Type 3 (in green):* There is one commodity (s_3^2, t_2^2) with demand 1.

In Figure 6b, we show a minimum congestion routing for the commodities. Every edge except for M_2O_2 is traversed by at most one commodity, and thus has congestion at most 1. Edge M_2O_2 is traversed by the type 3 commodity from I_3 to O_2 and the type 2 commodity from I_2 to O_2, and thus has congestion $3/2$. Therefore, the routing of the figure has congestion $3/2$.

The next theorem formalizes the fundamental property of Clos networks concerning the special case where the demands form a permutation, meaning that there is at most one commodity leaving each source and entering each destination. We refer to this theorem repeatedly throughout the remainder of the paper.

Theorem 4.1 ([43,44]). *Consider a Clos network $C_{N,R}$. For every set of commodities such that the number of commodities per source and per destination is at most one, there is a edge-disjoint routing of the commodities, and one such a routing can be found in polynomial-time.*

The backbone of this property is the existence of a bijection between a set of commodites in a Clos network with N servers per ToR switch and a bipartite multi-graph, and between a routing for these commodities and an edge-coloring of the bipartite multi-graph using N colors. The left and right vertex sets of the bipartite multi-graph corresponds to the input and output switches in the network, each left vertex corresponding to an input switch and each right vertex to an output switch. The edge set corresponds to the commodity set, each edge from a left to a right vertex corresponding to a commodity from the related

input switch to the related output switch. An edge coloring of the bipartite multi-graph using N colors corresponds to a routing of the commodities, each edge colored m if and only if the related commodity is assigned to the m-th middle switch, $m \in [N]$. Notably, there is a edge-disjoint routing for the commodities precisely when there is a proper edge-coloring of the bipartite multi-graph using N colors.

5 (Main Result) Offline: Upper Bounds via a New Routing Algorithm

We introduce a new algorithm for the minimizing congestion in Clos networks. In §5.1, we design the algorithm, and, in §5.2, we analyze it to show that the algorithm approximates a minimum congestion routing by a factor at most $9/5$. A straightforward generalization of the analysis shows that the algorithm also returns a routing with congestion at most $9/5$.

5.1 Designing the Algorithm

The new routing algorithm is presented as Algorithm 1. Given a Clos network $C_{N,R}$, and a set $\mathcal{F}$ of commodities, the algorithm routes the commodities in two phases. Let $1/Q$ be a demand, p be an approximation factor, L be a lower bound on the minimum congestion OPT, all of which are specified in the next section. In Phase 1, the algorithm finds a routing for a subset of the commodities that includes all commodities with demand greater than $1/Q \times OPT$ via a edge-coloring based procedure with congestion at most $p \times L$. In Phase 2, the algorithm finds a routing for the remaining commodities via a greedy procedure without increasing congestion beyond $p \times OPT$.

Algorithm 1 The input of the algorithm is a set $\mathcal{F}$ of commodities in the Clos network $C_{N,R}$ together with parameters P and Q, where P is a non-negative real and Q is a positive integer. The output is a routing r for $\mathcal{F}$. Variable $c(I_iM_m)$ ($c(M_mO_j)$) maintains the congestion of edge I_iM_m (M_mO_j).

$\mathcal{F}_1 \coloneqq$ UpholdProperties$(\mathcal{F}, P, Q)$ ▷ *Phase 1:*
$r(\mathcal{F}_1) \coloneqq$ EdgeDisjointRouting$(\mathcal{F}_1, C_{N,R\times K}))$
for $i \in [R], m \in [N]$ **do** ▷ *Phase 2:*
    $c(I_iM_m) \coloneqq \sum_{\substack{f\in\mathcal{F}_1: i_{(f)}=i,\\ r_{(f)}=m}} \text{dem}(f)$; $c(M_mO_i) \coloneqq \sum_{\substack{f\in\mathcal{F}_1: j_{(f)}=i,\\ r_{(f)}=m}} \text{dem}(f)$
end for
$\mathcal{F}_2 \coloneqq \mathcal{F} \setminus \mathcal{F}_1$
for $j = 1, 2, \cdots, |\mathcal{F}_2|$ **do**
    $r(f_{i_j}) \coloneqq \arg\min_{m\in[N]} \max\{ c(I_{i(f_{i_j})}M_m), c\,(M_mO_{j(f_{i_j})})\}$
    $c(I_{i(f_{i_j})}M_{r(f_{i_j})}) \mathrel{+}\coloneqq \text{dem}(f_{i_j})$; $c(M_{r(f_{i_j})}O_{j(f_{i_j})}) \mathrel{+}\coloneqq \text{dem}(f_{i_j})$
end for
return r

5.1.1 Phase 1

The first phase of the algorithm is divided into two sub-phases. Phase 1.a obtains a new instance of the problem: a new Clos network, and a subset of the commodities that includes all commodities with demand greater than $1/Q \times OPT$ mapped to the new network. Phase 1.b routes these commodities with congestion at most $P \coloneqq p \times L$ via an edge-coloring of the bipartite graph corresponding to the new Clos network.

Phase 1.a. The new Clos network is denoted by $C_{N,R\times K}$, with $K = \lceil F/N \rceil$ and F the maximum number of commodities incident to a ToR switch in the original one. The new network is obtained from

the original one by creating K copies of each ToR switch. The k'th copy of the i'th input switch is denoted by I_i^k, and the l'th copy of the j'th output switch by O_j^l.

The subset of the commodities mapped to the new instance is denoted by $\mathcal{F}_1$. Consider that commodities in $\mathcal{F}$ are indexed in decreasing order of demands, $\mathcal{F} \coloneqq \{f_1, f_2, \ldots, f_{|\mathcal{F}|}\}$ with $i \leq j$ implying $\text{dem}(f_i) \geq \text{dem}(f_j)$. The subset $\mathcal{F}_1$ is obtained from $\mathcal{F}$ and mapped to $C_{N,R\times K}$ as follows:

(1) The incidence of the commodities in ToR switches in the original network is respected in the new network, that is, if a commodity in the original network leaves an input switch I_i and enters an output switch O_j, then in the new network it leaves a copy of I_i and enters a copy of O_j.

(2) The assignment from commodities to copies of input-output switch pairs in the new network is such that $\mathcal{F}_1$ is a maximal subset of $\mathcal{F}$ satisfying the properties P1 through P3; in particular, it is the maximal such subset returned by the procedure UpholdProperties (line 1 of Algorithm 1). The properties and the procedure are introduced next.

Properties. Given a commodity $f \in \mathcal{F}_1$, let $k(f)$ be the copy of its input switch that f leaves, and let $l(f)$ be the copy of its output switch that f enters. Denote by ${}^{+}N_i^k$ the number of commodities in $\mathcal{F}_1$ leaving I_i^k, and by ${}^{+}D_i^k$ the maximum demand over all commodities in $\mathcal{F}_1$ leaving I_i^k,

$$ {}^{+}N_i^k \coloneqq |\{f \in \mathcal{F}_1 \,|\, i(f) = i \text{ and } k(f) = k\}| \text{ and } {}^{+}D_i^k \coloneqq \max_{\substack{f\in\mathcal{F}_1: i(f)=i \\ k(f)=k}} \text{dem}(f). $$

Likewise, denote by ${}^{-}N_i^k$ the number of commodities in $\mathcal{F}_1$ entering O_i^k, and by ${}^{-}D_i^k$ the maximum demand over all commodities in $\mathcal{F}_1$ entering O_i^k.

(P1) For all $i \in [R]$ and $k \in [K]$, ${}^{+}N_i^k \leq N$ and ${}^{+}N_i^k > 0$ with $k > 1$ implies ${}^{+}N_i^{k-1} = N$. Similarly, for all $i \in [R]$ and $k \in [K]$, ${}^{-}N_i^k \leq N$ and ${}^{-}N_i^k > 0$ with $k > 1$ implies ${}^{-}N_i^{k-1} = N$. In words, the number of commodities per copy of a ToR switch is at most N, with commodities incident to a common switch assigned to copies of that switch from the lowest to the highest copy, such that a commodity is assigned to a higher copy if there are N commodities incident to each of the lower copies.

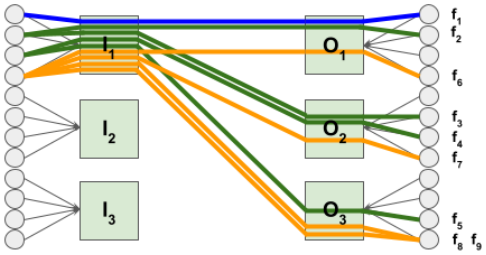
(P2) For all $f_i, f_j \in \mathcal{F}_1$ such that $i(f_i) = i(f_j)$ and $i \leq j$, $k(f_i) \leq k(f_j)$. Similarly, for all $f_i, f_j \in \mathcal{F}_1$ such that $j(f_i) = j(f_j)$ and $i \leq j$, $l(f_i) \leq l(f_j)$. In words, the assignment of commodities incident to a common ToR switch to copies of that switch is in non-increasing order of demands.

(P3) For all $i \in [R]$, ${}^{+}N_i^Q > 0$ implies $\sum_{k\in[K]} {}^{+}D_i^k \leq P$. Similarly, for all $i \in [R]$, ${}^{-}N_i^Q > 0$ implies $\sum_{k\in[K]} {}^{-}D_k^l \leq P$. In words, if there are N commodities incident to each of the lowest $Q-1$ copies of a ToR switch, then the total demand over the highest demand commodities incident to each of the copies of that switch is at most P.

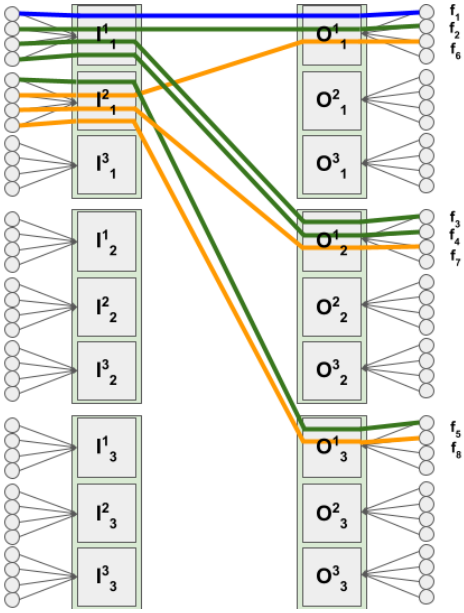
The previous properties are illustrated in Figure 7.

Procedure. Given the set $\mathcal{F}$ of commodities in $C_{N,R}$, the procedure UpholdProperties$(\mathcal{F}, P, Q)$ returns the posited subset $\mathcal{F}_1$. Initially, the set $\mathcal{F}_1$ is empty. The procedure tests each commodity in $\mathcal{F}$ for inclusion in $\mathcal{F}_1$ in decreasing order of demands. For each $i \in [\,|\mathcal{F}|\,]$, the procedure includes f_i in $\mathcal{F}_1$ if there are copies k and l such that the assignment of f_i to $(I_{i(f_i)}^k, O_{j(f_i)}^l)$ satisfies properties P1 through P3.

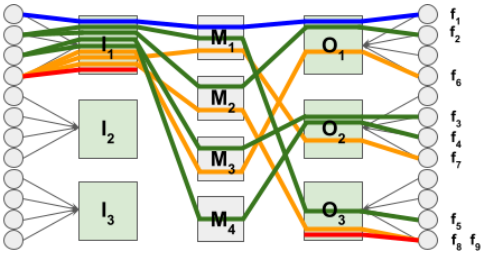
In detail, denote by $\mathcal{G} \subseteq \mathcal{F}_1$ the set of commodities included in $\mathcal{F}_1$ prior to testing f_i. Let x be the lowest copy of $I_{i(f_i)}$ such that the number of commodities in $\mathcal{G}$ leaving that copy is strictly less than N,



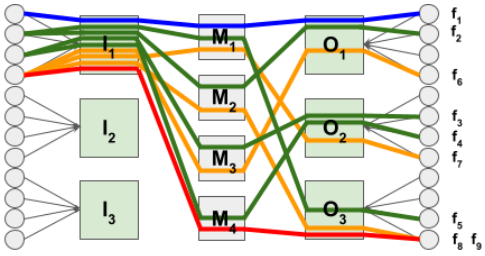
(a) *A set of commodities in the original network.*



(b) *A subset of the commodities mapped to the new network.*



(c) *Routing of the commodities upon Phase 1.*



(d) *Routing of the commodities upon Phase 2.*

Figure 7: *Algorithm 1 with $Q = 3$ and $P = 5/3$ for a set of commodities in the Clos network $C_{4,3}$. Commodity f_1 has demand 1 (in blue), commodities f_2 through f_5 have demand $1/2$ (in green), and commodities f_6 through f_9 have demand $1/4$ (in orange). While the subset of the commodities in the new instance satisfy properties $P1$ through $P3$, the complete set does not: if f_9 (in red) is included in the subset, then it must be assigned to (I_1^3, O_3^1), in which case, since ${}^{+}N_i^3 > 0$, P3 implies $1 + 1/2 + 1/4 < 5/3$. In the end of Phase 1, the minimum congestion path from I_1 to O_3 is M_4; hence, in Phase 2, commodity f_9 is assigned to M_4.*

$x \coloneqq \min\{k \in [K] \,|\, {}^{+}N^k_{i(f_i)} < N\}$. Likewise, let $y \coloneqq \min\{l \in [K] \,|\, {}^{-}N^l_{j(f_i)} < N\}$. We say that $I_{i(f_i)}$ *accepts* f_i if it holds that:

$$x \geq Q \text{ implies } \sum_{k \in [x-1]} {}^{+}D^k_{i(f_i)} + \max\{{}^{+}D^x_{i(f_i)}, \mathrm{dem}(f_i)\} \leq P;$$

and that $I_{i(f_i)}$ *rejects* f_i otherwise. Since commodities are tested in decreasing order of demands, if ${}^{+}N^x_{i(f_i)} > 0$, then $I_{i(f_i)}$ accepts f_i. Likewise, we say that $O_{j(f_i)}$ *accepts* f_i if it holds that:

$$y \geq Q \text{ implies } \sum_{l \in [y-1]} {}^{-}D^l_{j(f_i)} + \max\{{}^{-}D^y_{j(f_i)}, \mathrm{dem}(f_i)\} \leq P;$$

and that $O_{j(f_i)}$ *rejects* f_i otherwise. Commodity f_i is included in $\mathcal{F}_1$ if its accepted by both $I_{i(f_i)}$ and $O_{j(f_i)}$, in which case it is assigned to $(I^x_{i(f_i)}, O^y_{j(f_i)})$.

