Impossibility Results for Data-Center Routing with Congestion Control and Unsplittable Flows

Miguel Alves Ferreira
CMU, USA
Inst. de Telecomunicações, IST/ULisboa, Portugal
maferrei@andrew.cmu.edu

Nirav Atre
CMU, USA
natre@andrew.cmu.edu

Justine Sherry
CMU, USA
sherry@cs.cmu.edu

João Luís Sobrinho
Inst. de Telecomunicações, IST/ULisboa, Portugal
joao.sobrinho@lx.it.pt

ABSTRACT

Clos networks have a long history in networking. In early telephone networks and classic network flow problems, Clos networks have been shown to emulate the performance properties of an ideal *macro-switch* connecting sources to destinations. Therefore, Clos networks are a natural choice for modern data-centers, and are widely deployed. However, data-centers operate on different traffic assumptions than those prevalent in telephone networks and network flow problems: sources and destinations are not limited to at most one flow, and each flow must be assigned to a single path. Subject to these constraints, the performance of a Clos network is no longer equivalent to that of a macro-switch.

In this paper, we study the discrepancies between a Clos network and a macro-switch in terms of throughput and fairness, considering routing inside the network as a design variable. In this context, we prove three fundamental results regarding the performance of Clos networks. First, we show that even if routing could replicate the macro-switch rates, imposing max-min fair rates halves the throughput. Second, we prove that routing for max-min fairness reduces the max-min fair rates of some flows by a factor of $\frac{1}{n}$ relative to the macro-switch, where n is the number of middle switches in the network. Finally, we find that routing for maximum throughput doubles the throughput but brings the max-min fair rates of some flows to zero. These results call into question several common assumptions about the design options for data-centers.

CCS CONCEPTS

• **Networks** → **Network performance evaluation**.

KEYWORDS

Data-center, clos network, routing, rate allocation, unsplittable flow

ACM Reference Format:
Miguel Alves Ferreira, Nirav Atre, Justine Sherry, and João Luís Sobrinho. 2024. Impossibility Results for Data-Center Routing with Congestion Control and Unsplittable Flows. In *ACM Symposium on Principles of Distributed Computing (PODC '24), June 17–21, 2024, Nantes, France.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3662158.3662777

PODC '24, June 17–21, 2024, Nantes, France

ACM ISBN 979-8-4007-0668-4/24/06
https://doi.org/10.1145/3662158.3662777

1 INTRODUCTION

Many data-centers are architected following *Clos networks* [2, 16, 29, 30]. In a Clos network, each server is connected to a single top-of-rack (ToR) switch, and each ToR switch is connected to multiple middle switches, such that there are as many link-disjoint paths from input to output ToR switches as there are middle switches [12]. Typically, each link has uniform capacity.

A basic property of Clos networks is their *full bisection bandwidth*, meaning that the minimum capacity of a global cut (separating all source-destination pairs) consisting of links *inside* the network (between ToR and middle switches) is at least the minimum capacity of a global cut consisting of links *outside* the network (between servers and ToR switches). This property has two well-known implications that make them an attractive design for a range of use cases, including telephone networks and data-centers.

Demand satisfaction: When flows are *splittable*, in the sense that they can be routed concurrently on possibly multiple source-destination paths (and thus be modeled as *classic network flow*), arbitrary *flow demands* (i.e., *flow rates* satisfying the capacities of links outside the network) can be routed inside the network such that the capacities of these links are satisfied [9]. Thus, the bottleneck of a flow is the link between its source and input switch, or the link between its destination and output switch, such that the inside of the network can be abstracted away from a performance modeling standpoint.

Throughput maximization: When each source and each destination is limited to at most one flow (due to *admission control*, as in early telephone networks), arbitrary flow demands can be routed on link-disjoint paths such that the capacities inside the network are also satisfied, and, if flows demands equal link capacities, then the *throughput* across the network (i.e., the total rate over all flows) is maximized [19].

However, a typical data-center does not meet the above premises that guarantee demand satisfaction or throughput maximization [2, 16, 29, 30]. Instead: (1) flows are usually *unsplittable*, in the sense that they must be routed on a single source-destination path, and (2) flows are usually subject to *congestion control*, whereby the network accepts all input flows (possibly multiple flows per source and per destination) and imposes a *max-min fair allocation* [6, 28] of the link capacities among the flow rates; a max-min allocation maximizes in *lexicographic order* [13] the vector whose components are the flow rates sorted from lowest to highest rate. Given these discrepancies, the overarching question of this work is: *How well do Clos networks perform with respect to satisfying flow demands and maximizing throughput given unsplittable flows and congestion control?*

1.1 Research Questions

The standard abstraction for analyzing data-center performance is a *macro-switch* [1, 3, 5, 8, 10, 20, 26].[1] A macro-switch is obtained from a Clos network by replacing all middle switches by a complete bipartite graph connecting input to output ToR switches with infinite capacity links, thus emulating a single switch connecting sources to destinations. The flow rates in a macro-switch match the flow demands in the corresponding Clos network, and are restricted only by the capacities of links outside the network.

Under admission control, if there is at most one flow per source and per destination, and flows are transmitted at link capacities, then the throughput across a macro-switch is maximized. However, under congestion control, there are possibly multiple flows per source and per destination, and flows are transmitted at max-min fair rates. Thus, our first question concerns only a macro-switch subject to congestion control: **(Q1) What is the throughput lost, if any, by the max-min fair allocation in a macro-switch relative to the maximum throughput across the macro-switch without max-min fair constraints?**[2]

In a macro-switch, there is a unique *routing*, for there is a single path between every source-destination pair. In contrast, in a Clos network, there are possibly multiple routings, determined by the source-destination path assigned to each flow. Therefore, flow rates depend on both routing and congestion control: there is a possibly different max-min fair allocation for each routing, such that changing the path assigned to some flow may affect the max-min fair rate of every other flow, even the rates of flows that do not share common links with that flow.

With splittable flows, a Clos network and a macro-switch are equivalent, meaning that the divisibility of flows ensures that arbitrary macro-switch rates can be replicated in the network. With admission control, they are also equivalent, now meaning that the link-disjoint routing of flows ensures that some maximum throughput macro-switch rates can also be replicated in the network. However, with congestion control, it is not known if there are max-min fair rates in a Clos network that replicate, either *exactly* or even *closely*, the max-min fair macro-switch rates. Thus, our second question is: **(Q2) To what extent can max-min fair allocations in a Clos network replicate the max-min fair allocation in the corresponding macro-switch?**

Many evaluations of data-center routing algorithms assume that their primary objective is *maximizing throughput*, such that fairness is a secondary objective ensured by congestion control [17, 26, 31, 32]. As both throughput and fairness in a Clos network are functions of the routing, we investigate a new angle of the classic throughput-fairness trade-off: the effect of routing on throughput and fairness when maximizing throughput while having congestion control maintain fairness constraints at each routing. Specifically: **(Q3) If routing maximizes throughput, then how does the max-min fair allocation in the network compare to that in the corresponding macro-switch?**

[1] A macro-switch has several denominations in the networking literature, including *hose model* [14], *fully-non-blocking switch* [3] and *big-switch model* [1].

[2] The gap between the throughput of an allocation subject to fairness constraints and the maximum throughput not subject to these constraints is known in the resource allocation literature as the *price of fairness* [7].

1.2 Contributions

The main contribution of this paper is the characterization of the impossibility of satisfying demands and maximizing throughput in data-center Clos deployments subject to unsplittable flows and max-min fair allocations at each routing via results **R1** to **R3**, which answer, respectively, questions **Q1** through **Q3**.

(R1) Impossibility of maximizing throughput assuming satisfiable demands: We show that the throughput of the max-min fair allocation in a macro-switch can be less than the maximum throughput across it (without max-main fair constraints).

Theorem 1.1. (Informal statement of Theorem 3.4) *For every macro-switch, the throughput of the max-min fair allocation is at least half of the maximum throughput; this bound is tight.*

While it is well-known that, in general, max-min fair allocations do not maximize throughput [6, 24], it is surprising that, given the simplicity of a macro-switch, up to half the maximum throughput may be forfeited.

(R2) Impossibility of satisfying demands: Lexicographic (lex) max-min fairness is the natural extension of max-min fairness from fixed to variable routing: a *lex-max-min fair allocation* maximizes in lexicographic order the vectors corresponding to max-min fair allocations for each routing over all routings [22]. We show that routing in a Clos network does not always *exactly* replicate the max-min fair macro-switch rates, and, moreover, it does not always replicate them *closely* when optimizing lex-max-min fairness.

