JOURNAL OF
Optical Communications
and Networking

Nonblocking conditions for Clos fabrics with non-uniform switch radixes

Takeru Inoue,[1,*] Toru Mano,[1] Kazuya Anazawa,[1] and Takeaki Uno[2]

[1]NTT Network Innovation Laboratories, NTT Corporation, Kanagawa, Japan
[2]National Institute of Informatics, Tokyo, Japan
*tkr.inoue@ntt.com

Received 29 August 2024; revised 28 October 2024; accepted 14 November 2024; published 17 December 2024

Datacenter networks (DCNs) evolve over years and so comprise switches from different generations. Thus, each stage/layer of the Clos fabric may consist of switches with varying radixes (i.e., different port counts), leading to *non-uniform* stages. While optical circuit switches are increasingly deployed in DCNs to enhance transmission capacity and energy efficiency, the nonblocking condition, crucial for determining the performance of circuit-switched networks, has been established only for Clos fabrics with uniform stages. This study extends the nonblocking condition to Clos fabrics with non-uniform stages. To facilitate practicality, we formulate the condition using integer linear programming (ILP). Using our novel, to our knowledge, condition, we quantitatively demonstrate how much the nonblocking property is compromised under two practical scenarios, random link failures and network expansion, which would break network uniformity. In particular, we reveal that network expansion, common in DCN evolution, could significantly undermine the nonblocking property. Additionally, we assess the computational efficiency of our ILP formulation, which can successfully evaluate the nonblocking property of a large Clos fabric accommodating 32K terminals/uplinks in just 19 min.

https://doi.org/10.1364/JOCN.540792

1. INTRODUCTION

The traffic within datacenter networks (DCNs) continues to experience rapid growth [1,2]. With the slowdown in Moore's law [3], traditional electrical packet switches (EPSs) are unable to support the expected traffic demands. Consequently, DCNs are beginning to migrate to optical circuit switches (OCSs) for their data-rate transparency and great energy efficiency; Google has announced the integration of OCSs in the spine stage of their DCNs [4,5]. To further improve the transmission capacity and energy efficiency, more stages of the Clos fabric will adopt OCSs in the near future DCNs [6–11].

Since DCNs evolve over years in response to advances in scale and technology, they are often composed of switches from different generations. Google has highlighted that this non-uniformity is a key challenge in managing DCNs as networks must handle switches of different radixes (i.e., different port counts) [12]; [13] clearly illustrates how DCNs have evolved with switches of different sizes. Meta, in a paper on their DCN migration plan [14], stated that they observed non-uniform structures within their DCNs. Although these experiences are drawn from EPS DCNs, multiple generations (switches with different radixes) will likely be mixed even in OCS DCNs, so DC operators must be able to manage Clos fabrics with non-uniform stages. Apart from DC evolution, there are other situations where Clos fabrics could be composed of non-uniform stages. Even if a network initially has a uniform structure, the structure could be corrupted by link or terminal failures. Suppose a pandemic or other major disruption were to severely impact the supply chain, similar to the delays experienced during the COVID-19 pandemic. In that case, a need might arise to construct a Clos fabric using only the available switches, which may not all be of the same size. Consequently, each stage could potentially consist of different-sized switches.

Since OCS DCNs are designed as circuit-switched networks, Clos fabrics should be designed to be *nonblocking* for efficient operation. Figure 1 illustrates a small example of a circuit-switched Clos fabric. Following the tradition of blocking research, we describe a Clos fabric as a three-stage network, as shown in Fig. 1 (recent DCNs are often depicted with a folded structure where the first and third stages are combined). The external ports (or terminals) at the first- and third-stage switches connect to top-of-rack (ToR) switches or servers. When a connection between terminals in the first and third stages is requested, the internal switch configurations are updated to establish a path between the terminals. If no path is available between the requested terminals, the connection request is *blocked*.