Phase 1.b. The commodities in $\mathcal{F}_1$ are routed in the original network according to a edge-disjoint routing in the new network (line 2 of Algorithm 1). Since the number of commodities incident to a copy of a ToR switch is at most the number N of middle switches, there is one such routing in the new network, and it can be found in polynomial time (see Theorem 4.1).

The routing of $\mathcal{F}_1$ assigns the commodities incident to a common copy of a ToR switch to different middle switches, with at most K commodities incident to that ToR switch assigned to the same middle switch. Therefore, the maximum congestion over all edges incident to a ToR switch is at most the total demand over the highest demand commodities incident to each copy of that switch. If there are N commodities incident to each of the lowest $Q-1$ copies of a ToR switch, then this congestion is at most P. In the analysis of the next section, Q is set such that the total demand over the highest demand commodities incident to the first $Q-1$ copies of a ToR switch is also at most P, so that the congestion of the routing of $\mathcal{F}_1$ is at most P.

5.1.2 Phase 2

The second phase of the algorithm routes each commodity not routed in Phase 1. The subset of the commodities not routed in Phase 1 is denoted by $\mathcal{F}_2 \coloneqq \mathcal{F} \setminus \mathcal{F}_1$. Consider that the commodities in $\mathcal{F}_2$ are indexed in decreasing order of demands, $\mathcal{F}_2 \coloneqq \{f_{i_1}, f_{i_2}, \dots f_{i_{|\mathcal{F}_2|}}\}$ with $i_j \in [\,|\mathcal{F}|\,]$ and $i_j \leq i_{j+1}$ for all $j \in [\,|\mathcal{F}_2|-1]$. Initially, the congestion of each edge is the total demand over all commodities in $\mathcal{F}_1$ routed through that edge (line 4). The algorithm routes each commodity in $\mathcal{F}_2$ in decreasing order of demands. For each $j \in [\,|\mathcal{F}_2|\,]$, the algorithm routes f_{i_j} on a path whose congestion is minimum, breaking ties arbitrarily (line 8). Then, the algorithm updates the congestion of that path (line 9). The execution of Algorithm 1 is illustrated in Figure 7.

5.2 Analyzing the Algorithm

Define the lower bound L on the optimal congestion OPT as follows:

$$L \coloneqq \max_{i \in [R]} \max\{{}^{+}L_i, {}^{-}L_i\},$$

where

$${}^{+}L_i \coloneqq \max\Big\{ \max_{f \in \mathcal{F}: i(f)=i} \mathrm{dem}(f), {}^{1}\!/_{N} \times \sum_{f \in \mathcal{F}: i(f)=i} \mathrm{dem}(f) \Big\} \text{ and}$$

$${}^{-}L_i \coloneqq \max\Big\{ \max_{f \in \mathcal{F}: j(f)=i} \mathrm{dem}(f), {}^{1}\!/_{N} \times \sum_{f \in \mathcal{F}: j(f)=i} \mathrm{dem}(f) \Big\}.$$

In words, OPT is at least the highest demand over all commodities leaving I_i (entering O_i), and at least the total demand over all commodities leaving I_i (entering O_i) averaged over the number of middle switches; if flow were arbitrarily splittable, then the latter term would correspond to the congestion of each edge leaving I_i (entering O_i) in the minimum congestion routing that divides the demand of each commodity uniformly over all source-destination paths.

The next theorem shows that, if P is set to ${}^{9}\!/_{5} \times L$ and Q to 3, then the algorithm approximates a minimum congestion routing by a factor at most ${}^{9}\!/_{5}$.

Theorem 5.1. *Fix $p \coloneqq {}^{9}\!/_{5}$ and $q \coloneqq 3$. For every set $\mathcal{F}$ of commodities, if Algorithm 1 sets $P = p \times L$ and $Q = q$, then it returns a routing with congestion at most $p \times OPT$.*

Preliminaries to the proof of Theorem 5.1. The proof of the theorem makes use of the additional properties of the set $\mathcal{F}_1$ of commodities routed in Phase 1 detailed in the next lemma. Denote by ${}^{+}d_i^k$ the minimum demand over all commodities in $\mathcal{F}_1$ leaving I_i^k,

$${}^{+}d_i^k := \min_{\substack{f \in \mathcal{F}_1 : i(f) = i \\ k(f) = k}} \text{dem}(f).$$

Likewise, denote by ${}^{-}d_i^k$ the minimum demand over all commodities in $\mathcal{F}_1$ entering O_i^k.

Lemma 5.2. *The set $\mathcal{F}_1$ of commodities satisfies the following additional properties:*

- ***(Q1)*** *Assume that $p \geq \sum_{k \in [q-1]} 1/k$. For all $i \in [R]$, $\sum_{k \in [K]} {}^{+}D_i^k \leq p \times OPT$. Similarly, for all $j \in [R]$, $\sum_{l \in [K]} {}^{-}D_j^l \leq p \times OPT$. In words, the congestion of the routing of $\mathcal{F}_1$ is at most $p \times OPT$.*
- ***(Q2)*** *For all $f \in \mathcal{F}$, dem$(f) > 1/q \times OPT$ implies $f \in \mathcal{F}_1$. In words, every commodity with demand greater than $1/q \times OPT$ is routed in Phase 1.*
- ***(Q3)*** *Assume that there is $f \in \mathcal{F}_2$ such that $I_{i(f)}$ rejected f when it was tested in Phase 1. Then, $\sum_{k \in [K]} {}^{+}d_{i(f)}^k > (p-1) \times L$. Similarly, assume that there is $f \in \mathcal{F}_2$ such that $O_{j(f)}$ rejected f when it was tested in Phase 1. Then, $\sum_{l \in [K]} {}^{-}d_{j(f)}^l > (p-1) \times L$. In words, if there is a commodity not routed in Phase 1, then, assuming that it was rejected by its input switch, every edge incident to that switch has congestion greater than $(p-1) \times L$.*

The proof of the lemma requires two preparatory claims. Since the proofs of the statements in the lemma and in the preparatory claims are analogous for both input and output switches, they are given only for input switches.

Claim 5.3. *For all $i \in [R]$ and all positive integers k, the number of commodities $f \in \mathcal{F}$ such that $i(f) = i$ and dem$(f) > 1/k \times OPT$ is at most $N \times (k-1)$. Similarly, for all $j \in [R]$ and all positive integers l, the number of commodities $f \in \mathcal{F}$ such that $j(f) = j$ and dem$(g) > 1/l \times OPT$ is at most $N \times (l-1)$.*

Proof. Assume, by contradiction, that the number of commodities $f \in \mathcal{F}$ such that $i(f) = i$ and dem$(f) > 1/k \times OPT$ is at least $N \times (k-1) + 1$. Then, for every routing of these commodities, there is a middle switch to which at least k commodities are assigned. Therefore, since the demand of each of these k commodities is strictly greater than $1/k \times OPT$, we conclude that the congestion a minimum congestion routing for $\mathcal{F}$ is strictly greater than OPT, thus arriving at a contradiction. □

Claim 5.4. *For all $f \in \mathcal{F}_1$, dem$(f) \leq 1/k(f) \times OPT$. Similarly, for all $f \in \mathcal{F}_1$, dem$(f) \leq 1/l(f) \times OPT$.*

Proof. Assume, by contradiction, that there is $f \in \mathcal{F}_1$ such that dem$(f) > 1/k(f) \times OPT$. From P1, we know that ${}^{+}N_i^k = N$ for all $k \in [k(f)-1]$, and, from P2, that, for all commodities $g \in \mathcal{F}_1$ such that $i(g) = i(f)$ and $k(g) < k(f)$, dem$(f) > 1/k(f) \times OPT$. Therefore, we conclude that the number of commodities $g \in \mathcal{F}$ such that $i(g) = i(f)$ and dem$(g) > 1/k(f) \times OPT$ is at least $N \times (k(f)-1) + 1$. However, from Claim 5.3, we know that this number is at most $N \times (k(f)-1)$, thus arriving at a contradiction. □

Proof of Lemma 5.2. We show that each of the properties Q1 through Q3 is satisfied.

Property Q1: From P3, we know that ${}^{+}N_i^q = 0$ or $\sum_{k\in[K]} {}^{+}D_i^k \leq p \times L$. If ${}^{+}N_i^q = 0$, then, from Claim 5.4, we write:

$$\begin{aligned}\sum_{k\in[K]} {}^{+}D_i^k &= \sum_{k\in[q-1]} {}^{+}D_i^k \\ &\leq \sum_{k\in[q-1]} 1/k \times OPT && \text{(from Claim 5.4)} \\ &\leq p \times OPT && \text{(from the assumption on } p\text{)}.\end{aligned}$$

If ${}^{+}N_i^q \neq 0$, then we write:

$$\begin{aligned}\sum_{k\in[K]} {}^{+}D_i^k &\leq p \times L \\ &\leq p \times OPT.\end{aligned}$$

In both cases, we conclude that $\sum_{k\in[K]} {}^{+}D_i^k \leq p \times OPT$.

Property Q2: Assume, by contradiction, that $\text{dem}(f) > 1/q \times OPT$ but $f \in \mathcal{F}_2$. Denote by $\mathcal{G} \subseteq \mathcal{F}_1$ the set of commodities included in $\mathcal{F}_1$ prior to testing f. From P1 and P3, ${}^{+}N_{i(f)}^k = N$ for all $k \in [q-1]$, or ${}^{-}N_{j(f)}^l = N$ for all $l \in [q-1]$. Assume, without loss of generality, that ${}^{+}N_{i(f)}^k = N$ for all $k \in [q-1]$. Since commodities are tested in non-increasing order of demands, we conclude that the number of commodities $g \in \mathcal{G}$ such that $i(g) = i(f)$ and $\text{dem}(g) > 1/q \times OPT$ is at least $N \times (q-1) + 1$. However, from Claim 5.3, we know that this number is at most $N \times (q-1)$, thus arriving at a contradiction.

Property Q3: Denote by $\mathcal{G} \subseteq \mathcal{F}_1$ be the set of commodities included in $\mathcal{F}_1$ prior to testing f, and by x the lowest copy of $I_{i(f)}$ such that the number of commodities in $\mathcal{G}$ leaving that copy is strictly less than N. From P2 and P3, we write:

$$\begin{aligned}\sum_{k\in[K]} {}^{+}d_{i(f)}^k &\geq \sum_{k\in[x-2]} {}^{+}d_{i(f)}^k + {}^{+}d_{i(f)}^{x-1} \\ &\geq \sum_{k\in[2,x-1]} {}^{+}D_{i(f)}^k + \max\{{}^{+}D_{i(f)}^x, \text{dem}(f)\} && \text{(from P2)} \\ &= \underbrace{\sum_{k\in[x-1]} {}^{+}D_{i(f)}^k + \max\{{}^{+}D_{i(f)}^x, \text{dem}(f)\}}_{> p\times L, \text{ since } I_{i(f)} \text{ rejected } f} - {}^{+}D_{i(f)}^1 \\ &> (p-1) \times L, && \text{(from P3)}\end{aligned}$$

to conclude that $\sum_{k\in[K]} {}^{+}d_{i(f)}^k > (p-1) \times L$. □

Proof of Theorem 5.1. The proof of the theorem is by contradiction. By choice of p and q, $p \geq \sum_{k\in[q-1]} 1/k$. Consequently, from Q1, we know that the congestion of the routing for $\mathcal{F}_1$ is at most $p \times OPT$. Consider the iteration in Phase 2 when a commodity $f \in \mathcal{F}_2$ is routed. Assume, without loss of generality, that $I_{i(f)}$ rejected f when it was tested in Phase 1. Denote by $\mathcal{H}$ the set of commodities routed in Phase 1 or in previous iterations in Phase 2. We say that a edge $I_i M_m$ ($M_m O_j$) is *congested* if the aggregate demand

over all commodities in $\mathcal{H}$ routed through $I_i M_m$ ($M_m O_j$) is greater than $D \coloneqq p \times OPT - \text{dem}(f)$,

$$\sum_{\substack{h \in H : i(h) = i(f), \\ \text{and } r(h) = r(f)}} \text{dem}(h) > D \quad \Big(\sum_{\substack{h \in H : i(h) = i(f), \\ \text{and } r(h) = r(f)}} \text{dem}(h) > D \Big),$$

and that a path $I_i M_m O_j$ is congested if at least one of $I_i M_m$ and $M_m O_j$ are congested. Assume, by contradiction, that $I_{i(f)} M_m O_{j(f)}$ is congested for all $m \in [N]$.