Theorem 1.2. (Informal statement of Theorem 4.3) *For every Clos network, there is a collection of flows such that the lex-max-min fair rates of some flows are smaller than their max-min fair rates in the macro-switch by a $\frac{1}{n}$-factor, where n is the number of middle switches.*

In other words, with lex-max-min fair rates, which are the fairest rates in a Clos network in a max-min sense, some flows may severely decrease their max-min fair rates relative to the macro-switch.

(R3) Incongruence between maximizing throughput and satisfying demands: A *throughput-max-min fair allocation* maximizes the throughput of the max-min fair allocation for each routing over all routings. We show that the throughput of a throughput max-min fair allocation in a Clos network can be greater than the throughput of the max-min fair allocation in its macro-switch.

Theorem 1.3. (Informal statement of Theorem 5.4) *For every Clos network, the throughput of a throughput-max-min fair allocation is at most twice that of the max-min fair allocation in its macro-switch; this bound becomes tight as the number of middle switches increases.*

This throughput increase, however, implies seriously reducing some flow rates relative to its macro-switch; in particular, doubling the throughput requires zeroing the rates of most flows.

1.3 Roadmap

In §2, we present a formal description of the problem setting. In §3, we show the impossibility of maximizing throughput if demands are assumed satisfiable and, in §4, the impossibility of satisfying these demands. In §5, we show the incongruence between maximizing throughput and satisfying demands. In §6, we review related work, and, in §7, we conclude the paper.

2 PROBLEM SETTING

We formalize our model of data-center Clos deployments with unsplittable flows subject to max-min fair rate allocations at each routing. In §2.1, we describe Clos networks and their macro-switch abstractions. In §2.2, we define max-min fair allocations and highlight their dependence on routing. In §2.3, we introduce the routing objectives considered in this paper.

2.1 Clos networks and macro-switches

Clos networks: A Clos network interconnects *source servers* to *destination servers* via three stages of switches: *input ToR switches*, *middle switches*, and *output ToR switches*. For simplicity, we assume that the degree of each middle switch is twice that of each ToR switch. The Clos network of size n, for some integer $n \geq 1$, denoted by C_n and depicted in Figure 1a for $n = 2$, designates the directed graph with the following node and link sets:

- **Node set:** There are n middle switches, with the m'th middle switch denoted by M_m, $m \in [n]$. There are $2n$ input (output) switches, with the i'th input (output) switch denoted by I_i (O_i), $i \in [2n]$. There are $2n^2$ source (destination) servers, with the j'th source (destination) servers of the i'th input (output) switch denoted by s_i^j (t_i^j), $i \in [2n]$ and $j \in [n]$.[3]
- **Link set:** There is one link $I_i M_m$ ($M_m O_i$), $i \in [2n]$ and $m \in [n]$. There is one link $s_i^j I_i$ ($O_i t_i^j$), $i \in [2n]$ and $j \in [n]$.

In C_n, there are n paths between every source-destination pair, each via a different middle switch. All links have unit capacity.

Macro-switches: The *macro-switch abstraction* of a Clos network of size n, denoted by MS_n and depicted in Figure 1b for $n = 2$, designates the directed graph obtained from the Clos network by replacing all middle switches (and respective links) with the complete bipartite graph whose start and end node sets are, respectively, the input and output switch sets. In MS_n, there is a single path between each source-destination pair. Links between servers and ToR switches have unit capacity, and links between ToR switches have infinite capacity. Therefore, the macro-switch emulates a single switch connecting $2n^2$ sources to $2n^2$ destinations.

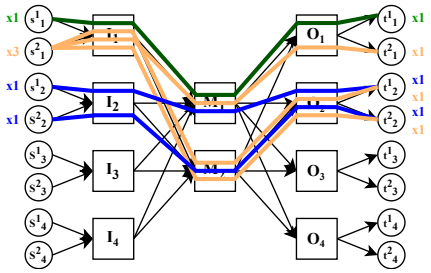
2.2 Routing and max-min fair allocations

Routing and allocations: A *flow* f maps to a source-destination pair (s_f, t_f); there may be multiple flows mapping to the same source-destination pair. Given a collection F of flows, a *routing* r is an assignment of each flow f to an s_f-t_f path, $r(f)$. Given a routing r for the collection F of flows, an *allocation* a is an assignment of each flow f to a non-negative rate $a(f)$. Let $c(u, v)$ be the capacity of a link (u, v). We say that an allocation a is *feasible* if for every link the total rate over all flows traversing that link is at most the link capacity:

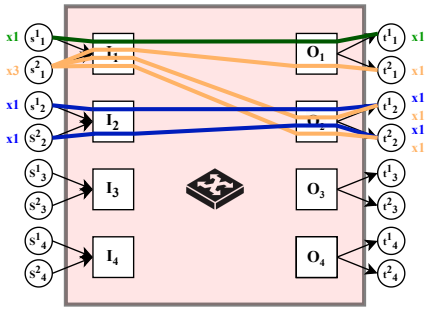
$$\sum_{f \in F:(u,v) \in r(f)} a(f) \leq c(u, v) \text{ for all links } (u, v).$$

The set of all feasible allocations is denoted by A. The *throughput* of an allocation a, denoted by $t(a)$, is the total rate over all flows.

[3] Given two non-negative integers x and y such that $x \leq y$, the notation $[x, y]$ denotes the set $\{x, x+1, \ldots, y\}$. If $x = 1$, then the notation becomes $[y]$.



(a) The Clos network C_2.



(b) The macro-switch MS_2.

Figure 1: A collection of flows in the Clos network for n = 2 and its macro-switch abstraction. Circles symbolize servers, and squares symbolize switches. The large colored box symbolizes the single switch. Lines symbolize flows from source to destination, with numbers next to sources and destinations (*e.g.*, ×3) indicating the number of flows leaving those sources or entering those destinations.

Max-min fair allocations: Let $a\uparrow$ be the sorted vector of an allocation a whose components are the rates sorted from lowest to highest rate. Given two allocations $a\uparrow$ and $a'\uparrow$, we write $a\uparrow \geq a'\uparrow$ if $a\uparrow$ is greater than or equal to $a'\uparrow$ in lexicographic order [13].

Definition 2.1 ([6, 28]). An allocation a is *max-min fair* if: (1) a is feasible; and (2) $a\uparrow$ is maximum in lexicographic order over the sorted vectors of all feasible allocations:

$$a \in A, \text{ and } a\uparrow \geq a'\uparrow \text{ for all } a' \in A.$$

A max-min fair allocation can be found in polynomial time in the number of flows and links by a *water-filling algorithm* [6, 28].

Bottleneck property: A well-known characterization of a max-min fair allocation is given by the *bottleneck property*. We say that a link (u, v) traversed by a flow f is a bottleneck for f if: (1) the total rate over all flows traversing (u, v) equals the link capacity; and (2) the rate of f is maximum over those of all flows traversing (u, v):

$$\sum_{f' \in F:(u,v) \in r(f')} a(f') = c(u, v), \text{ and}$$

$$a(f) \geq a(f') \text{ for all } f' \in F \text{ such that } (u, v) \in r(f').$$

A link may be a bottleneck for multiple flows, and a flow may have multiple bottleneck links. While in a macro-switch a flow may only be bottlenecked on links between servers and ToR switches, in a Clos network it may also be bottlenecked on links between ToR and middle switches.

Lemma 2.2 ([6, 28]). *A feasible allocation is max-min fair if and only if all flows have a bottleneck link.*

The forthcoming example first instantiates the computation of the max-min fair allocation in a macro-switch. Then, it demonstrates that, when going from the macro-switch to the corresponding Clos network, flows may transfer their bottleneck links from links outside to links inside the network.

Example 2.3. Consider the collection of flows in the Clos network for $n = 2$ illustrated in Figure 1. There are three types of flows, each identified with a different color in the figure:

- *Type 1 (orange)*: There is one flow (s_1^2, t_1^2), one flow (s_1^2, t_2^1), and one flow (s_1^2, t_2^2).
- *Type 2 (blue)*: There is one flow (s_2^i, t_2^i), $i \in [2]$.
- *Type 3 (green)*: There is one flow (s_1^1, t_1^1).

In a macro-switch, there is a unique routing, implying that there is a unique max-min fair allocation for each collection of flows. In Figure 1b, we derive the max-min fair allocation in the macro-switch. First, since $s_1^2 I_1$ is traversed by the greatest number of flows, each type 1 flow is assigned rate $\frac{1}{3}$ and bottlenecked on $s_1^2 I_1$. Second, since the rate of the type 2 flow (s_2^1, t_2^1) is only constrained by the rate $\frac{1}{3}$ that the type 1 flow (s_1^2, t_2^1) consumes from $O_2 t_2^1$, it is assigned rate $\frac{2}{3}$ and bottlenecked on $O_2 t_2^1$; likewise, the type 2 flow (s_2^2, t_2^2) is assigned rate $\frac{2}{3}$ and bottlenecked on $O_2 t_2^2$. Finally, since the rate of the type 3 flow is not constrained by the rates of other flows, it is assigned rate 1 and bottlenecked on both $s_1^1 I_1$ and $O_1 t_1^1$. The sorted vector of the max-min fair allocation is $\left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, 1\right]$.

In a Clos network, there are multiple routings, implying that there are possibly multiple max-min fair allocation depending on the routing for each collection of flows. In Figure 1a, we show a possible routing in the Clos network. First, suppose that the type 1 flow (s_1^2, t_2^1) is assigned to M_1. Then, type 1 and type 2 flows keep the same bottlenecks links that in the macro-switch, whereas the type 3 flow transfers its bottleneck to $I_1 M_1$; hence, the type 1 and type 2 flows uphold their macro-switch rates, while the type 3 flow decreases its rate to $\frac{2}{3}$. Second, suppose that the type 1 flow (s_1^2, t_2^1) is re-assigned to M_2. Then, the type 3 flow keeps the same bottleneck links that in the macro-switch, thus increasing its rate back to 1; contrarily, the type 2 flow (s_2^2, t_2^2) now transfers its bottleneck to $M_2 O_2$, thus decreasing its rate to $\frac{1}{3}$. The sorted vector $\left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, \frac{2}{3}\right]$ of the max-min fair allocation for the first routing is greater in lexicographic order than the sorted vector $\left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, 1\right]$ for the second routing; the sorted vector of the max-min fair allocation in the macro-switch is greater than the latter two.

2.3 Routing objectives

Lex-max-min fairness: The first objective, considered in §4, is to maximize in lexicographic order the sorted vectors of the max-min fair allocations in a Clos network [22]. Given a collection of flows in a Clos network C_n, denote by R the set of all routings in C_n, and by a_r^{MmF} the max-min fair allocation for routing $r \in R$.

Definition 2.4. A *lex-max-min fair allocation*, denoted by $a^{\text{L-MmF}}$, designates a max-min fair allocation for some routing such that $a^{\text{L-MmF}}\uparrow$ is maximum in lexicographic order over the sorted vectors of the max-min fair allocations for all routings:[4]

$$a^{\text{L-MmF}} = a_r^{\text{MmF}} \text{ for some } r \in R, \text{ and}$$
$$a^{\text{L-MmF}}\uparrow \geq a_r^{\text{MmF}}\uparrow \text{ for all } r \in R.$$

Importantly, the sorted vector of the max-min fair allocation in a macro-switch is greater than or equal to (in lexicographic order) that of a lex max-min fair allocation in the corresponding Clos network. This assertion follows from two facts: (1) if an allocation in a Clos network is feasible, then it is also feasible in its macro-switch (the easy proof is omitted); and (2) in a macro-switch, the sorted vector of the max-min fair allocation is greater than or equal to that of every feasible allocation. Since a lex max-min fair allocation in a Clos network is feasible, the conclusion follows.

Maximum throughput: The second objective, considered in §5, is to maximize the throughput of the max-min fair allocations in a Clos network.

Definition 2.5. A *throughput-max-min fair allocation*, denoted by $a^{\text{T-MmF}}$, designates a max-min fair allocation for some routing such that $t(a^{\text{T-MmF}})$ is maximum over the throughputs of the max-min fair allocations for all routings:

$$a^{\text{T-MmF}} = a_r^{\text{MmF}} \text{ for some } r \in R, \text{ and}$$
$$t(a^{\text{T-MmF}}) \geq t(a_r^{\text{MmF}}) \text{ for all } r \in R.$$

Denote $t(a^{\text{T-MmF}})$ by $T^{\text{T-MmF}}$. In contrast to the first objective, in §5 we show that the throughput of the max-min fair allocation in a macro-switch can be smaller than the throughput of a throughput-max-min fair allocation in the corresponding Clos network.

3 IMPOSSIBILITY OF MAXIMIZING THROUGHPUT ASSUMING SATISFIABLE DEMANDS

We show that the factor of throughput lost by the max-min fair allocation in a macro-switch relative to the maximum throughput across it (without max-min fair constraints) is at most $\frac{1}{2}$, and that this bound is tight.

Maximum throughput across a macro-switch: We make use of the well-known characterization of the maximum throughput across a macro-switch.

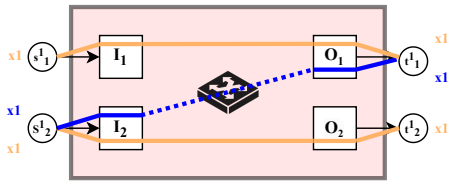
Definition 3.1. An allocation a is *maximum throughput* if: (1) a is feasible; and (2) $t(a)$ is maximum over the throughputs of all feasible allocations:

$$a \in A, \text{ and } t(a) \geq t(a') \text{ for all } a' \in A.$$

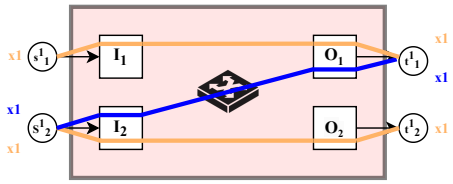
Let a^{MmF} the max-min fair allocation in a macro-switch MS_n, and a^{MT} be a maximum throughput allocation in MS_n. Denote $t(a^{\text{MmF}})$ by T^{MmF}, and, likewise, $t(a^{\text{MT}})$ by T^{MT}; we call T^{MT} the *maximum throughput across the macro-switch*. By the definition of maximum throughput, we write $T^{\text{MT}} \geq T^{\text{MmF}}$.

The lemma below specifies a maximum throughput allocation in terms of a bipartite maximum matching. The bipartite multigraph pertaining to a collection of flows in the macro-switch MS_n, denoted

[4]The notation for allocations reads as follows. In the superscript, prefixes 'L-' and 'T-' indicate the routing objective, and suffixes 'MmF' and 'MT' indicate whether or not max-min fair constraints are imposed at each routing. The absence of subscript denotes an optimal allocation for some routing objective, or a macro-switch allocation.



(a) The maximum throughput allocation in MS_1.



(b) The max-min fair allocation in MS_1.

Figure 2: Instantiation of the adversarial flows underlying Theorem 3.4 for n=1; in the proof, there are k type 2 (blue) flows.

by G^{MS}, is the graph whose sets of start and end nodes are, respectively, the sets of sources and destinations in the macro-switch, and whose collection of links is the collection of flows.

Lemma 3.2 (Folklore). *Consider a macro-switch MS_n, for an arbitrary integer $n \geq 1$. Let F' be some maximum matching in graph G^{MS}. For every collection F of flows in MS_n, there is a maximum throughput allocation a^{MT} such that:*

$$a^{MT}(f) = \begin{cases} 1, & \text{if } f \in F', \\ 0, & \text{otherwise,} \end{cases} \qquad \text{for all } f \in F.$$

Therefore, $T^{MT} = |F'|$.

Throughput decrease by introducing max-min fair rate constraints: The next example shows that the throughput of the max-min fair allocation for a given collection of flows in a macro-switch is smaller than the maximum throughput by a $\frac{3}{4}$-factor. The construction of the example leads to the tightness of the impossibility.

Example 3.3. Consider the collection of flows in the macro-switch for $n = 1$ illustrated in Figure 2. There are two types of flows, each identified with a different color in the figure:

- *Type 1 (orange)*: There is one flow (s_1^1, t_1^1) and one flow (s_2^1, t_2^1).
- *Type 2 (blue)*: There is one flow (s_2^1, t_1^1).

In Figure 2a, in the maximum throughput allocation, by Lemma 3.2, both type 1 flows are assigned rate 1, and the type 2 flow is assigned rate 0, thus leading to throughput 2. From the viewpoint of admission control, both type 1 flows are accepted, and transmitted at link capacity, while the type 2 flow is rejected.