The derivation of the contradiction makes use of the next two claims. Let C be the number of congested edges leaving $I_{i(f)}$.

Claim (i): D satisfies $D \geq (p - 1/q) \times OPT$. From Q2, we write

$$\begin{aligned} D &\triangleq p \times OPT - \text{dem}(f) \\ &\geq p \times OPT - 1/q \times OPT \\ &= (p - 1/q) \times OPT. \end{aligned}$$ ◁

Claim (ii): C satisfies $C > N \times (1 - \frac{1}{p - 1/q})$. From the assumption that $I_{i(f)} M_m O_{j(f)}$ is congested for all $m \in [N]$, we know that if $I_{i(f)} M_m$ is not congested, then $M_m O_{i(f)}$ is congested, and we deduce that

$$\sum_{h \in \mathcal{H} : j(h) = j(f)} \text{dem}(h) > (N - C) \times D \Leftrightarrow C > N - \frac{1}{D} \sum_{h \in \mathcal{H} : j(h) = j(f)} \text{dem}(h).$$

Then, from Claim (i), we write

$$\begin{aligned} C &> N - \frac{1}{D} \times \sum_{h \in \mathcal{H} : j(h) = j(f)} \text{dem}(h) \\ &\geq N - \frac{1}{(p - 1/q) \times OPT} \times \sum_{h \in \mathcal{H} : j(h) = j(f)} \text{dem}(h) \\ &\geq N \times (1 - \frac{1}{p - 1/q}). \end{aligned}$$ ◁

The contradiction follows from the forthcoming lower bound on the total demand over all commodities in $\mathcal{H}$ leaving $I_{i(f)}$. From Q3 and the previous claims, we write

$$\begin{aligned} \sum_{h \in \mathcal{H} : i(h) = i(f)} \text{dem}(h) &> C \times D + (N - C) \times (p - 1) \times L \\ &= N \times L \times (p - 1) + C \times (D - (p - 1) \times L) \\ &\geq N \times L \times (p - 1) + C \times (D - (p - 1) \times OPT) \\ &> N \times L \times (p - 1) + N \times OPT \times (1 - \frac{1}{p - 1/q})(1 - 1/q) \\ &\geq \sum_{h \in \mathcal{H} : i(h) = i(f)} \text{dem}(h) \times (p - 1 + (1 - \frac{1}{p - 1/q})(1 - 1/q)). \end{aligned}$$

By setting $p \triangleq 9/5$ and $q \triangleq 3$, we write

$$p - 1 + \Big(1 - \frac{1}{p - 1/q}\Big)\Big(1 - 1/q\Big) = 167/165$$

to conclude that

$$\sum_{h \in \mathcal{H} : i(h) = i(f)} \text{dem}(h) > \sum_{h \in \mathcal{H} : i(h) = i(f)} \text{dem}(h).$$ □

The proof of the previous theorem can be generalized to show that, if P and Q are set as before, then not only does the algorithm approximate the minimum congestion by a factor at most $9/5$, but it also returns a routing with congestion at most $9/5$. This generalization only involves introducing the preparatory Claim 5.5 analogous to Claim 5.3, and replacing the term OPT with the term $\min\{OPT, 1\}$ everywhere in the analysis.

Claim 5.5. *For all $i \in [R]$ and all positive integers k, the number of commodities $f \in \mathcal{F}$ such that $i(f) = i$ and $\text{dem}(f) > 1/k$ is at most $N \times (k-1)$. Similarly, for all $j \in [R]$ and all positive integers l, the number of commodities $f \in \mathcal{F}$ such that $j(f) = j$ and $\text{dem}(f) > 1/l$ is at most $N \times (l-1)$.*

Proof. From the assumption that the total demand over all commodities leaving a source is at most 1, we know that, for all $s \in [N]$, the number of commodities $f \in \mathcal{F}$ such that $i(f) = f$, $s(f) = s$, and $\text{dem}(f) > 1/k$ is at most $k-1$. Therefore, the number of commodities $f \in \mathcal{F}$ such that $i(f) = f$ and $\text{dem}(f) > 1/k$ is at most $N \times (k-1)$. □

Theorem 5.6. *Fix $p := 9/5$ and $q := 3$. For every set $\mathcal{F}$ of commodities, if Algorithm 1 sets $P = p \times L$ and $Q = q$, then it returns a routing with congestion at most $p \times \min\{OPT(\mathcal{F}), 1\}$.*

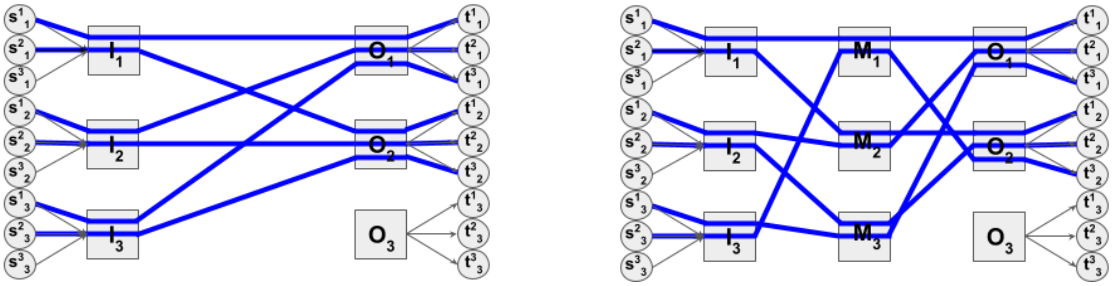
6 Offline: Lower Bounds on Congestion and Approximation

We present lower bounds on worst-case congestion and approximation of minimum congestion routings in Clos networks by polynomial-time algorithms. In §6.1, we show that there are sets of commodities for which the congestion of a minimum congestion routing is at least $3/2$, and in §6.2, we show that it is impossible for any polynomial-time algorithm to approximate a minimum congestion routing by a factor less than $3/2$ unless $P = NP$.

The funnel gadget. The building block of the constructions yielding the posited lower bounds is the following set of commodities. The *funnel gadget* of size N, for some integer $N \geq 2$, illustrated in Figure 8a for $N = 3$, corresponds to the set of commodities in the Clos network $C_{N,N}$ where for each of the N input switches there are $N-1$ commodities with demand 1 leaving that input switch and entering each of the first $N-1$ output switches:

- There is one commodity (s_i^j, t_j^i) with demand 1, for all $i \in [N]$ and $j \in [N-1]$.

The key property of the funnel gadget, detailed in the next lemma, is that there is a unique routing with congestion 1 (modulus the numbering of the middle switches), which assigns the commodities incident to a common ToR switch to different middle switches, and does not assign a commodity to a different middle switch at each input switch.



***(a)** The funnel gadget without routing.* ***(b)** Elemental routing of the funnel gadget.*

***Figure 8:** The funnel gadget of size $N = 3$.*

Lemma 6.1. *Consider the funnel gadget of size N. For every routing with congestion 1, the following properties hold:*

(1) *For all $i \in [N]$, the $N-1$ commodities leaving I_i are assigned to $N-1$ different middle switches, and, for all $j \in [N-1]$, the N commodities entering O_j are assigned to N different middle switches.*

(2) *For all $i_1, i_2 \in [N]$ such that $i_1 \neq i_2$, the middle switch to which no commodity leaving I_{i_1} is assigned differs from the middle switch to which no commodity leaving I_{i_2} is assigned.*

Proof. The first property follows from the fact that all commodities have demand 1. We show that the second property holds. First, since the N commodities entering each of the first $N-1$ output switches are assigned to N different middle switches, we deduce that each middle switch is assigned $N-1$ commodities. Second, since the $N-1$ commodities leaving each of the N input switches are assigned to $N-1$ different middle switches, we further deduce that for each middle switch the $N-1$ commodities assigned to it leave $N-1$ different input switches. Therefore, if there are two input switches I_{i_1} and I_{i_2}, $i_1 \neq i_2$, and a middle switch M_m such that no commodity leaving I_{i_1} is assigned to M_m and no commodity leaving I_{i_2} is assigned to M_m, then there are at most $N-2$ commodities assigned to M_m, which contradicts the existence of $N-1$ commodities assigned to M_m. □

Therefore, all routings of the funnel gadget with congestion 1 reduce to the routing described below and illustrated in Figure 8b for $N = 3$.

- The commodity (s_i^j, t_j^i) is assigned to M_{m+1}, where $m = i + j - 2 \pmod N$, for all $i \in [N]$ and $j \in [N-1]$.

We call this routing the *elemental* routing for the funnel gadget.

6.1 Limits to Congestion

If all commodities have demand 1, in which case there is at most one commodity per source and per destination, then for every set of commodities there is a routing with congestion 1. (This is a corollary of Theorem 4.1.) In contrast, the next theorem shows that if this premise is relaxed such that each commodity has demand 1 or $1/2$, in which case there are at most two commodities per source and per destination, then there are now sets of commodities for which every routing has congestion greater than 1.

Theorem 6.2. *Consider a Clos network $C_{N,R}$, for $N \geq 2$ and $R \geq N+1$. There is a set of commodities with demands 1 or $1/2$ for which the congestion of a minimum congestion routing is $3/2$.*

Proof of Theorem 6.2. We design a set of commodities in the Clos network $C_{N,N+1}$ composed of commodities with demand 1 or $1/2$ for which every routing has congestion at least $3/2$. Since increasing the number of input and output switches from $N+1$ to R does not change the number of source-destination paths from the first $N+1$ input switches to the first $N+1$ output switches, if a set of commodities satisfies the theorem requirements in the network $C_{N,N+1}$, then the same set also satisfies them in the network $C_{N,R}$. Consequently, the conclusion follows.

Designing the set of commodities: The posited set of commodities, illustrated in Figure 9a for $N = 3$, is composed of three types of commodities:

- The type 1 commodities (in blue) correspond to the funnel gadget of size N.
- *Type 2 (in orange):* There is one commodity $(s_i^N, t_N^{\lceil i/2 \rceil})$ with demand $1/2$, for all $i \in [N]$.

- *Type 3 (in green):* There is one commodity (s^N_{N+1}, t^N_N) with demand 1.

Calculating a minimum congestion routing: We show that the congestion of a minimum congestion routing is $3/2$. The proof involves two steps. The first step shows that there exists a routing with congestion $3/2$. The posited minimum congestion routing is described below and illustrated in Figure 9b for $N = 3$.

- The type 1 commodities are assigned according to the elemental routing for the funnel gadget.
- The type 2 commodity $(s^N_i, t^{\lceil i/2 \rceil}_N)$ is assigned to M_{m+1}, where $m = i - 2 \pmod N$, for all $i \in [N]$.
- The type 3 commodity is assigned to M_N.

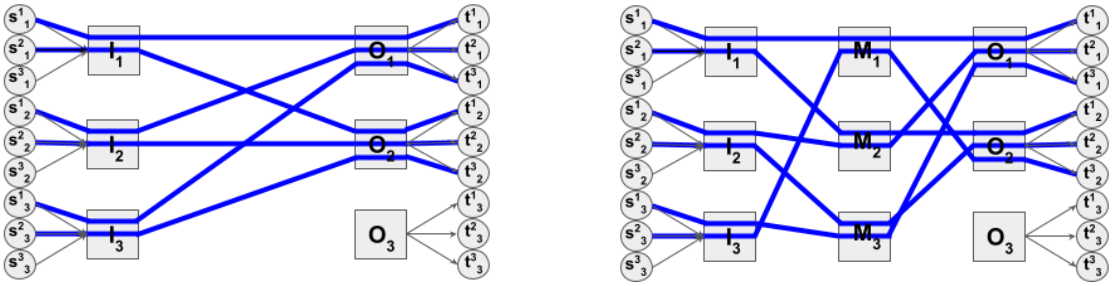
Every edge except for $M_N O_N$ is traversed by at most one commodity, and thus has congestion at most 1. Edge $M_N O_N$ is traversed by both the type 2 commodity leaving I_1 and the type 3 commodity, and thus has congestion $3/2$. Therefore, the routing has congestion $3/2$.