In Figure 2b, in the max-min fair allocation, all flows are assigned rate $\frac{1}{2}$, such that $O_1 t_1^1$ is a bottleneck for the type 1 flow (s_1^1, t_1^1) and the type 2 flow, and $s_2^1 I_2$ is a bottleneck for the type 1 flow (s_2^1, t_2^1) and (again) the type 2 flow, thus leading to throughput $\frac{3}{2}$. From the viewpoint of congestion control, the type 2 flow claims a portion of the capacity of both $O_1 t_1^1$ and $s_2^1 I_2$; hence, it restricts the rates of both type 1 flows even though they are unrestricted elsewhere in the macro-switch, such that a $\frac{1}{4}$-fraction of the maximum throughput across the macro-switch is lost.

The forthcoming theorem establishes an upper bound of $\frac{1}{2}$ on the fraction of throughput lost by the max-min fair allocation in a macro-switch relative to the maximum throughput across it, and generalizes the above example to show that this bound is tight.

Theorem 3.4. *Consider a macro-switch MS_n, for an arbitrary integer $n \geq 1$. For every collection of flows in MS_n, it holds that $T^{MmF} \geq \frac{1}{2} T^{MT}$. Moreover, there is a collection of flows in MS_n such that $T^{MmF} \leq \frac{1}{2}(1+\epsilon) T^{MT}$, for arbitrary small ϵ.*

Proof. First, we show that for every collection F of flows in MS_n it holds that $T^{\text{MmF}} \geq \frac{1}{2} T^{\text{MT}}$. We make use of the characterizations of max-min fair and maximum throughput allocations provided by Lemmas 2.2 and 3.2, respectively. Denote by S_n and T_n the sets of sources and destinations in MS_n, respectively. For each source s, let τ_s be the total max-min fair rate over all flows $f \in F$ such that $s_f = s$ and, likewise, for each destination t, let τ_t be the total max-min fair rate over all flows $f \in F$ such that $t_f = t$:

$$\tau_s = \sum_{f \in F : s_f = s} a^{\text{MmF}}(f), \quad \text{and} \quad \tau_t = \sum_{f \in F : t_f = t} a^{\text{MmF}}(f).$$

We write

$$T^{\text{MmF}} = \sum_{s \in S_n} \tau_s = \sum_{t \in T_n} \tau_t.$$

We deduce two facts. Let F' be a maximum matching in G^{MS}. First, since for each source s there is at most one flow $f \in F'$ such that $s_f = s$, and, likewise, for each destination t there is at most one flow $f \in F'$ such that $t_f = t$, we deduce that

$$\sum_{s \in S_n} \tau_s \geq \sum_{f \in F'} \tau_{s_f}, \quad \text{and} \quad \sum_{t \in T_n} \tau_t \geq \sum_{f \in F'} \tau_{t_f}.$$

Let $I_{(s)}$ be the input switch of source s, and, likewise, let $O_{(t)}$ be the output switch of destination t. Second, from Lemma 2.2, we know that $s_f I_{(s_f)}$, or $O_{(t_f)} t_f$, are bottleneck links for f, for all flows $f \in F'$; hence, since the total rate over a bottleneck link is 1, we have $\tau_{s_f} + \tau_{t_f} \geq 1$, for all flows $f \in F'$. Summing over all flows $f \in F'$, we deduce that

$$\sum_{f \in F'} \left(\tau_{s_f} + \tau_{t_f} \right) \geq \left| F' \right|.$$

From the previous facts, we write

$$\begin{aligned} T^{\text{MmF}} &\geq \max\left(\sum_{f \in F'} \tau_{s_f}, \sum_{f \in F'} \tau_{t_f} \right) \\ &\geq \frac{1}{2} \sum_{f \in F'} \left(\tau_{s_f} + \tau_{t_f} \right) \\ &\geq \frac{1}{2} \left| F' \right|. \end{aligned}$$

Finally, from Lemma 3.2, we have $|F'| = T^{\text{MT}}$, to conclude that $T^{\text{MmF}} \geq \frac{1}{2} T^{\text{MT}}$.

Second, we show that there is an adversarial collection of flows in MS_n such that $T^{\text{MmF}} \leq \frac{1}{2}(1+\epsilon) T^{\text{MT}}$, for arbitrary small ϵ. The adversarial flows are obtained from those designed for Example 3.3, and illustrated in Figure 2a for $n = 1$, by replacing the type 2 flow with k type 2 flows between the same source-destination pair, for some integer $k \geq 1$. Therefore, there are two types of flows:

- *Type 1*: There is one flow (s_1^1, t_1^1) and one flow (s_2^1, t_2^1).
- *Type 2*: There are k flows (s_2^1, t_1^1).

In the maximum throughput allocation, both type 1 flows are assigned rate 1, and all k type 2 flows are assigned rate 0, such that $T^{\mathrm{MT}} = 2$. In the max-min fair allocation, all flows are assigned rate $\frac{1}{k+1}$, such that $T^{\mathrm{MmF}} = 1 + \frac{1}{k+1}$. We conclude that

$$T^{\mathrm{MmF}} \leq \frac{1}{2}(1+\epsilon)T^{\mathrm{MT}},$$

where $\epsilon = \frac{1}{k+1}$, which tends to 0 as k tends to infinity. □

4 IMPOSSIBILITY OF SATISFYING DEMANDS

We establish the impossibility of the lex-max-min fair allocation in a Clos network *closely* replicating the max-min fair allocation in its macro-switch. In favor of readability, we divide this result into two parts, the latter superseding the former. In §4.1, we show that there is a collection of flows for which the max-min fair allocation in a macro-switch is greater (in lexicographic order) than the lex-max-min fair allocation. In §4.2, we show that there is a collection of flows for which the lex-max-min fair rates of some flows are smaller than their max-min fair rates in the macro-switch by a $\frac{1}{n}$-factor, where n is the size of the network.

4.1 Lex-max-min fairness does not ensure the macro-switch abstraction

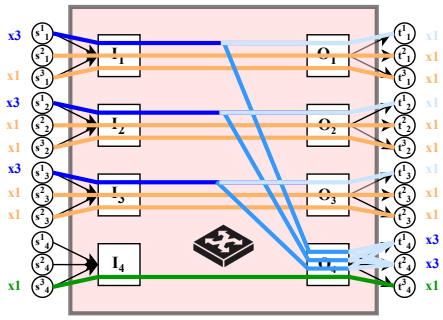
Given a collection of flows, we say that a feasible allocation in a macro-switch can be *replicated* in the corresponding Clos network if, assuming that each flow is offered to the data-center with its macro-switch rate, there is a *feasible routing* (i.e., a routing in which the capacities of all links are satisfied). The next example shows that the max-min fair allocation of a given collection of flows in a macro-switch cannot be replicated in the corresponding Clos network; thus, it shows that, for all max-min fair allocations in the network, the rate of some flow is smaller than its macro-switch rate, implying that the max-min fair allocation in the macro-switch is greater than a lex max-min fair allocation. The construction of the example leads to the first part of the impossibility; a slight modification, and a more nuanced argument, leads to the second part.

Example 4.1. Consider the collection of flows in the macro-switch for $n = 3$ illustrated in Figure 3. (The figure shows only the first four input and the first four output switches.) There are three types of flows, each identified with a different color:

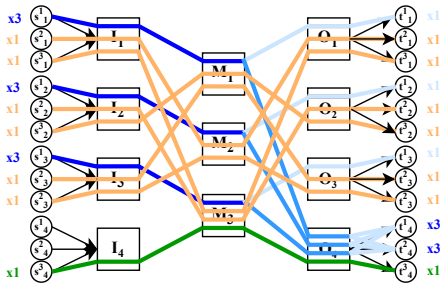
- *Type 1 flows (orange):* There is one flow (s_i^j, t_i^j), $i \in [n]$ and $j \in [2, n]$.
- *Type 2.a flows (blue):* There is one flow (s_i^1, t_i^1), $i \in [n]$.
- *Type 2.b flows (blue):* There is one flow (s_i^1, t_{n+1}^j), $i \in [n]$ and $j \in [n-1]$.
- *Type 3 flow (green):* There is one flow (s_{n+1}^n, t_{n+1}^n).

Type 2.b flows are placed such that there are three flows entering each of the first two destinations of O_4. In Figure 3a, in the max-min fair allocation in the macro-switch, all type 1 and the type 3 flows are assigned rate 1, and all type 2 flows are assigned rate $\frac{1}{3}$.

In Figure 3b, assume that each flow is offered to the data-center with its macro-switch max-min fair rate. We show that there is no feasible routing. Let the *available capacity* of a link be the difference



(a) The macro-switch MS_3



(b) The Clos network C_3.

Figure 3: Instantiation of the adversarial flows underlying Theorems 4.2 and 4.3 for n = 3. While in the proof of Theorem 4.2 there is one type 1 (orange) flow between each corresponding source-destination pair, in that of Theorem 4.3 there are $n+1$ type 1 flows.

between the link unit capacity and the total rate over all flows that have already been assigned a path traversing that link.

We derive two conditions that all possibly feasible routings met. (1) Since the rate of each type 1 flow is 1, the type 1 flows leaving I_i are assigned to different middle switches; thus, we deduce that all type 2 flows leaving I_i are assigned to the same middle switch, for all $i \in [3]$. (2) Since the total rate over all type 2 flows leaving I_i and entering O_4 is $\frac{2}{3}$, and, by the first condition, all type 2 flows leaving I_i are assigned to the same middle switch, we deduce that the set of type 2 flows leaving I_i is assigned to a different middle switch than the set of type 2 flows leaving $I_{i'}$, for all $i, i' \in [3]$, $i \neq i'$.

Therefore, without loss of generality, we assume that the type 1 flow (s_i^j, t_i^j) is assigned to M_{k+1}, where $k = i + j - 2 \pmod 3$, $j \in [2, 3]$, and that the set of type 2 flows leaving I_i is assigned to M_i, $i \in [3]$. (See Figure 3b.) Finally, since the available capacity at $M_m O_4$ is $\frac{1}{3}$, and the rate of the type 3 flow is 1, we conclude that the assignment of the type 3 flow to M_m violates the capacity of $M_m O_4$, for all $m \in [3]$, which is what we wanted to show.

The theorem below generalizes the example above to a Clos network of arbitrary large size.

Theorem 4.2. *Consider a Clos network C_n, for an arbitrary integer $n \geq 3$. There is a collection of flows in C_n such that $a^{MmF}\uparrow \succ a^{L\text{-}MmF}\uparrow$.*

The proof of the theorem corresponds exactly to the construction and argument of the example by interpreting n as the size of the Clos network, and is not repeated. We remark that this part of the impossibility is not entirely new: prior work on multi-rate Clos networks showed that, for every Clos network, there is a collection

of flows for which some feasible macro-switch rates, not necessarily max-min fair, cannot be replicated in the network [11]; in contrast, the next part of the impossibility is fundamentally new.

4.2 Lex-max-min fairness leads to starvation

Since both a lex-max-min fair allocation in a Clos network and the max-min fair allocation in its macro-switch optimize allocations in lexicographic order, it would be expected that the lex-max-min fair rate of each flow was less than its macro-switch rate by at most some constant factor. The next theorem refutes this expectation.

Theorem 4.3. *Consider a Clos network C_n, for an arbitrary integer $n \geq 3$. There is a collection F of flows in C_n such that $a^{L\text{-}MmF}(f) = \frac{1}{n} a^{MmF}(f)$, for some flow $f \in F$.*

Proof details of Theorem 4.3: We assume a Clos network C_n of size n, for some integer $n \geq 3$. We show that there is an adversarial collection of flows in C_n such that the max-min fair rate in MS_n of a specific flow is 1, whereas its lex-max-min fair rate in C_n is $\frac{1}{n}$.

We describe a collection of flows that we show meets the requirements. The adversarial flows are obtained from those designed for Theorem 4.2, and illustrated in Figure 3a for $n = 3$, by replacing each type 1 flow between every source-destination pair with $n+1$ type 1 flows between that pair. Thus, there are three types of flows:

- *Type 1 flows:* There are n+1 flow (s_i^j, t_i^j), $i \in [n]$ and $j \in [2, n]$.
- *Type 2.a flows:* There is one flow (s_i^1, t_i^1), $i \in [n]$.
- *Type 2.b flows:* There is one flow (s_i^1, t_{n+1}^j), $i \in [n]$ and $j \in [n-1]$.
- *Type 3 flow:* There is one flow (s_{n+1}^n, t_{n+1}^n).

The key idea behind the adversarial flows is that, given that not all flows can uphold their macro-switch rates, lex max-min fairness prioritizes upholding the lower macro-switch rates of type 1 and type 2 flows over the higher macro-switch rate of the type 3 flow. In detail, if we consider only type 1 and type 2 flows, then their network rates are upper bounded in lexicographic order by their macro-switch rates; thus, if the macro-switch rates of type 1 and type 2 flows can be upheld, then, provided that the network rate of the type 3 flow exceeds them, their lex max-min fair rates match their macro-switch rates. However, similarly to the proof of Theorem 4.2, the macro-switch rates of type 1 and type 2 flows are *incompatible*, in the sense that, for each input switch, upholding their rates implies no type 1 sharing a common link with a type 2 flow. Therefore, all type 2 flows leaving that input switch are assigned to the same middle switch, with a different middle switch for each input switch; thus, the type 2 flows entering O_{n+1} are evenly divided over all middle switches. Consequently, the available capacity at each link entering O_{n+1} implies that the rate of the type 3 flow decreases to the rate of the type 2 flows.

We evaluate the max-min fair allocation in MS_n, and a lex max-min fair allocation in C_n, of the adversarial flows.

Lemma 4.4. *In the max-min fair allocation in MS_n for the adversarial flows: (1) all type 1 flows are assigned rate $\frac{1}{n+1}$; (2) all type 2 flows are assigned rate $\frac{1}{n}$; and (3) the type 3 flow is assigned rate* 1.

The proof of the lemma follows from the routine application of the bottleneck property to the posited max-min fair allocation, and is not presented. Intuitively, for each type of flow, a flow of that type does not share links with flows of other types, so that their rates are independent. Therefore, since each type 1 flow shares both its link with n other type 1 flows, it is assigned rate $\frac{1}{n+1}$; since each type 2 flow shares both its link with $n-1$ other type 2 flows, it is assigned rate $\frac{1}{n}$; and since the type 3 flow does not share its links with other flows, it is assigned rate 1.

The evaluation of a lex max-min fair allocation requires the claim below, which clarifies the *incompatibility* between the macro-switch rates of type 1 and type 2 flows.

Claim 4.5. *Consider the collection of flows in MS_n consisting only of the adversarial flows of type 1 and 2. Assume that type 1 and type 2 flows are offered to the data-center with rates $\frac{1}{n+1}$ and $\frac{1}{n}$, respectively. For all feasible routings, the following conditions hold:*

(1) The number of type 1 and type 2 flows that leave I_i and are assigned to M_m is, respectively, 0 *and* n*, or* n+1 *and* 0*, for all $i, m \in [n]$; the assignment of type 1 flows to middle switches is independent of the sources of these flows.*
(2) The number of type 2.b flows that are assigned to M_m is $n-1$, for all $m \in [n]$.

Proof. First, we show that the first condition is met. Let x_i^m and y_i^m be, respectively, the number of type 1 and 2 flows that leave I_i and are assigned to M_m, for $i, m \in [n]$. Since the total rate over all type 1 and type 2 flows that leave I_i is n, and the capacity of $I_i M_m$ is 1, we write

$$\frac{x_i^m}{n+1} + \frac{y_i^m}{n} = 1 \Leftrightarrow x_i^m = \frac{(n - y_i^m)(n+1)}{n}, \tag{1}$$

where $x_i^m = [0, n+1]$ and $y_i^m = [0, n]$, to deduce that x_i^m is an integer exactly in case $(n - y_i^m)(n+1)$ is divisible by n. From the fact that the least common multiple between n and $n+1$ is $n(n+1)$ we conclude that the only pairs of integer solutions to Equation 1 are $(x_i^m, y_i^m) = (0, n)$ and $(x_i^m, y_i^m) = (n+1, 0)$.