The second step shows that there is no routing with congestion less than $3/2$. First, from Lemma 6.1, we know that every routing of the type 1 commodities with congestion less than $3/2$ reduces to the elemental routing for the funnel gadget, and thus assume that the type 1 commodities are routed according to it. Then, we distinguish two cases, which assert that every subsequent routing of the type 2 and the type 3 commodities lead to congestion at least $3/2$, thereby concluding the proof.

- *Case 1:* If for all $i \in [N]$ the type 2 commodity leaving I_i is assigned to the middle switch to which no type 1 commodity leaving I_i is assigned to, then the N type 2 commodities are assigned to N different middle switches (as in Figure 9b). Consequently, the type 3 commodity is assigned to the same middle switch than some type 2 commodity, in which case the routing has congestion $3/2$.
- *Case 2:* Otherwise, there is $i \in [N]$ such that the type 2 commodity leaving I_i is assigned to the same middle switch than some type 1 commodity leaving I_i, in which case the routing has congestion at least $3/2$. □

6.2 Limits to Approximation

If all commodities have demand 1, then for every set of commodities a routing with congestion 1 can be found in polynomial-time. (This is again a corollary of Theorem 4.1.) In contrast, the theorem below shows that if this premise is again relaxed such that each commodity has demand 1 or $1/2$, then it becomes impossible to distinguish in polynomial-time between a routing with congestion at most 1 and a routing with congestion at least $3/2$. Consequently, the theorem implies that it is impossible to approximate a minimum congestion routing by a factor less than $3/2$.



(a) *The commodities without routing.* ***(b)*** *A minimum congestion routing for the commodities.*

Figure 9: *The set of commodities underlying the proof of Theorem 6.2 for $N = 3$ and $R = 4$.*

Theorem 6.3. *For special case of the minimum congestion routing problem in Clos networks where commodities have demand* 1 *or* $1/2$*, the question of deciding if there is a routing with congestion at most* 1 *is* NP*-complete.*

Proof of Theorem 6.3. We introduce a reduction from the 3-edge coloring problem, which is known to be NP-complete [56], to the question of deciding if there is a routing with congestion at most 1 for commodities whose demands are 1 or $1/2$. In the 3-edge coloring problem, the input is an undirected graph whose maximum vertex degree is 3. The question is to decide if there exists a *proper edge-coloring*, meaning an assignment from edges to colors such that no two adjacent edges are assigned the same color, using at most 3 colors.

Description of the reduction: Consider an input graph $G = (V(G), E(G))$ to the 3-edge coloring problem, with vertices indexed from 1 to $|V(G)|$, $V(G) = \{v_1, v_2, \ldots, v_{|V(G)|}\}$, and edges indexed from 1 to $|E(G)|$, $E(G) = \{e_1, e_2, \ldots, e_{|E(G)|}\}$. For each vertex $v_k \in V(G)$, let rank_{v_k} be an arbitrary ranking of the neighbors of v_k, such that $\text{rank}_{v_k}(v_l) \in \{1, 2, 3\}$ denotes the ranking of a neighbor v_l of v_k. We construct an instance of the minimum congestion routing problem from G. The construction is illustrated in Figure 10 for a particular input graph.

The Clos network is denoted by $C_{3,3|V(G)|+|E(G)|}$. The ToR switches are divided into $|V(G)|$ vertex blocks and $|E(G)|$ edge blocks, with the k'th vertex block corresponding to input switches $I_{3(k-1)+1}$ through $I_{3(k-1)+3}$ and output switches $O_{3(k-1)+1}$ through $O_{3(k-1)+3}$, $k \in [\,|V(G)|\,]$, and the m'th edge block corresponding to input switch $I_{3|V|+m}$ and output switch $O_{3|V|+m}$, $m \in [\,|E(G)|\,]$.

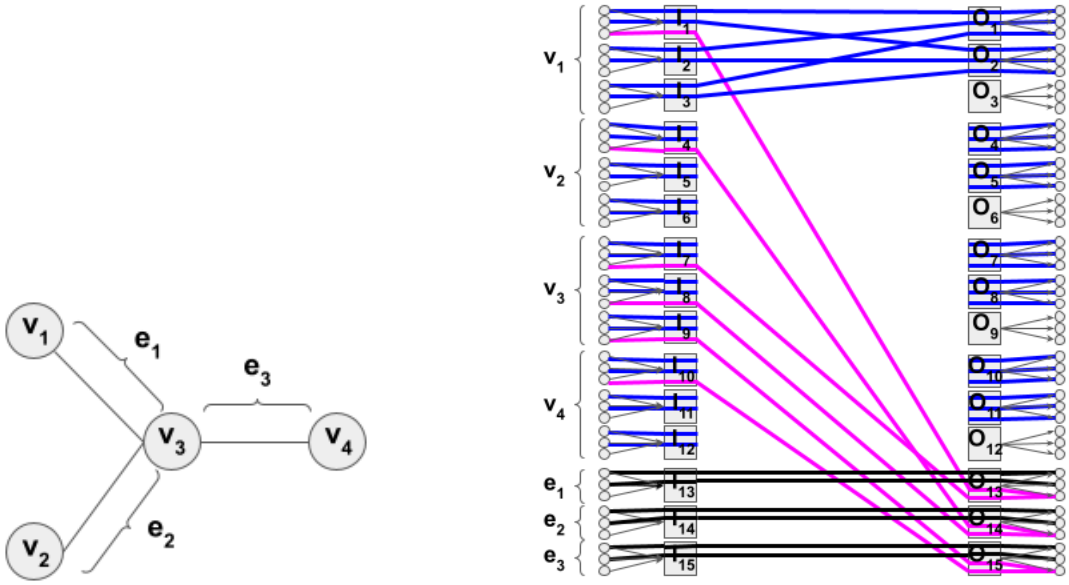
The set of commodities is denoted by $\mathcal{F}(G)$, and is composed of three types of commodities:

- For each vertex $v_k \in V(G)$, there is a set of *vertex commodities* (in blue) in the k'th vertex block corresponding to the translation of the funnel gadget of size 3 to that block.
- For each edge $e_m = \{v_k, v_l\} \in E(G)$, there are two *edge commodities* in the m'th edge block (in black): one commodity $(s^1_{3|V(G)|+m}, t^1_{3|V(G)|+m})$ with demand 1, and one commodity $(s^2_{3|V(G)|+m}, t^2_{3|V(G)|+m})$ with demand 1.
- For each edge $e_m = \{v_k, v_l\} \in E(G)$, there is one *incident commodity* from the k'th vertex block to the m'th edge block, and one *incident commodity* from the l'th vertex block to the m'th edge block (in pink): one commodity $(s^3_{3(k-1)+\text{rank}_{v_k}(v_l)}, t^3_{3|V(G)|+m})$ with demand $1/2$ and one commodity $(s^3_{3(l-1)+\text{rank}_{v_l}(v_k)}, t^3_{3|V(G)|+m})$ with demand $1/2$.

The objective of the construction is to establish the following correspondence between a routing of $\mathcal{F}(G)$ and an edge-coloring of G. Given a routing of $\mathcal{F}(G)$ with congestion 1, the routing of the incident commodities in $\mathcal{F}(G)$ yields a proper edge-coloring of G using at most 3 colors; conversely, given a proper edge-coloring of G using at most 3 colors, the coloring yields a routing of the incident commodities in $\mathcal{F}(G)$ that can be extended to a routing of $\mathcal{F}(G)$ with congestion 1. The next lemma paves the way for this correspondence by identifying the properties met by the routing of the incident commodities in a routing of $\mathcal{F}(G)$ with congestion 1.

Lemma 6.4. *For every routing of* $\mathcal{F}(G)$ *with congestion* 1*, the routing of the incident commodities satisfies the following properties:*

- ***(P1)*** *For every edge* $e_m \in E(G)$*, the two incident commodities entering the* m*'th edge block are assigned to the same middle switch.*
- ***(P2)*** *For every vertex* $v_k \in V(G)$*, each of the incident commodities leaving the* k*'th vertex block is assigned to a different middle switch.*



(a) *Input graph G to the 3-edge coloring problem.*

(b) *Input commodities $\mathcal{F}(G)$ to the minimum congestion routing problem.*

Figure 10: *The reduction underlying the proof of Theorem 6.3 for a particular input graph. The vertex commodities in the 2'nd, 3'rd, and 4'th vertex blocks are not shown. The coloring of the different types of commodities in $\mathcal{F}(G)$ is unrelated to the edge coloring in G.*

If there is a routing of the incident commodities in $\mathcal{F}(G)$ satisfying the previous properties, then there is a routing of $\mathcal{F}(G)$ with congestion 1.

Proof. To show the first part, suppose that there is a routing of $\mathcal{F}(G)$ with congestion 1. We prove that the first property. Since the routing assigns the two edge commodities entering the m'th edge block to two different middle switches, we conclude that the two incident commodities are assigned to the remaining middle switch. We now prove that the second property. From Lemma 6.1, we known that the routing does not assign a vertex commodity to a different middle switch at each input switch in the k'th vertex block. Therefore, we conclude that the incident commodities leaving the k'th vertex block are assigned to different middle switches.

To show the second part, suppose that there is a routing of the incident commodities in $\mathcal{F}(G)$ satisfying both properties. We prove that it can be extended to a routing of $\mathcal{F}(G)$ with congestion 1. From the first property, the two edge commodities in the m'th edge block can be routed with congestion 1 if assigned to the two remaining middle switches. From the second property, the vertex commodities leaving the k'th vertex block can be routed with congestion 1 if assigned according to the elemental routing of the funnel gadget (upon the appropriate numbering of the middle switches). The conclusion follows. □

Correctness of the reduction: We show that there is a routing of $\mathcal{F}(G)$ with congestion 1 if and only if there is a proper edge-coloring of G using at most 3 colors.

($\Rightarrow$) Suppose that there is a routing of $\mathcal{F}(G)$ with congestion 1. Consider the edge-coloring of G using at most 3 colors obtained from the routing where, for each edge $e_m \in E(G)$, edge e_m is assigned color c if the two incident commodities entering the m'th edge block are assigned to middle switch M_c, for some $c \in \{1, 2, 3\}$. From the first part of Lemma 6.4, since the routing of the incident commodities meets the first property, we deduce that the posited coloring is well-defined, and, since the routing of the incident

commodities meets the second property, we further deduce that it is proper, to conclude that there is a proper edge-coloring of G using at most 3 colors.

($\Leftarrow$) Suppose that there is a proper edge-coloring of G using at most 3 colors. Consider the routing of the incident commodities in $\mathcal{F}(G)$ obtained from the coloring where, for each edge $e_m \in E(G)$, the two incident commodities entering the m'th edge block are assigned to middle switch M_c if e_m is assigned color c, for some $c \in \{1, 2, 3\}$. By construction, we deduce that the posited routing of the incident commodities meets the first property, and, since the coloring is proper, we further deduce that it meets the second property, to conclude from the second part of Lemma 6.4 that there is a routing of $\mathcal{F}(G)$ with congestion 1. □

7 Online: Lower Bounds on Congestion and Approximation

We present lower bounds on the worst-case congestion and approximation of minimum congestion routings in Clos networks by online algorithms. An online algorithm is presented with a sequence of commodities. A *deterministic online algorithm* defines a routing for every sequence of commodities that satisfies the following property: for all prefixes P of a sequence F of commodities, the routing for P when the algorithm is given P equals the routing for P when the algorithm is given F. A *randomized online algorithm* defines a probability distribution over the set of all deterministic algorithms; a randomized algorithm with a single-point distribution reduces to a deterministic algorithm.

In §7.1, we show that, for any deterministic online algorithm, there is a sequence of commodities for which, while the congestion of a minimum congestion routing is 1, the congestion of the routing returned by the algorithm is at least 2. In §7.2, we generalize the previous result from any deterministic online to any randomized online algorithm. These results implies that it is impossible for any online algorithm, randomized or deterministic, to approximate a minimum congestion routing by a factor less than 2.

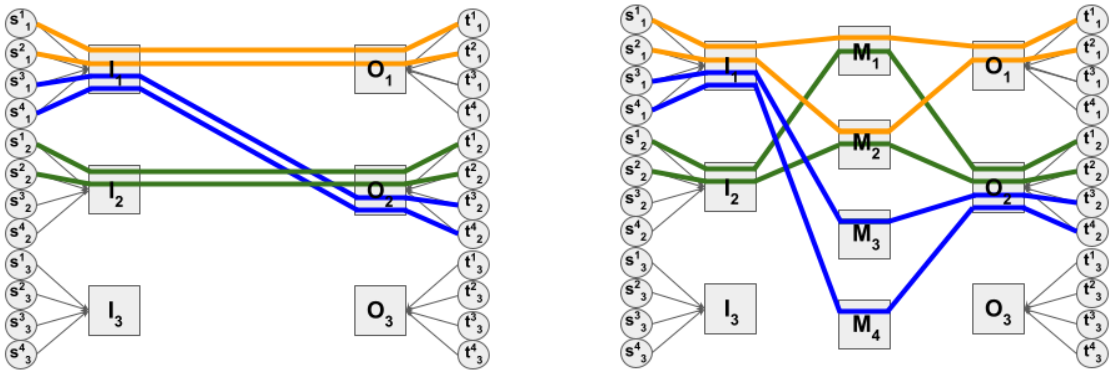
7.1 Limits to Deterministic Online Algorithms

In this section, we assume that all commodities have demand 1, meaning that there is at most one commodity per source and per destination. Under this assumption, the congestion of a minimum congestion routing is 1 for every set of commodities. In contrast, the theorem below shows no deterministic online algorithm can avoid returning a routing with congestion at least 2 for some sequence of commodities.

Theorem 7.1. *Consider a Clos network $C_{N,R}$, for $N \geq 2$ and $R \geq 3$. For every deterministic online algorithm, there is a sequence of commodities with demand 1 for which the congestion of the routing returned by the algorithm is at least 2.*

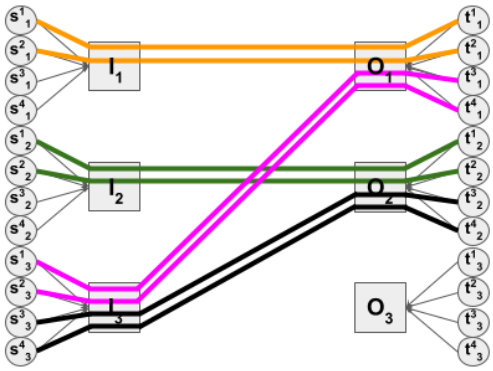
Proof of Theorem 7.1. We design two sequences of commodities in the Clos network $C_{N,3}$ composed of commodities with demand 1 for which every deterministic online algorithm returns a routing with congestion at least 2 for at least one of the sequences. As in the proof of Theorem 6.2, if a sequence of commodities satisfies the theorem requirements in the network $C_{N,3}$, then the same sequence satisfies them in the network $C_{N,R}$.

Designing the sequences of commodities. The posited sequences are denoted by $X = (X_1, X_2)$ and $Y = (Y_1, Y_2)$, and are illustrated in Figure 11 for $N = 4$ and $R = 3$. Each of the sequences consists of one subsequence followed by another, with the sequences agreeing on the prefix, $X_1 = Y_1$, and disagreeing on suffix, $X_2 \neq Y_2$. The arrival order among the commodities within a subsequence is arbitrary, and each commodity is identified by its input-output switch pair (with each commodity incident to a common ToR switch incident to a different server of that switch).

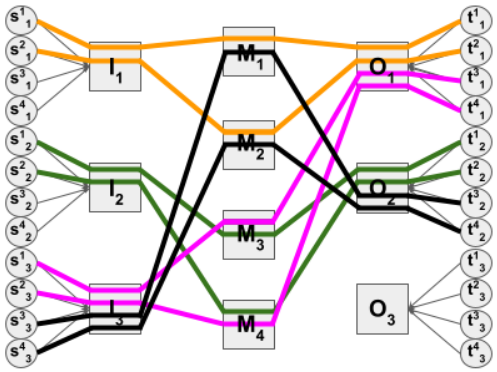


(a) *Sequence X.*

(b) *Edge-disjoint routing of X.*



(c) *A minimum congestion routing for X.*



(d) *A minimum congestion routing for Y.*

Figure 11: *The sequences of commodities underlying Theorem 7.1 for $N = 4$.*

- Subsequence X_1 consists of $N/2$ commodities (I_1, O_1) (in orange), and $N/2$ commodities (I_2, O_2) (in green).
- Subsequence X_2 consists of $N/2$ commodities (I_1, O_2) (in blue).
- Subsequence Y_2 consists of $N/2$ commodities (I_3, O_1) (in pink) and $N/2$ commodities (I_3, O_2) (in black).

The key property of sequences X and Y, detailed in the next lemma, is that, although $X_1 = Y_1$, for any edge-disjoint routing of X and any edge-disjoint routing of Y, the routing of X_1 in the former differs from the routing of X_1 in the latter. Consequently, if the prefix of a given sequence is X_1 and the suffix is either X_2 or Y_2, then any deterministic online algorithm must choose between one of the previous routings of X_1 without knowing in advance whether X_1 will be followed by X_2 or by Y_2, such that the algorithm fails to returns a edge-disjoint routing for at least one of X and Y, as formalized in the final proof step.

Lemma 7.2. *Let $\mathcal{M}$ be the set of all middle switches in the Clos network, $\mathcal{M} \coloneqq \{M_i \mid i \in [N]\}$. For every edge-disjoint routing of sequence X, it holds that:*

(P1) *Each of the $N/2$ commodities (I_1, O_1) is assigned to a different middle switch from a set $\mathcal{N}$ of $N/2$ middle switches; likewise, each of the $N/2$ commodities (I_2, O_2) is assigned to a different middle switch from the same set $\mathcal{N}$.*

For every edge-disjoint routing of sequence Y, it holds that:

(P2) *Each of the $N/2$ commodities (I_1, O_1) is assigned to a different middle switch from a set $\mathcal{N}$ of $N/2$ middle switches; oppositely, each of the $N/2$ commodities (I_2, O_2) is assigned to a different middle switch from the set $\mathcal{M} \setminus \mathcal{N}$.*

Therefore, no routing of X_1 can satisfy both properties.

Proof. To show the first statement, consider an arbitrary edge-disjoint routing of X. Since the N commodities leaving I_1 are spread over all N middle switches, a middle switch is assigned a commodity (I_1, O_1) precisely when it is not assigned a commodity (I_1, O_2); similarly, since the N commodities entering O_2 are spread over all N middle switches, a middle switch is assigned a commodity (I_2, O_2) precisely when it is not assigned a commodity (I_1, O_2). Therefore, each of the $N/2$ middle switches to which a commodity (I_1, O_1) is assigned is also assigned a commodity (I_2, O_2), thus proving the first statement.

To show the second statement, consider now an arbitrary edge-disjoint routing of Y. Following the same argument as above, a middle switch is assigned a commodity (I_1, O_1) precisely when it is not assigned a commodity (I_3, O_1), and a middle switch is assigned a commodity (I_3, O_2) precisely when it is not assigned a commodity (I_3, O_1); consequently, a middle switch is assigned a commodity (I_1, O_1) precisely when it is assigned a commodity (I_3, O_2). Since a middle switch is assigned a commodity (I_3, O_2) precisely when when it is not assigned a commodity (I_2, O_2), each of the $N/2$ middle switches to which a commodity (I_1, O_1) is not assigned assigned a commodity (I_2, O_2), thus proving the second statement. □

Calculating the congestion of a deterministic algorithm. Consider an arbitrary deterministic online algorithm. We show that the algorithm fails to return a edge-disjoint routing for at least one of X and Y, to conclude that it returns a routing with congestion at least 2 for at least one of them. Let r be the routing returned by the algorithm for prefix X_1. We distinguish two cases, depending on whether r satisfies property P1 in Lemma 7.2.

Case 1: Suppose that r satisfies property P1, in which case r does not satisfy property P2. Then, we deduce that the algorithm returns a routing for Y that does not property P2, which means that the routing for Y is not edge-disjoint.

Case 2: Suppose that r does not satisfy property P1. Then, we deduce that the algorithm returns a routing for X that does not satisfy property P1, which means that the routing for X is not edge-disjoint. □

7.2 Limits to Randomized Online Algorithms

The previous theorem does not preclude the possibility that a randomized online algorithm can avoid with non-negligible probability returning a routing with congestion greater than 2 for every sequence of commodities. The next theorem refutes this possibility.

Theorem 7.3. *Consider a Clos network $C_{N,R}$, for $N \geq 2$ and $R \geq 3$. For every randomized online algorithm, there is a sequence of commodities with demand 1 for which the expected congestion of the routing returned by the algorithm is at least $2 - 1/2^S$, where $S = \lfloor R/3 \rfloor$.*

Proof of Theorem 7.3. The proof of the theorem makes use of Yao's Minimax Principle, which is recalled below in the context of the minimum congestion routing problem in Clos networks. Let $\mathscr{S}$ be the set of all sequences of commodities, and $\mathscr{A}$ be the set of all deterministic algorithms. Given a probability distribution p over $\mathscr{S}$ and a probability distribution q over $\mathscr{A}$, denote by $\mathcal{S}_p$ the random sequence drawn from p, and by A_q the randomized algorithm drawn from q. We write $E[A_q(\mathcal{S})]$ for the expected congestion of A_q for a sequence $\mathcal{S}$, and $E[A(\mathcal{S}_p)]$ for the expected congestion of a deterministic algorithm A for $\mathcal{S}_p$.

Lemma 7.4 (Yao's Minimax Principle [48, 49])**.** *For every randomized algorithm A_q and every random sequence $\mathcal{S}_p$, there is a sequence $\mathcal{S}$ of commodities for which the expected congestion of A_q for $\mathcal{S}$ is at least the expected congestion of the optimal deterministic algorithm for $\mathcal{S}_p$:*

$$\max_{\mathcal{S}\in\mathscr{S}} E[A_q(\mathcal{S})] \geq \min_{A\in\mathcal{A}} E[A(\mathcal{S}_p)].$$

Therefore, we design a random sequence based on the sequences X and Y introduced earlier for which the expected congestion of an optimal deterministic algorithm for that random sequence is at least $2-1/2^S$.

Designing the random sequence of commodities. The posited random sequence is denoted by $\mathcal{S}_p$ and is drawn from the uniform distribution p over the following set of 2^S sequences. The set of 2^S sequences is denoted by $\{\, \mathcal{S}_i \mid i \in [2^S] \,\}$, with $\mathcal{S}_i := (\, \mathcal{S}_i^j \mid j \in [S] \,)$. Consider that the ToR switches are divided into S blocks, with the j'th block designating input switches $I_{3(j-1)+1}$ through $I_{3(j-1)+3}$ and output switches $O_{3(j-1)+1}$ through $O_{3(j-1)+3}$, $j \in [S]$. Denote by X_j and Y_j, respectively, the translation of X and Y to the j'th block. Each of the 2^S sequences S_i is composed of S subsequences, with each of the subsequences S_i^j equal to X_j or Y_j. The arrival order among the subsequences within a sequence is arbitrary. Let $< b_j(i) \mid j \in [S] >$ be the binary representation of an integer $i \in [0, 2^S-1]$.

- If $b_j(i-1) = 0$, then $\mathcal{S}_i^j = X_j$; otherwise, $\mathcal{S}_i^j = Y_j$, for all $i \in [2^S]$ and $j \in [S]$.

The key property of the set $\{\, \mathcal{S}_i \mid i \in [2^S] \,\}$, detailed in the next lemma, is that no online deterministic algorithm can return a edge-disjoint routing for more than one sequence in the set. Consequently, every deterministic online algorithm returns a routing with congestion at least 2 for at least 2^S-1 of the 2^S sequences, as formalized in the final proof step.

Lemma 7.5. *Every deterministic online algorithm returns a edge-disjoint routing for at most one sequence in the set $\{\mathcal{S}_i \mid i \in [2^S]\}$.*

Proof. We show that for every pair of sequences in the set $\{\mathcal{S}_i \mid i \in [2^S]\}$ no deterministic online algorithm can return a edge-disjoint routing for both sequences. Consider an arbitrary deterministic online algorithm and two sequences $\mathcal{S}_{i_1}, \mathcal{S}_{i_2} \in \{\mathcal{S}_i \mid i \in [2^S]\}$ such that $\mathcal{S}_{i_1} \neq \mathcal{S}_{i_2}$, meaning that $\mathcal{S}_{i_1}^j \neq \mathcal{S}_{i_2}^j$ for some $j \in [S]$. From Lemma 7.2, the algorithm cannot return a edge-disjoint routing for both $\mathcal{S}_{i_1}^j$ and $\mathcal{S}_{i_2}^j$, implying that it cannot return a edge-disjoint routing for both $\mathcal{S}_{i_1}$ and $\mathcal{S}_{i_2}$. □

Calculating the expected congestion of a deterministic algorithm. Consider an arbitrary deterministic online algorithm A. From Lemma 7.5, we write

$$\begin{aligned} E[A(\mathcal{S}_p)] &\geq \frac{1}{2^S} \times 1 + (1 - \frac{1}{2^S}) \times 2 \\ &= 2 - \frac{1}{2^S}, \end{aligned}$$

to deduce that the expected congestion of an optimal deterministic algorithm for $\mathcal{S}_p$ is at least $2-1/2^S$. Consequently, from Yao's Minimax Principle, we conclude that for every randomized algorithm there is a sequence of commodities for which the expected congestion is at least $2-1/2^S$. □

Corollary 7.6. *Consider a Clos network $C_{N,R}$, for $N \geq 2$ and $R \geq 3$. For every $c < 2 - 1/2^S$, where $S = \lceil R/3 \rceil$, there is no c-approximation randomized online algorithm for the minimum congestion routing problem in Clos networks.*

It not hard to show that, if every commodity has demand 1, then the *Unsorted-Greedy algorithm* returns a routing with congestion at most 2, and thus approximates a minimum congestion routing by a factor of 2. Hence, in this special case, the lower bound of Corollary 7.6 is tight.