Second, we show that violating the second constraint implies violating the capacity of some link, thereby concluding that the second condition is also met. From the first constraint, the number of type 2.b flows that leave I_i, and are assigned to M_m, is either 0 or $n-1$, for all $i, m \in [n]$. Suppose that there are $i, i' \in [n]$, $i \neq i'$, such that the number of type 2.b flows that leave I_i and are assigned to M_m is $n-1$, and that the number of type 2.b flows that leave $I_{i'}$ and are assigned to M_m is also $n-1$, for some $m \in [n]$. Then, the total rate over all flows that traverse $M_m O_{n+1}$ is at least $2(1 - \frac{1}{n})$, which violates the capacity of $M_m O_{n+1}$. □

Lemma 4.6. *In a lex max-min fair allocation in C_n of the adversarial flows: (1) all type 1 flows are assigned rate $\frac{1}{n+1}$; (2) all type 2 flows are assigned rate $\frac{1}{n}$; and (3) the type 3 flow is assigned rate $\frac{1}{n}$.*

Proof. The proof is composed of two steps. The first step shows that the posited lex-max-min fair allocation is the max-min fair allocation for some routing; the second step shows that it is maximum in lexicographic order over all possible routings. Let a^* be the posited lex max-min fair allocation in C_n.

Step 1: *There is a routing $r \in R$ such that $a^* = a_r^{MmF}$.* We claim that the routing described below, and illustrated in Figure 3b for $n = 3$, meets the requirements:

- All $n+1$ type 1 flows (s_i^j, t_i^j) are assigned to M_{k+1}, where $k = i + j - 2 \pmod n$, $i \in [n]$ and $j \in [2, n]$.
- The type 2.a flow (s_i^1, t_i^1) is assigned to M_i, $i \in [n]$.
- The type 2.b flow (s_i^1, t_i^j) is assigned to M_i, $i \in [n]$ and $j \in [n-1]$.
- The type 3 flow (s_{n+1}^n, t_{n+1}^n) is assigned to M_n.

The proof of the claim follows again from the routine application of the bottleneck property to the posited lex max-min fair allocation for this routing, and is also not presented. Intuitively, since all type 1 and type 2 flows share their links inside the network with no more flows than they share their links outside the network, their bottleneck links in the macro-switch are also bottlenecks links in the Clos network, such that their network rates match their macro-switch rates. In contrast, since the type 3 flow shares M_nO_{n+1} with $n-1$ type 2 flows, its bottleneck link in the Clos network is M_nO_{n+1}, such that its network rate decreases to $\frac{1}{n}$.

Step 2: *For all routings $r \in R$, we have $a^*\uparrow \geq a_r^{MmF}\uparrow$.* We show that for all routings r that increase the max-min fair rate of a type 1 flow above $\frac{1}{n+1}$, or increase the max-min fair rate of a type 2 flow above $\frac{1}{n}$, or increase the max-min fair rate of a type 3 flow above $\frac{1}{n}$, we have $a_*\uparrow \geq a_r^{\text{MmF}}\uparrow$. We divide the proof into two cases.

Case 1: We show that if a routing r increases the rate of a type flow above $\frac{1}{n+1}$, then it decreases the rate of another type 1 flow below $\frac{1}{n+1}$, and that, likewise, if a routing r increases the rate of a type 2 flow above $\frac{1}{n}$, then it decreases the rate of another type 2 flow below $\frac{1}{n}$. We conclude that $a^*\uparrow \succ a_r^{\text{MmF}}\uparrow$, since each increase is obtained at the cost of decreasing a flow of equal rate.

A first property of the adversarial flows is that each type 1 flow leaving s_i^j shares $s_i^j I_i$ with n other type 1 flows leaving s_i^j, for all $i \in [n]$ and $j \in [2, n]$, and that, likewise, each type 2 flow leaving s_i^1 shares $s_i^1 I_i$ with $n-1$ other type 2 flows leaving s_i^1, for all $i \in [n]$. Thus, if the rate of a type 1 flow leaving s_i^j is greater than $\frac{1}{n+1}$, then the capacity of $s_i^j I_i$ implies that the rate of another type 1 flow leaving s_i^j is smaller than $\frac{1}{n+1}$; likewise, if the rate of a type 2 flow leaving s_i^1 is greater than $\frac{1}{n}$, then the capacity of $s_i^1 I_i$ implies that the rate of another type 2 flow leaving s_i^1 is smaller than $\frac{1}{n}$.

Case 2: We show that if a routing r increases the rate of a type 3 flow above $\frac{1}{n}$, then it decreases the rate of a type 1 flow below $\frac{1}{n+1}$, or decreases the rate of a type 2 flow below $\frac{1}{n}$. We conclude that $a^*\uparrow \succ a_r^{\text{MmF}}\uparrow$, since the increase is obtained at the cost of decreasing flows of no greater rate.

A second property of the adversarial flows is that, if the rates of type 1 and 2 flows are, respectively, $\frac{1}{n+1}$ and $\frac{1}{n}$, then, ommiting of the rate of the type 3 flow, the total rate over all type 2.b flows that traverse M_mO_{n+1} is $1 - \frac{1}{n}$, for all $m \in [n]$; this property follows from Claim 4.5. Thus, if the rate of the type 3 flow is greater than $\frac{1}{n}$, then there is a type 1 flow with rate different from $\frac{1}{n+1}$, or a type 2 with rate different from $\frac{1}{n}$. In the first event, we conclude that the rate of this type 1 flow is smaller than $\frac{1}{n+1}$, or Case 1 implies there is another type 1 flow with rate smaller than $\frac{1}{n+1}$. Likewise, in the second event, we conclude that the rate of this type 2 flow is smaller $\frac{1}{n}$, or Case 1 again implies there is another type 2 flow with rate smaller than $\frac{1}{n}$. □

Proof of Theorem 4.3. We show that the adversarial collection of flows in C_n meets the requirements. From Lemma 4.4, the max-min fair rate in MS_n of the type 3 flow is 1, and, from Lemma 4.6, the lex max-min fair rate in C_n of the type 3 flow is $\frac{1}{n}$, yielding a $\frac{1}{n}$-factor decrease of its rate. □

5 INCONGRUENCE BETWEEN MAXIMIZING THROUGHPUT AND SATISFYING DEMANDS

We show that the factor of throughput gained by a throughput-max-min fair allocation in a Clos network relative to the max-min fair allocation in its macro-switch is at most 2, and that this bound becomes tight as the network size increases, with tightness achieved at the cost of canceling out the rates of most flows.

Maximum throughput across a Clos network: We build upon the well-known characterization of maximum throughput across a Clos network (without max-min fair constraints at each routing). Let a_r^{MT} be a maximum throughput allocation for a routing $r \in R$.

Definition 5.1. A *throughput-maximum throughput allocation*, denoted by $a^{\text{T-MT}}$, designates a maximum throughput allocation for some routing such that $t(a^{\text{T-MT}})$ is maximum over the throughputs of a maximum throughput allocation for all routings:

$$a^{\text{T-MT}} = a_r^{\text{MT}} \text{ for some } r \in R, \text{ and}$$
$$t(a^{\text{T-MT}}) \geq t(a_r^{\text{MT}}) \text{ for all } r \in R.$$

Denote $t(a^{\text{T-MT}})$ by $T^{\text{T-MT}}$; we call $T^{\text{T-MT}}$ the *maximum throughput across the Clos network*. Since every feasible allocation in a Clos network is also feasible in its macro-switch, we write $T^{\text{MT}} \geq T^{\text{T-MT}}$. The lemma below establishes the equivalence between the maximum throughput across a macro-switch and across the corresponding Clos network, meaning that some maximum throughput allocations in the macro-switch can be replicated in the network.

Lemma 5.2 (Folklore). *Consider a Clos network C_n, for an arbitrary integer $n \geq 1$. For every collection of flows in C_n, there is a maximum throughput allocation a^{MT} in MS_n and a throughput-maximum throughput allocation $a^{T\text{-}MT}$ in C_n such that $a^{MT} = a^{T\text{-}MT}$. Therefore, $T^{MT} = T^{T\text{-}MT}$.*

The bipartite multigraph pertaining to a collection of flows in the Clos network C_n, denoted by G^C, is the graph whose sets of start and end nodes are, respectively, the sets of input and output switches in the network, and whose collection of links is the collection of flows identified by their input-output switch pairs. The proof of the lemma introduces the correspondence between a n-link coloring in G^C and a link-disjoint routing in C_n.[5]

Throughput increase by sacrificing max-min fair rate constraints: The heart of this section is a simple new algorithm, called *Doom-Switch* algorithm and formalized in Algorithm 1, for approximating a throughput-max-min fair allocation in a Clos network. The algorithm executes two steps: (1) It computes a sub-collection of

[5]We recall that *Konig's link coloring theorem* [23] states that if the maximum degree over all nodes of a bipartite multigraph is n, then the graph has an n-link coloring. If the bipartite multigraph pertaining to a collection of flows in a Clos network has degree at most the number n of middle switches, then Konig's link coloring theorem establishes a correspondence between an n-link coloring in the graph and a link-disjoint routing in the network, which equates to associating each of the n colors with each of the n middle switches, and assigning each flow to the corresponding middle switch.

the flows that induces maximum throughput if assigned rate 1, and a link-disjoint routing for these flows, and assigns them accordingly (lines 1-2). (2) It assigns all remaining flows to a common middle switch (line 3). Thus, it perverts the max-min fair constraints by decreasing the rates of the latter flows relative to the macro-switch to the increase of the former flows.

Algorithm 1 An algorithm that given a collection F of flows in a Clos network C_n finds a routing for which the max-min fair allocation approximates a throughput-max-min fair allocation in C_n.

1: Compute a maximum matching $F' \subseteq F$ in G^{MS}.
2: Compute a n-coloring of G^C pertaining to F'. Let $F'_m \subseteq F'$ be the sub-collection of flows colored m, $m \in [n]$. Assign all flows in F'_m to middle switch M_m.
3: Let m' be a color for which $|F_m|$ is minimum over all $m \in [n]$. Assign all flows in $F \setminus F'$ to middle switch $M_{m'}$.

The next example applies the Doom-Switch algorithm to a given collection of flows in a Clos network, and shows that the throughput of the max-min fair allocation in the network yield by the algorithm can exceed that of the max-min fair allocation in its macro-switch by coercing some flows into decreasing their rates, thus allowing other flows to increase their rates to overall throughput gain.

Example 5.3. Consider the collection of flows in the Clos network for $n = 7$ illustrated in Figure 4, and obtained by stacking three copies of the construction introduced in Example 3.3 (upon re-indexing the flows such that they leaving a common input switch and enter a common output switch). (The figure shows only the first input and the first output switches.) Therefore, there are two types of flows, each identified with a different color:

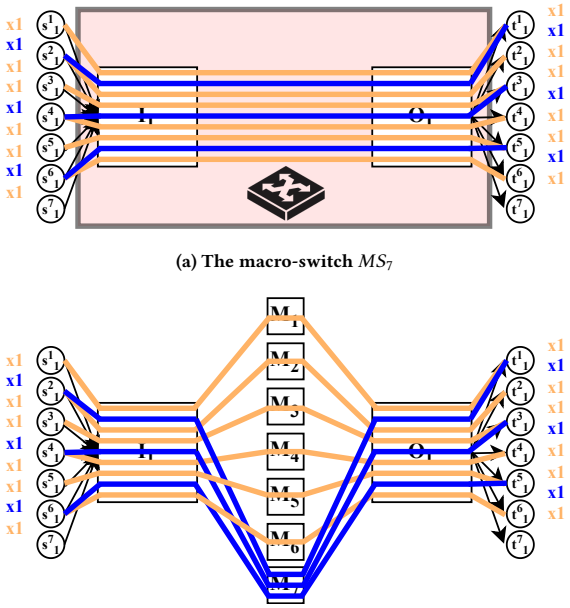
- *Type 1 (orange)*: There is one flow (s_1^j, t_1^j), $j \in [6]$.
- *Type 2 (blue)*: There is one flow (s_1^j, t_1^{j-1}), $j \in [6]$ with j is even.

In Figure 4a, in the max-min fair allocation in the macro-switch, all flows are assigned rate $\frac{1}{2}$, thus leading to throughput $\frac{9}{2}$. In Figure 4b, we apply the Doom-Switch algorithm. A maximum matching in the bipartite multigraph pertaining to the macro-switch consists of all type 1 flows, and the algorithm, for instance, assigns type 1 flow (s_1^j, t_1^j) to M_j, $j \in [6]$; the algorithm assigns all type 2 flows to M_7. In a max-min fair allocation for this routing, all type 2 flow transfer their bottlenecks to I_1M_7 and M_7O_1, thus decreasing their rates from $\frac{1}{2}$ to $\frac{1}{3}$. Hence, all type 1 flows keep the same bottleneck links that in the macro-switch, and increase their rates from $\frac{1}{2}$ to $\frac{2}{3}$. Thus, the throughput increases from $\frac{9}{2}$ to 5.

The theorem below establishes an upper bound of 2 on the factor of throughput gained by a throughput-max-min fair allocation relative to the max-min fair allocation in a macro-switch, and generalizes the example above to show that this bound becomes tight as the network size grows large.

Theorem 5.4. *Consider a Clos network C_n, for an arbitrary integer $n \geq 3$. For every collection of flows in C_n, it holds that $T^{\text{T-MmF}} \leq 2T^{\text{MmF}}$. Moreover, there is a collection of flows in C_n such that $T^{\text{T-MmF}} \geq 2(1-\epsilon)T^{\text{MmF}}$, for ϵ arbitrary close from above to $\frac{1}{n-1}$.*

Proof. First, we show that for every collection of flows in C_n it holds that $T^{\text{T-MmF}} \leq 2T^{\text{MmF}}$. We make use of the characterization



(a) The macro-switch MS_7

(b) The Clos network C_7.

Figure 4: Instantiation of the adversarial flows underlying Theorem 5.4 for n = 7; in the proof, there are $\frac{n-1}{2}$ copies of the construction illustrated in Figure 2, and k type 2 (blue) flows in each gadget.

of maximum throughput across a Clos network, and of the relationship between the max-min fair allocation in a macro-switch and the maximum throughput across it, provided by Lemma 5.2 and Theorem 3.4, respectively. We write:

$$\begin{aligned} T^{\text{T-MmF}} &\leq T^{\text{T-MT}} && \text{(by definition)} \\ &= T^{\text{MT}} && \text{(from Lemma 5.2)} \\ &\leq 2T^{\text{MmF}} && \text{(from Theorem 3.4).} \end{aligned}$$

Second, we show that there is an adversarial collection of flows in C_n such that $T^{\text{T-MmF}} \geq 2(1-\epsilon)T^{\text{MmF}}$, for ϵ arbitrary close from above to $\frac{1}{n-1}$. Without loss of generality, we assume that n is odd. The adversarial flows generalize those designed for Example 5.3 by stacking $\frac{n-1}{2}$ copies of the construction introduced in Example 3.3, and replacing each type 2 flow with k type 2 flows between the same source-destination pair. Thus, there are two types of flows:

- *Type 1*: There is one flow (s_1^j, t_1^j), $j \in [n-1]$.
- *Type 2*: There are k flow (s_1^j, t_1^{j-1}), $j \in [n-1]$ with j even.

In the max-min fair allocation in MS_n, all $\frac{n-1}{2}(k+2)$ flows have rate $\frac{1}{k+1}$, such that $T^{\text{MmF}} = \frac{n-1}{2}\left(1 + \frac{1}{k+1}\right)$. In the max-min fair allocation in C_n yield by the Doom-Switch algorithm, all $n-1$ type 1 flows have rate $1 - \frac{2}{n-1}$, and all $\frac{n-1}{2}k$ type 2 flows have rate $\frac{2}{k(n-1)}$, such that $T^{\text{T-MmF}} \geq n-2$. We conclude that

$$T^{\text{T-MmF}} \geq 2(1-\epsilon)T^{\text{MmF}},$$

where $\epsilon = \frac{k+n}{(n-1)(k+2)}$, which tends to $\frac{1}{n-1}$ as k tends to infinity. □

6 RELATED WORK

Multirate rearrangeability of Clos networks: The multirate rearrangeability of Clos networks has been studied by both networking and theory communities [11, 15, 21, 25, 27]. The setting consists of a Clos network with a fixed number n of servers per ToR switch, a fixed number of ToR switches, and a *variable* number m of middle switches; and a feasible allocation in its macro-switch. The goal is to find a routing that replicates the feasible allocation in a Clos network while minimizing the number of middle switches used. It is attained by combinations of first-fit heuristics with Konig's theorem [15, 21]. A Clos network is *multirate rearrangeable* if all feasible allocations in its macro-switch can be replicated [25]. A popular conjecture states that a Clos network is multirate rearrangeable if and only if $m \geq 2n - 1$ [11], with the best known lower [27] and upper [21] bounds given by $\lceil \frac{5n}{4} \rceil$ and $\lceil \frac{20}{9} n \rceil$, respectively.

The setting and goals of this work are different. We investigate the lex-max-min fairness and throughput provided by Clos networks of fixed parameter relative to those idealized by their macro-switches in the presence of max-min fair constraints at each routing, not the sizing of Clos networks of variable parameter to replicate their macro-switches in the absence of these constraints.