8 Conclusion

At the heart of the operation of data-centers is solving a minimum congestion routing problem of unsplittable flows. We investigated this problem in Clos networks, the family of graphs underlying most data-centers, and established that the worst-case congestion of a minimum congestion routing and its approximation-factor by a polynomial-time algorithm are between $3/2$ and $9/5$, thus showing that Clos networks fail to closely emulate a macro-switch. The key idea for the lower bounds was the introduction of a set of flows called a funnel gadget, whereas that for the upper bounds was the design of a new routing algorithm that interpolates between two known heuristics. We leave open the question of closing this gap. While the analysis of our new algorithm can be improved, arriving at the $3/2$ factor, which we believe is the tight factor, should require new techniques. We further proved that the worst-case congestion of a minimum congestion routing and its approximation-factor by an online algorithm are between 2 and 3, deferring the problem of narrowing this gap to future work.

The main takeaway from these results is that, if Clos networks wish to emulate a macro-switch, then routing must be coupled with additional degrees of freedom. We point two such degrees available at the technological forefront of data-centers, and put forward the question of investigating if they allow worst-case congestion close to 1. First, most applications are hosted in virtual machines, and cloud providers can choose in which servers to place each virtual machine [37–39, 57]. Accordingly, an idea is to jointly assign these virtual machines to servers and route the flows between them. Second, while arbitrary flow splitting is unrealistic, several proposals already support a limited degree of splittability [32, 58]. Therefore, another idea is to jointly route flows and divide their demands over a small number of paths.

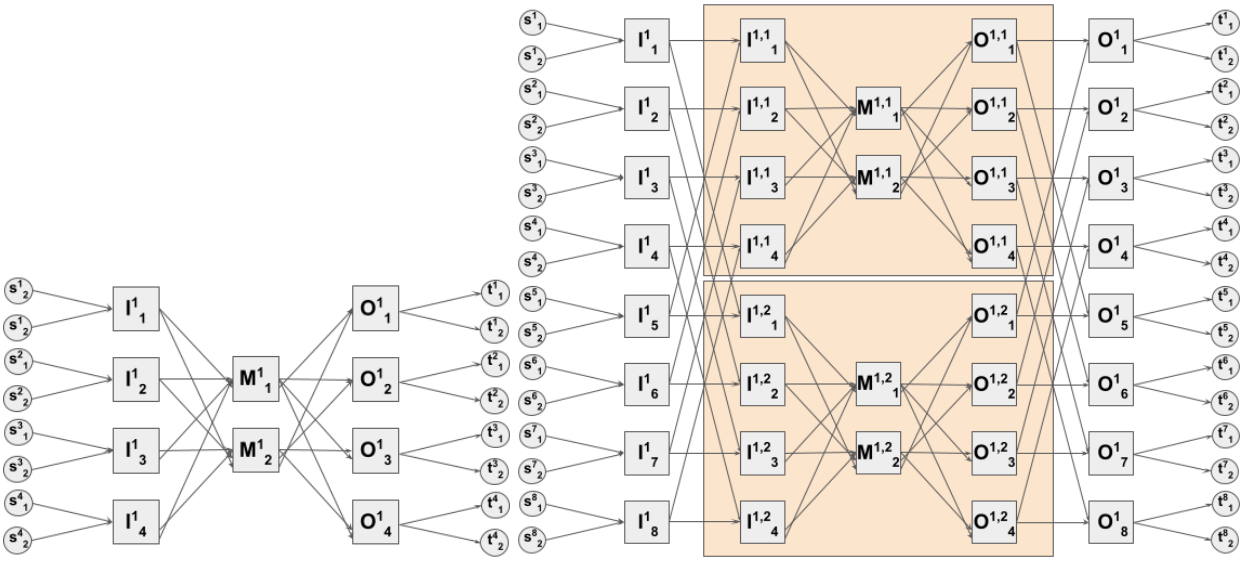
While our investigation was premised on Clos networks with five layers, hyper-scale data-center often built their data-center networks after Clos networks with seven or more layers. It is easy to see that the lower bounds of $3/2$ on the minimum congestion and its approximation-factor by a polynomial-time algorithm carry over to Clos with an arbitrary number of stages, whereas the upper bounds yield by our new algorithm do not. More interestingly, it is not obvious how to raise these lower bounds away from $3/2$, which motivates the following open question: is the minimum congestion in Clos network with arbitrary number of layers and its approximation factor by a polynomial time algorithm is at most a constant (independent of the number of layers)?

A The Generalized Melen-Turner Algorithm

We show that the generalization of the Melen-Turner algorithm to Clos networks with an arbitrary number of layers yields a routing with congestion at most logarithmic in the number of layers (up to constant factors) while approximating the minimum congestion by that same factor. In Section A.1, we review the construction of a Clos network with an arbitrary number of layers. In Section A.2, we present the natural generalization of the Melen-Turner algorithm to a Clos network with an arbitrary number of layers, and, in Section A.3, we analyze the algorithm to show the posited result.

A.1 Clos Networks with an Arbitrary Number of Layers

The construction of a Clos network can be generalized to an arbitrary number of layers. A Clos network with $2K + 3$ layers, for some positive integer K, is called a K-Clos network, and is defined below. The main body of this paper investigates the minimum congestion problem in 1-Clos networks. The key idea is that, whereas in a 1-Clos network the R input switches are connected to the R output switches via N middle switches, in a K-Clos network the R input switches are connected to the R output switches via N parallel $(K-1)$-Clos networks with R/N input and output switches. In other words, a K-Clos network corresponds to a 1-Clos network where each middle switch is replaced by a $(K-1)$-Clos network.



(a) The Clos network $C^1_{4,2}$, whose input and output switches are linked by 2 middle switches. **(b)** The Clos network $C^2_{8,2}$, whose input and output switches are linked by 2 parallel Clos networks $C^1_{4,2}$

Figure 12: A 5-layer and a 7-layer Clos network. The colored boxes in the 7-stage Clos network represent the middle switches that the underlying 5-stage Clos networks seek to emulate.

Definition A.1 ([35,36]). *A K-Clos network $C^K_{N,R}$ is parameterized by positive integers K, N, and R, where R is a multiple of N^K, and designates the directed graph with the following vertex and edge sets (see Figure 12).*

- ***Vertex set.*** *The vertices V are partitioned into $2K+3$ layers V_1 through V_{2K+3}. In V_1 (respectively V_{2K+3}) there are $N \times R$ vertices, and each vertex is a* source (destination) *and is denoted by $s^{j_1}_{i_1}$ ($t^{j_1}_{i_1}$), where $j_1 \in [N]$ and $i_1 \in [R]$. For all $k \in [2, K+1]$, in V_k (respectively V_{2K+4-k}) there are R vertices, and each vertex is called an* input (output) switch *and is denoted by $I^{j_2,\ldots,j_k}_{i_k}$ ($O^{j_2,\ldots,j_k}_{i_k}$), where $j_l \in [N^{l-2}]$, $l \in [2,k]$, and $i_k \in [R/N^{k-2}]$. In V_{K+2}, there are R/N^{K-1} vertices, and each vertex is called a* middle switch *and is denoted by $M^{j_2,\ldots,j_{K+1}}_{i_{K+2}}$, where $j_l \in [N^{l-2}]$, $l \in [2, K+1]$, and $i_{K+2} \in [N]$.*

- ***Edge set.*** *The edges E are from V_i to V_{i+1} for all $i \in [2K+2]$. For all sources $s^{j_1}_{i_1} \in V_1$, there is an edge $s^{j_1}_{i_1} I^1_{i_1}$, and, for all destinations $t^j_i \in V_{2K+3}$, there is an edge $O^1_{i_1} t^{j_1}_{i_1}$. For all $k \in [2, K]$, for all input switches $I^{j_2,\ldots,j_k}_{i_k} \in V_k$ and $I^{j_2,\ldots,j_{k+1}}_{i_{k+1}} \in V_{k+1}$, there is an edge $I^{j_2,\ldots,j_k}_{i_k} I^{j_2,\ldots,j_{k+1}}_{i_{k+1}}$ if $i_k \pmod{R/N^{k-1}} = i_{k+1}$, and, for all output switches $O^{j_2,\ldots,j_{k+1}}_{i_{k+1}} \in V_{2K+3-k}$ and $O^{j_2,\ldots,j_k}_{i_k} \in V_{2K+4-k}$, there is an edge $O^{j_2,\ldots,j_{k+1}}_{i_{k+1}} O^{j_2,\ldots,j_k}_{i_k}$ if $i_k \pmod{R/N^{k-1}} = i_{k+1}$. There are complete bipartite graphs between V_{K+1} and V_{K+2} and between V_{K+2} and V_{K+3}: for all $I^{j_2,\ldots,j_{K+1}}_{i_{K+1}} \in V_{K+1}$ and $M^{j_2,\ldots,j_{K+1}}_{i_{K+1}} \in V_{K+2}$, there is an edge $I^{j_2,\ldots,j_{K+1}}_{i_{K+1}} M^{j_2,\ldots,j_{K+1}}_{i_{K+2}}$, and for all $M^{j_2,\ldots,j_{K+1}}_{i_{K+2}} \in V_{K+2}$ and $O^{j_2,\ldots,j_{K+1}}_{i_{K+1}} \in V_{K+3}$, there is an edge $M^{j_2,\ldots,j_{K+1}}_{i_{K+2}} O^{j_2,\ldots,j_{K+1}}_{i_{K+1}}$.*

Given a Clos network $C^K_{N,R}$, the *sub-network* $C^{j_2,\ldots,j_k}$ designates the sub-graph of $C^K_{N,R}$ induced by the set of all vertices in V_k through V_{2K+4-k} whose superscript is prefixed with $j_2, \ldots, j_k$. In particular, the *sub-networks* C^{j_2,j_3} are called *direct sub-networks*. In the context of the minimum congestion problem, a routing r for a set $\mathcal{F}$ of commodities is now an assignment from each flow in the set to a sequence

consisting of $K-1$ sub-networks $C^{j_2,j_3}, C^{j_2,j_3,j_4}, \ldots, C^{j_2,j_3,j_4,\ldots,j_{K+1}}$ with common prefix followed by a middle switch $M_{i_{K+2}}^{j_2,j_3,j_4,\ldots,j_{K+1}}$.

A.2 Designing the Algorithm

The Melen-Turner algorithm can be generalized from 1-Clos to K-Clos networks. The idea is to use recursion: first, the Melen-Turner algorithm (from 1-Clos networks) is applied to the input commodities in the K-Clos network to divide the commodities over its direct sub-networks; then, the algorithm recurs on the commodities in each of the direct sub-networks.

The algorithm, called the *Generalized Melen-Turner Algorithm*, is detailed in Algorithm 2. The input is a set $\mathcal{F}$ of commodities in a Clos network $C_{N,R}^K$, and the output is a routing r for these commodities. First, interpreting the network $C_{N,R}^K$ as the network $C_{N,R}^1$ such that the i-th direct sub-network of $C_{N,R}^K$ corresponds to the i-th middle switch of $C_{N,R}^K$, the algorithm applies the Melen-Turner algorithm (from 1-Clos networks) to $\mathcal{F}$ in $C_{N,R}^K$. The routing of each commodity is updated by adding the direct sub-network of $C_{N,R}^K$ that the Melen-Turner algorithm assigns that commodity to. Second, if $K=1$, then the algorithm terminates. Otherwise, the algorithm calls the recursion on each of the direct sub-networks of $C_{N,R}^K$. For all $i \in [N]$, the set $\mathcal{F}^i$ of commodities that form the input to $C^{1,i}$ corresponds to the set of commodities that the Melen-Turner algorithm assigned to $C^{1,i}$, with their input and output switches mapped from $C_{N,R}^K$ to $C^{1,i}$: the commodities leaving input switch I_j^1 in $C_{N,R}^K$ leave input switch $I_{j \pmod{R/N}}^1$ in $C^{1,i}$, and those entering output switch O_j^1 in $C_{N,R}^K$ enter output switch $O_{j \pmod{R/N}}^{1,i}$ in $C^{1,i}$.

Algorithm 2 The generalized Melen-Turner algorithm for approximating a minimum congestion routing for a set F of flows in a Clos network $C_{N,R}^K$. In the initial call of the algorithm, the routing r corresponds to a empty assignment for each commodity.

function GeneralizedMelenTurner($\mathcal{F}, C_{N,R}^K, r$)
  $m \coloneqq$ MelenTurner($\mathcal{F}^i, C_{N,R}^K$)
  for f in F **do**
    $r(f) \coloneqq r(f) \circ m(f)$
  end for
  if $K = 1$ **then**
    return r
  else
    for $i \in [N]$ **do**
      $\mathcal{F}^i \coloneqq \{ (I_{i(f) \pmod{R/N}}^{1,i}, O_{j(f) \pmod{R/N}}^{1,i}) \mid f \in \mathcal{F}$ such that $m(f) = C^{1,i} \}$
      return GeneralizedMelenTurner($\mathcal{F}^i, C^{1,i}, r$)
    end for
  end if
end function

A.3 Analyzing the Algorithm

The next proposition shows that the generalization of the Melen-Turner algorithm to K-Clos networks approximates the minimum congestion by a factor of $O(\log K)$.

Proposition A.2. *For every set $\mathcal{F}$ of commodities in a Clos network $C_{N,R}^K$, Algorithm 2 returns a routing with congestion at most $O(\log K) \times OPT$, where OPT denotes the minimum congestion.*

Proof of Proposition A.2. The proof considers separately heavy and light commodities. We say that a commodity is *heavy* if its demand is greater than $1/2K \times OPT$, and that it is *light* otherwise. In Lemma A.4, we prove that the congestion over all heavy commodities is at most logarithmic in the number of stages; this lemma is supported by Claim A.3, which shows that the heavy commodities incident at each switch satisfy a particular structure. In Lemma A.6, we prove that the congestion over all light commodities is at most $2 \times OPT$; this lemma is supported by Claim A.5, which shows that, for every sub-network, the demand over light commodities is almost uniformly distributed over all its own sub-networks.

Since the statements and the proofs of the forthcoming claims and lemmas are analogous for both input and output switches, we present them only for input switches.

Claim A.3. *Fix a positive integer p. For every $k \in [2, K+1]$, the number of commodities traversing a switch $I_{i_k}^{j_2,\ldots,j_k}$ with demand at least $1/p \times OPT$ is at most $p \times N$.*

Proof. The proof is by induction on k.

- **Base case: $\mathbf{k = 2}$.** We show by contradiction that the statement holds for all switches $I_{i_2}^1$ in stage 2. Assume that the number of commodities traversing $I_{i_2}^1$ is at least $p \times N + 1$. Then, for every routing of the commodities, there is a edge leaving $I_{i_2}^1$ traversed by at least $p+1$ commodities. Since the demand of each commodity is at least $1/p \times OPT$, we deduce that the congestion of that edge is at least $(1 + 1/p) \times OPT$, thus arriving at a contradiction.

- **Inductive step: $\mathbf{2 < k \leq K + 1}$.** We assume that the statement holds for all switches $I_{i_{k-1}}^{j_1,\ldots,j_{k-1}}$ in stage $k-1$, and show that it also holds for all switches $I_{i_k}^{j_2,\ldots,j_k}$ in stage k. Consider the execution of the Melen-Turner algorithm in the sub-network $C^{j_2,\ldots,j_{k-1}}$. The algorithm creates multiple copies of each switch $I_{i_{k-1}}^{j_2,\ldots,j_{k-1}}$ in stage $k-1$. Since from the induction hypothesis there are at most $p \times N$ commodities with demand at least $1/p \times OPT$ traversing $I_{j_{k-1}}^{i_2,\ldots,i_{k-1}}$, the commodities with demand at least $1/p \times OPT$ are restricted to the first p copies of $I_{i_{k-1}}^{j_2,\ldots,j_{k-1}}$. Furthermore, the fact that the Melen-Turner algorithm assigns each commodity traversing a copy of $I_{i_{k-1}}^{j_2,\ldots,j_{k-1}}$ to a different edge implies that there are at most p commodities with demand at least $1/p \times OPT$ routed on a common edge from stage $k-1$ to stage k. Therefore, because there are N switches in stage $k-1$ linking to each switch $I_{i_k}^{j_2,\ldots,j_k}$ in stage k, the conclusion follows. □

Lemma A.4. *The congestion restricted to heavy commodities is at most $O(\log K) \times OPT$.*

Proof. We show that the congestion restricted to heavy flows on any edge $I_{i_k}^{j_2,\ldots,j_k} I_{i_{k+1}}^{j_2,\ldots,j_{k+1}}$ is at most $O(\log K)$. Consider the execution of the Melen-Turner algorithm in the sub-network $C^{j_2,\ldots,j_k}$. We make use of two immediate consequences of Claim A.3. First, the number of copies of $I_{i_k}^{j_2,\ldots,j_k}$ created by the algorithm is at most $2K$. Second, the demand of each commodity assigned to the l-th copy of $I_{i_k}^{j_2,\ldots,j_k}$ is at most $1/l \times OPT$ for all $l \in [2K]$. Therefore, since there is at most one commodity from each copy routed on edge $I_{i_k}^{j_2,\ldots,j_k} I_{i_{k+1}}^{j_2,\ldots,j_{k+1}}$, the congestion on that edge is at most $\sum_{l \in [2K]} 1/l \times OPT$, which is $O(\log K) \times OPT$. □

Claim A.5. *Fix a real c such that $0 < c \leq 1$. For all $k \in [2, K+1]$, the difference between the maximum and the minimum congestion restricted to commodities with demand less than c over all edges leaving a switch $I_{i_k}^{j_1,\ldots,j_k}$ is at most $2c$.*

Proof. Consider the execution of the Melen-Turner algorithm in the sub-network $C^{j_2,\ldots,j_k}$. Let x_1 be the highest indexed of copy of $I_{i_k}^{j_2,\ldots,j_k}$ containing a commodity with demand less than c, and x_2 be the number

of copies of $I_{i_k}^{j_1,\ldots,j_k}$. Denote respectively by ${}^{+}D_l$ and ${}^{+}d_l$ the demands of the heaviest and the lightest commodity contained in the l-th copy of $I_{i_k}^{j_2,\ldots,j_k}$. First, we respectively upper and lower bound the maximum and the minimum congestion restricted to commodities with demand less than c over all edges leaving $I_{j_k}^{j_2,\ldots,i_k}$ by

$$\max_{j_{k+1}\in[N]} c(I_{i_k}^{j_2,\ldots,j_k} I_{i_{k+1}}^{j_1,\ldots,j_{k+1}}) \leq \sum_{l\in[x_1,x_2]} {}^{+}D_l \quad \text{and}$$
$$\min_{j_{k+1}\in[N]} c(I_{i_k}^{j_2,\ldots,j_k} I_{i_{k+1}}^{j_1,\ldots,j_{k+1}}) \geq \sum_{l\in[x_1+1,x_2]} {}^{+}d_l$$

(where $i_{k+1} = i_k \pmod{R/N^{k-1}}$). In the x_1-th copy, the number of commodities with demand less than c may be strictly less than N, such that the difference between the maximum and the minimum number of commodities routed on some edge leaving $I_{j_k}^{i_2,\ldots,i_k}$ is at most 2.

Then, since commodities are assigned to copies in decreasing order of demands, we write

$$\begin{aligned}
\max_{j_{k+1}\in[N]} c(I_{i_k}^{j_2,\ldots,j_k} I_{i_{k+1}}^{j_2,\ldots,j_{k+1}}) - \min_{j_{k+1}\in[N]} c(I_{i_k}^{j_2,\ldots,j_k} I_{i_{k+1}}^{j_2,\ldots,j_{k+1}}) &\leq \sum_{l\in[x_1,x_2]} {}^{+}D_l - \sum_{l\in[x_1+1,x_2]} {}^{+}d_l \\
&\leq \sum_{l\in[x_1,x_2]} {}^{+}D_l - \sum_{l\in[x_1+2,x_2]} {}^{+}D_l \\
&= {}^{+}D_{x_1} + {}^{+}D_{x_1+1} \\
&\leq 2c
\end{aligned}$$

to arrive at the desired conclusion. □

The proof of the next lemma makes use of the following lower bound on the minimum congestion:

$$OPT \geq 1/N \sum_{f\in\mathcal{F}: i_2(k) \pmod{R/N^{l-2}} = i_l} d(k)$$

for all $l \in [2, K+1]$ and $i_l \in [R/N^{l-2}]$. In words, the congestion of an edge leaving a switch $I_{i_k}^{j_2,\ldots,j_k}$ in a minimum congestion routing with unsplittable flow is at least the congestion of that edge in a minimum congestion routing with splittable flow, which corresponds to the average demand over all commodities such that there is a path from its source to its destination containing that edge.

Lemma A.6. *The congestion restricted to light commodities is at most* $2 \times OPT$.

Proof. We show that the congestion restricted to light commodities on any edge $I_{i_k}^{j_2,\ldots,j_k} I_{i_{k+1}}^{j_2,\ldots,j_{k+1}}$ is at most $2 \times OPT$. First, we write the next inequalities:

$$c(I_{i_2}^1 I_{i_3}^{1,j_3}) \leq 1/N \sum_{f\in\mathcal{F}: i_2(k) = i_2} d(k) \;+\; 1/K \times OPT$$

for all $i_2 \in [R]$ such that $i_2 \pmod{R/N^{k-2}} = i_k$ (where $i_3 = i_2 \pmod{R/N}$), and

$$c(I_{i_l}^{j_2,\ldots,j_l} I_{i_{l+1}}^{j_2,\ldots,j_{l+1}}) \leq 1/N \sum_{\substack{i_{l-1}\in[R/N^{k-3}]: \\ i_{l-1} \pmod{R/N^{l-2}} = i_l}} c(I_{i_{l-1}}^{j_2,\ldots,j_{l-1}} I_{i_l}^{j_2,\ldots,j_l}) \;+\; 1/K \times OPT$$

for all $l \in [2,k]$ and $i_l \in [R/N^{l-2}]$ such that $i_l \pmod{R/N^{k-2}} = i_k$. In words, the congestion on edge $I_{i_2}^1 I_{i_3}^{1,j_3}$ is at most the average demand over all commodities entering switch $I_{i_2}^1$ plus $1/K \times OPT$, and, similarly, the

congestion on an edge $I_{il}^{j_2,\ldots,j_l} I_{il+1}^{j_2,\ldots,j_{l+1}}$ is at most the average congestion over all edges entering switch $I_{il}^{j_2,\ldots,j_l}$ plus $1/K \times OPT$. These inequalities follow from Claim A.5 together with the fact that the total demand entering a switch equals the total demand leaving that switch.

Second, we apply the previous inequalities in succession,

$$\begin{aligned}
c(I_{i_k}^{j_2,\ldots,j_k} I_{i_{k+1}}^{j_2,\ldots,j_{k+1}}) &\leq 1/N \sum_{\substack{i_{k-1}\in[R/N^{k-3}]:\\ i_{k-1} \pmod{R/N^{k-2}}=i_k}} c(I_{i_{k-1}}^{j_2,\ldots,j_{k-1}} I_{i_k}^{j_2,\ldots,j_k}) \quad + \quad 1/K \times OPT \\
&\leq 1/N^{K-1} \sum_{\substack{i_2\in[R]:\\ i_2 \pmod{R/N^{k-1}}=i_{k+1}}} c(I_{i_2}^{1} I_{i_3}^{1,j_3}) \quad + \quad (k-1)/K \times OPT \\
&\leq 1/N^{K} \sum_{\substack{f\in\mathcal{F}:\\ i_2(k) \pmod{R/N^{k-1}}=i_{k+1}}} d(f) \quad + \quad k/K \times OPT \\
&\leq 2 \times OPT
\end{aligned}$$

to conclude that $c(I_{i_k}^{j_1,\ldots,j_k} I_{i_{k+1}}^{j_1,\ldots,j_{k+1}}) \leq 2 \times OPT$. □

The proof of the proposition is immediate from Lemmas A.4 and A.6. □

Similarly to the proof of Theorem 5.1 concerning the approximation guarantees of our new algorithm, the proof of the previous theorem can also be generalized to show that the algorithm not only approximates the minimum congestion by a factor of $O(\log K)$, but also returns a routing with congestion at most $O(\log K)$. This generalization requires only the adaptation of the base case of the proof of Claim A.3, and the replacement of the term OPT with $\min\{1, OPT\}$ everywhere in the proof.

Proposition A.7. *For every set $\mathcal{F}$ of commodities in a Clos network $C_{N,R}^K$, Algorithm 2 returns a routing with congestion at most $O(\log K) \times \min\{OPT, 1\}$, where OPT denotes the minimum congestion.*

References

[1] Bernhard Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer Publishing Company, Incorporated, 5th edition, 2012.

[2] Prabhakar Raghavan and Clark Tompson. Randomized Rounding: A Technique for Provably Good Algorithms and Algorithmic Proofs. *Combinatorica*, 7(4):365–374, 1987.

[3] David Harris and Aravind Srinivasan. The Moser–Tardos Framework with Partial Resampling. *The Journal of the ACM*, 66(5):1–45, 2019.

[4] Thomson Leighton, Satish Rao, and Aravind Srinivasan. Multicommodity Flow and Circuit Switching. In *Proceedings of the Hawaii International Conference on System Sciences*, volume 7, pages 459–465, 1998.