Routing subject to max-min fair constraints: The problem of finding a lex-max-min fair allocation in an arbitrary network for flows having a common source and possibly different destinations was introduced in [22]. The same work also shows that this problem is NP-complete, and designs an approximation algorithm which returns a feasible allocation that is *approximately* max-min fair, and assigns each flow at least a $\frac{1}{2}$-fraction of its lex max-min fair rate.

The setting and goals of this work are also different. We consider Clos networks, rather than arbitrary networks, and flows having possibly different sources and different destinations, with flow rates *exactly* max-min fair; these premises rule out the application of the algorithm from Reference [22] to our setting. We analyze the closeness between lex-max-min fair allocations in Clos networks and max-min fair allocation in their macro-switches, rather than approximating lex-max-min fair allocations.

Data-center routing algorithms: The long-standing data-center routing algorithm is ECMP (Equal Cost Multi-Path), which assigns flows to source-destination paths chosen uniformly at random [2]. State-of-the-art algorithms assume that flows are offered to the data-center with their macro-switch rates, and their goal is to minimize maximum link congestion, where the *congestion* of a link is the ratio between the total macro-switch rate over all flows traversing the link and the link capacity [3]. It is attained by greedy [3, 4, 18], and local-search algorithms [3, 9] that assign flows to source-destination paths based on path congestion, where the congestion of a path is the maximum over the congestion of its links.

The simulation-based evaluation presented in the extended version of this paper shows that, for stochastic inputs, algorithms that first calculate the macro-switch rates, and then borrow these rates to assign flows based on path congestion, can approximate well the macro-switch rates. However, for worst-case inputs, it can be shown there are adversarial flows for which the Clos network max-min fair rates of some flows are arbitrarily smaller than their max-min macro-switch rates.

7 CONCLUSIONS AND DISCUSSION

We proved three fundamental performance bounds characterizing the discrepancy between a Clos network subject to unsplittable flows and max-min fair constraints at each routing and the corresponding macro-switch with respect to throughput and fairness.

(R1): We showed that, for every interconnection network connecting sources to destinations (not necessarily a Clos network), the imposition of max-min fair constraints up to halves the maximum throughput even if we assume that arbitrary macro-switch rates can be replicated in the network. This suggests that, if data-center performance is measured in terms of flow completion times, then it may benefit from avoiding max-min fair constraints. A mechanism to circumvent these constraints is *scheduling*: by delaying the transmission of some flows, the remaining flows can be transmitted at link capacity (analogously to admission control). With scheduling, the average throughput across the network over time may increase such that the average flow completion times may decrease relative to those obtained in the presence of max-min fair constraints.

(R2): We showed that the unsplittability of flows implies that a lex-max-min fair allocation prioritized upholding the rates of flows with lower macro-switch rate at the cost of decreasing the rates of flows with higher macro-switch rate; thus, we showed that there can be a large discrepancy between the network rates of the latter flows and their macro-switch rates. This reveals that, if data-centers wish to provide a macro-switch abstraction for max-min fairness, then lex-max-min fairness may not be the right routing objective. An alternative objective is *relative-max-min fairness*, which aims at ensuring that the network rate of each flow is at least some constant fraction of its macro-switch rate. We believe that showing whether or not relative-max-min fairness can closely implement this abstraction, and, if so, designing approximation algorithms for this objective, is an important open question.

(R3): We showed that routing can pervert the enforcement of max-min fair constraints at each routing to up to double the throughput over the macro-switch at the cost of nullifying the rates of some flows. This further supports the assertion that max-min fair constraints at each routing alone cannot ensure a macro-switch abstraction for max-min fairness, and, moreover, calls into question the use of throughput as *the* metric for data-center performance.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful feedback. We are grateful to Adithya Philip, Anup Agarwal, Hugo Sadok, Inês Ferreira, Margarida Ferreira, and Ricardo Santos for their meticulous reviews, and to Misha Lavrov for his help implementing scalable bipartite matching. This work was partially supported by the Portuguese Foundation for Science and Technology under the grant SFRH/BD/151468/2021 (through the Carnegie Mellon Portugal Program) and the project UIDB/50008/2020, a VMWare Systems Research Award, NSF Award 2007733, a Cylab Presidential Fellowship, and a Google Research Gift.

REFERENCES

[1] Saksham Agarwal, Shijin Rajakrishnan, Akshay Narayan, Rachit Agarwal, David Shmoys, and Amin Vahdat. 2018. Sincronia: Near-optimal Network Design for Coflows. In *Proceedings of the ACM SIGCOMM Conference*. 16–29.

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

[3] Mohammad Al-Fares, Sivasankar Radhakrishnan, Barath Raghavan, Nelson Huang, and Amin Vahdat. 2010. Hedera: Dynamic Flow Scheduling for Data Center Networks. In *Proceedings of the USENIX Conference on Networked Systems Design and Implementation*, Vol. 10. 89–92.

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

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

[6] Dimitri Bertsekas and Robert Gallager. 1992. *Data Networks, 2nd Edition*. Prentice-Hall, Inc.

[7] Dimitris Bertsimas, Vivek F. Farias, and Nikolaos Trichakis. 2011. The Price of Fairness. *Operations Research* 59, 1 (2011), 17–31.

[8] Qizhe Cai, Mina Tahmasbi Arashloo, and Rachit Agarwal. 2022. dcPIM: Near-Optimal Proactive Datacenter Transport. In *Proceedings of the ACM SIGCOMM Conference*. 53–65.

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

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

[11] Shun-Ping Chung and Keith W. Ross. 1991. On Nonblocking Multirate Interconnection Networks. *SIAM J. Comput.* 20, 4 (1991), 726–736.

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

[13] Brian Davey and Hilary Priestley. 2002. *Introduction to Lattices and Order, 2nd Edition*. Cambridge University Press.

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

[15] Uriel Feige and Mohit Singh. 2008. Edge Coloring and Decompositions of Weighted Graphs. In *Proceedings of the European Symposium on Algorithms*. 405–416.

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

[17] Tobias Harks, Martin Hoefer, Kevin Schewior, Alexander Skopalik, Tobias Harks, Martin Hoefer, Kevin Schewior, and Alexander Skopalik. 2016. Routing Games With Progressive Filling. *IEEE/ACM Transactions on Networking* 24, 4 (2016).

[18] Kuo-Feng Hsu, Ryan Beckett, Ang Chen, Jennifer Rexford, and David Walker. 2020. Contra: A Programmable System for Performance-Aware Routing. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation*. 701–721.

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

[20] Sangeetha Abdu Jyothi, Ankit Singla, P. Brighten Godfrey, and Alexandra Kolla. 2016. Measuring and Understanding Throughput of Network Topologies. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 761–772.

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

[22] Jon Kleinberg, Eva Tardos, and Yuval Rabani. 1999. Fairness in Routing and Load Balancing. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*. 568–578.

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

[24] Laurent Massoulie and James Roberts. 2002. Bandwidth Sharing: Objectives and Algorithms. *IEEE/ACM Transactions on Networking* 10, 3 (2002), 320–328.

[25] Riccardo Melen and Jonathan S Turner. 1989. Nonblocking Multirate Networks. *SIAM J. Comput.* 18, 2 (1989), 301–313.

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

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

[28] Bozidar Radunovic and Jean-Yves Le Boudec. 2007. A Unified Framework for Max-Min and Min-Max Fairness With Applications. *IEEE/ACM Transactions on Networking* 15, 5 (2007), 1073–1083.

[29] Arjun Roy, Hongyi Zeng, Jasmeet Bagga, George Porter, and Alex Snoeren. 2015. Inside the Social Network's (Datacenter) Network. *ACM SIGCOMM Computer Communication Review* 45, 4 (2015), 123–137.

[30] Arjun Singh, Joon Ong, Amit Agarwal, Glen Anderson, Ashby Armistead, Roy Bannon, Seb Boving, Gaurav Desai, Bob Felderman, Paulie Germano, Anand Kanagala, Jeff Provost, Jason Simmons, Eiichi Tanda, Jim Wanderer, Urs Hölzle, Stephen Stuart, and Amin Vahdat. 2015. Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network. *ACM SIGCOMM Computer Communication Review* 45, 4 (2015), 183–197.

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

[32] Asaf Valadarsky, Gal Shahaf, Michael Dinitz, and Michael Schapira. 2016. Xpander: Towards Optimal-Performance Datacenters. In *Proceedings of the International on Conference on Emerging Networking EXperiments and Technologies*. 205–219.