[5] Julia Chuzhoy, Venkatesan Guruswami, Sanjeev Khanna, and Kunal Talwar. Hardness of Routing with Congestion in Directed Graphs. In *Proceedings of the ACM Symposium on Theory of Computing*, page 165–178, 2007.

[6] Steve Cosares and Iraj Saniee. An Optimization Problem related to Balancing Loads on SONET Rings. *Telecommunication Systems*, 3(2):165–181, 1994.

[7] Alexander Schrijver, Paul Seymour, and Peter Winkler. The Ring Loading Problem. *SIAM Journal on Discrete Mathematics*, 11(1):1–14, 1998.

[8] Karl Däubel. An Improved Upper Bound for the Ring Loading Problem. *SIAM Journal on Discrete Mathematics*, 36(2):867–887, 2022.

[9] Richard Shapley and David B Shmoys. Small Additive Error for Unsplittable Multicommodity Flow in Outerplanar Graphs. In *International Workshop on Approximation and Online Algorithms*, pages 167–182. Springer, 2024.

[10] Mohammed Majthoub Almoghrabi, Martin Skutella, and Philipp Warode. Integer and Unsplittable Multiflows in Series-Parallel Digraphs. In *Proceedings of the International Conference on Integer Programming and Combinatorial Optimization*, pages 427–441. Springer, 2025.

[11] Jon Kleinberg. Single-Source Unsplittable Flow. In *Proceedings of IEEE Symposium on Foundations of Computer Science*, pages 68–77, 1996.

[12] Yefim Dinitz, Naveen Garg, and Michel Goemans. On the Single-Source Unsplittable Flow Problem. *Combinatorica*, 19(1):17–41, 1999.

[13] Georg Baier, Ekkehard Köhler, and Martin Skutella. The k-Splittable Flow Problem. *Algorithmica*, 42(3):231–248, 2005.

[14] James Aspnes, Yossi Azar, Amos Fiat, Serge Plotkin, and Orli Waarts. Online routing of virtual circuits with applications to load balancing and machine scheduling. *Journal of the ACM*, 44(3):486–504, 1997.

[15] Serge Plotkin. Competitive Routing of Virtual Circuits in ATM Networks. *IEEE Journal on Selected Areas in Communications*, 13(6):1128–1136, 2002.

[16] Richard Barry and Pierre Humblet. On the Number of Wavelengths and Switches in All-optical Networks. *IEEE Transactions on Communications*, 42(234):583–591, 2002.

[17] Alok Aggarwal, Amotz Bar-Noy, Don Coppersmith, Rajiv Ramaswami, Baruch Schieber, and Madhu Sudan. Efficient Routing and Scheduling Algorithms for Optical Networks. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, page 412–423, USA, 1994. Society for Industrial and Applied Mathematics.

[18] Prabhakar Raghavan and Eli Upfal. Efficient Routing in All-Optical Networks. In *Proceedings of the ACM symposium on Theory of Computing*, pages 134–143, 1994.

[19] Richard M Karp, Frank Thomson Leighton, Ronald L Rivest, Clark D Thompson, Umesh V Vazirani, and Vijay V Vazirani. Global Wire Routing in Two-Dimensional Arrays. *Algorithmica*, 2(1):113–129, 1987.

[20] Mohammad Al-Fares, Sivasankar Radhakrishnan, Barath Raghavan, Nelson Huang, and Amin Vahdat. Hedera: Dynamic Flow Scheduling for Data Center Networks. In *Proceedings of the USENIX Conference on Networked Systems Design and Implementation*, pages 89–92, 2010.

[21] Mohammad Alizadeh, Tom Edsall, Sarang Dharmapurikar, Ramanan Vaidyanathan, Kevin Chu, Andy Fingerhut, Vinh The Lam, Francis Matus, Rong Pan, Navindra Yadav, and George Varghese. CONGA: Distributed Congestion-Aware Load Balancing for Data Centers. *ACM SIGCOMM Computer Communication Review*, 44(4):503–514, 2014.

[22] Ankit Singla, Philip Brighten Godfrey, and Alexandra Kolla. High Throughput Data Center Topology Design. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation*, pages 29–41, 2014.

[23] Marco Chiesa, Guy Kindler, and Michael Schapira. Traffic Engineering With Equal-Cost-MultiPath: An Algorithmic Perspective. *IEEE/ACM Transactions on Networking*, 25(2):779–792, 2017.

[24] Pooria Namyar, Sucha Supittayapornpong, Mingyang Zhang, Minlan Yu, and Ramesh Govindan. A Throughput-Centric View of the Performance of Datacenter Topologies. In *Proceedings of the ACM SIGCOMM Conference*, page 349–369, 2021.

[25] Mubashir Qureshi, Yuchung Cheng, Qianwen Yin, Qiaobin Fu, Gautam Kumar, Masoud Moshref, Junhua Yan, Van Jacobson, David Wetherall, and Abdul Kabbani. PLB: Congestion Signals are Simple and Effective for Network Load Balancing. In *Proceedings of the ACM SIGCOMM Conference*, pages 207–218, 2022.

[26] Mohammad Al-Fares, Alexander Loukissas, and Amin Vahdat. A Scalable, Commodity Data Center Network Architecture. *ACM SIGCOMM Computer Communication Review*, 38(4):63–74, 2008.

[27] Albert Greenberg, James Hamilton, Navendu Jain, Srikanth Kandula, Changhoon Kim, Parantap Lahiri, David Maltz, Parveen Patel, and Sudipta Sengupta. VL2: A Scalable and Flexible Data Center Network. *ACM SIGCOMM Computer Communication Review*, 39(4):51–62, 2009.

[28] Radhika Niranjan Mysore, Andreas Pamboris, Nathan Farrington, Nelson Huang, Pardis Miri, Sivasankar Radhakrishnan, Vikram Subramanya, and Amin Vahdat. PortLand: A Scalable Fault-Tolerant Layer 2 Data Center Network Fabric. *ACM SIGCOMM Computer Communication Review*, 39(4):39–50, 2009.

[29] Kun Qian, Yongqing Xi, Jiamin Cao, Jiaqi Gao, Yichi Xu, Yu Guan, Binzhang Fu, Xuemei Shi, Fangbo Zhu, and Rui Miao. Alibaba HPN: A Data Center Network for Large Language Model Training. In *Proceedings of the ACM SIGCOMM Conference*, pages 691–706, 2024.

[30] Adithya Gangidi, Rui Miao, Shengbao Zheng, Sai Jayesh Bondu, Guilherme Goes, Hany Morsy, Rohit Puri, Mohammad Riftadi, Ashmitha Jeevaraj Shetty, and Jingyi Yang. RDMA Over Ethernet for Distributed Training at Meta Scale. In *Proceedings of the ACM SIGCOMM Conference*, pages 57–70, 2024.

[31] Ka-Cheong Leung, Victor Li, and Daiqin Yang. An Overview of Packet Reordering in Transmission Control Protocol (TCP): Problems, Solutions, and Challenges. *IEEE Transactions on Parallel and Distributed Systems*, 18(4):522–535, 2007.

[32] Costin Raiciu, Sebastien Barre, Christopher Pluntke, Adam Greenhalgh, Damon Wischik, and Mark Handley. Improving Datacenter Performance and Robustness with Multipath TCP. *ACM SIGCOMM Computer Communication Review*, 41(4):266–277, 2011.

[33] Advait Dixit, Pawan Prakash, Yu Hu, and Ramana Rao Kompella. On the Impact of Packet Spraying in Data Center Networks. In *Proceedings of the IEEE International Conference on Computer Communications*, pages 2130–2138, 2013.

[34] Keqiang He, Eric Rozner, Kanak Agarwal, Wes Felter, John Carter, and Aditya Akella. Presto: Edge-Based Load Balancing for Fast Datacenter Networks. *ACM SIGCOMM Computer Communication Review*, 45(4):465–478, 2015.

[35] Charles Clos. A Study of Non-Blocking Switching Networks. *The Bell System Technical Journal*, 32(2):406–424, 1953.

[36] Junming Xu. *Topological Structure and Analysis of Interconnection Networks*. Springer, 1st edition, 2010.

[37] Hitesh Ballani, Paolo Costa, Thomas Karagiannis, and Ant Rowstron. Towards Predictable Datacenter Networks. *ACM SIGCOMM Computer Communication Review*, 41(4):242–253, 2011.

[38] Hitesh Ballani, Keon Jang, Thomas Karagiannis, Changhoon Kim, Dinan Gunawardena, and Greg O'Shea. Chatty Tenants and the Cloud Network Sharing Problem. In *Proceedings of the USENIX Conference on Networked Systems Design and Implementation*, pages 171–184, 2013.

[39] Jeongkeun Lee, Yoshio Turner, Myungjin Lee, Lucian Popa, Sujata Banerjee, Joon-Myung Kang, and Puneet Sharma. Application-Driven Bandwidth Guarantees in Datacenters. *ACM SIGCOMM Computer Communication Review*, 44(4):467–478, 2014.

[40] Nick Duffield, Pawan Goyal, Albert Greenberg, Partho Mishra, Kadangode Ramakrishnan, and Jacobus van der Merive. A Flexible Model for Resource Management in Virtual Private Networks. *ACM SIGCOMM Computer Communication Review*, 29(4):95–108, 1999.

[41] Mohammad Alizadeh, Shuang Yang, Milad Sharif, Sachin Katti, Nick McKeown, Balaji Prabhakar, and Scott Shenker. pFabric: Minimal Near-Optimal Catacenter Transport. *ACM SIGCOMM Computer Communication Review*, 43(4):435–446, 2013.

[42] Mosharaf Chowdhury, Yuan Zhong, and Ion Stoica. Efficient Coflow Scheduling with Varys. In *Proceedings of the ACM SIGCOMM Conference*, page 443–454, 2014.

[43] Frank Hwang. Control Algorithms for Rearrangeable Clos Networks. *IEEE Transactions on Communications*, 31(8):952–954, 1983.

[44] László Lovász and Michael Plummer. *Matching Theory*. American Mathematical Society, 2009.

[45] Abdul Kabbani, Balajee Vamanan, Jahangir Hasan, and Fabien Duchene. Flowbender: Flow-level Adaptive Routing for Improved Latency and Throughput in Datacenter Networks. In *Proceedings of the ACM Conference on Emerging Networking EXperiments and Technologies*, pages 149–160, 2014.

[46] Naga Katta, Mukesh Hira, Changhoon Kim, Anirudh Sivaraman, and Jennifer Rexford. Hula: Scalable Load Balancing using Programmable Data Planes. In *Proceedings of the Symposium on SDN Research*, pages 1–12, 2016.

[47] Riccardo Melen and Jonathan Turner. Nonblocking Multirate Networks. *SIAM Journal on Computing*, 18(2):301–313, 1989.

[48] Andrew Yao. Probabilistic Computations: Towards a Unified Measure of Complexity. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 222–227, 1977.

[49] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[50] Gregory Pfister. An Introduction to the Infiniband Architecture. *High Performance Mass Storage and Parallel I/O*, 42(617-632):10, 2001.

[51] IEEE Standard for Ethernet. *IEEE Std 802.3-2022 (Revision of IEEE Std 802.3-2018)*, pages 1–7025, 2022.

[52] Claudio Desanti, Robert Nixon, and Craig Carlson. Transmission of IPv6, IPv4, and Address Resolution Protocol (ARP) Packets over Fibre Channel. RFC 4338, January 2006.

[53] Mark Birrittella, Mark Debbage, Ram Huggahalli, James Kunz, Tom Lovett, Todd Rimmer, Keith Underwood, and Robert Zak. Intel® Omni-path Architecture: Enabling Scalable, High Performance Fabrics. In *Proceedings of the IEEE Symposium on High-Performance Interconnects*, pages 1–9, 2015.

[54] Arindam Khan and Mohit Singh. On Weighted Bipartite Edge Coloring. In *Proceedings of the IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, 2015.

[55] Hung Ngo and Van Vu. Multirate Rearrangeable Clos networks and a Generalized Edge-Coloring Problem on Bipartite Graphs. *SIAM Journal on Computing*, 32(4):1040–1049, 2003.

[56] Richard Karp. Reducibility Among Combinatorial Problems. In *Proceedings of a Symposium on the Complexity of Computer Computations*, page 85–103, 1972.

[57] Xiaoqiao Meng, Vasileios Pappas, and Li Zhang. Improving the Scalability of Data Center Networks with Traffic-Aware Virtual Machine Placement. In *Proceedings of the IEEE INFOCOM Conference*, pages 1–9, 2010.

[58] Yuanwei Lu, Guo Chen, Bojie Li, Kun Tan, Yongqiang Xiong, Peng Cheng, Jiansong Zhang, Enhong Chen, and Thomas Moscibroda. Multi-Path Transport for RDMA in Datacenters. In *Proceedings of the USENIX Conference on Networked Systems Design and Implementation*, pages 357–371, 2018.