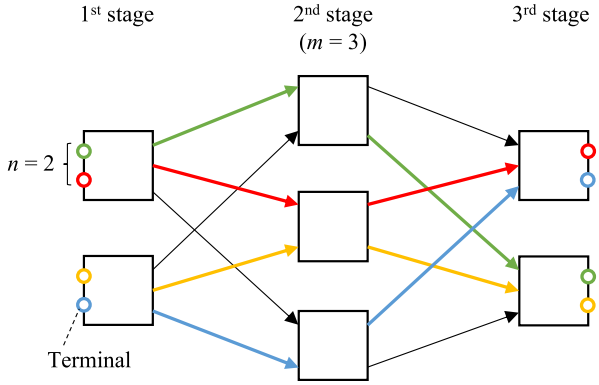


Fig. 1. An example of a nonblocking three-stage Clos fabric with a uniform structure. The terminals can be connected as shown by the colored paths without being blocked.

In 1953, Clos proved the nonblocking condition for a three-stage switching network with symmetric and uniform structures like Fig. 1 [15]. The necessary and sufficient nonblocking condition is $m \geq 2n-1$, where m is the number of switches in the second stage, and n is the number of terminals at each switch in the first and third stages. Here, axially symmetric structures around the second stage are called *symmetric*, whereas structures that have the same switch size (radix) in each stage are called *uniform* (rigorous definitions are given in Section 2.A). The classic condition has been extended to more general (asymmetric and non-uniform) structures. References [16–20] show nonblocking conditions for some "general" structures. However, [16,17] only deal with asymmetry. Even the most general condition [18] assumes the second stage to be uniform. Nevertheless, the nonblocking condition of [18] is revealed to have limitations in certain network structures, as described in Section 4.B. References [19,20] studied nonblocking conditions for the general structure, but only sufficient conditions. Thus, no existing work presents necessary and sufficient nonblocking conditions for the general structure.

We shall briefly elucidate why the nonblocking condition of [18] may fail to correctly assess nonblocking properties in some structures. Following past work [15–17,19], [18] relies on "combinatorial arguments" to formulate the nonblocking condition. Nonetheless, [21] indicates that combinatorial arguments are quite intricate and somewhat error-prone. Although combinatorial arguments are still used in the analysis of recent optical switching networks [22–25], the approach we adopt here is different.

Instead of combinatorial arguments, mathematical programming such as ILP has been used recently to analyze switching networks [20,21,26–28]. These studies, however, do not help in deriving the necessary and sufficient nonblocking conditions for the general structure. Reference [26] uses mathematical programming to analyze the multirate nonblocking condition for the symmetric and uniform structure. Reference [21] also uses mathematical programming to deal with crosstalk in a nonblocking structure called multilog. Although the two studies well utilized mathematical programming to analyze complex signal properties such as multirate and crosstalk, they only dealt with symmetric and uniform network structures. References [27,28] utilized mathematical programming to search for the optimal structural parameters (such as m and n in Fig. 1). However, both analyzed symmetric and uniform structures. Reference [20] derives a sufficient nonblocking condition for any topology, and so is not limited to Clos fabrics. However, [20] employs mathematical programming just to convert the topology into the three-stage structure in pre-processing. Thus, their formulation does not provide any indications or elements for understanding the nonblocking property of the general structure. As no work has shown how to use mathematical programming for blocking analysis of the general structure, we develop our own formulation.

This paper presents the nonblocking condition for Clos fabrics with non-uniform stages. Among several classes of nonblocking properties [29], we study here strict-sense nonblocking (SNB). In an SNB network, any pair of idle terminals can be connected without interrupting already established connections. This constitutes the most fundamental nonblocking property, because if a switching network is SNB, then other nonblocking properties also hold [19]. Although we deal with high generality in terms of the fabric structure, three simple assumptions are made on connections, links, and multiplexing: rate-agnostic connections (we do not care how much traffic each connection conveys), per-fiber basis links (we do not care how many wavelength channels an optical fiber conveys), and space-division multiplexing. Clos fabrics consisting of OCSs [4,5,30–34] satisfy our assumptions. In the following, the nonblocking condition for the Clos fabric with non-uniform stages is simply referred to as the *general* nonblocking condition, as it is considered a structural generalization.

The contributions of this paper are summarized as follows.

- Section 3: The general nonblocking condition is presented; precisely, it is the necessary and sufficient condition for a Clos fabric with general structure that includes non-uniform stages. The condition is formulated in integer linear programming (ILP) form, making it efficient to evaluate.
- Section 4: Our general nonblocking condition is assessed with reference to past studies. First, we prove the equivalency of our condition with a well-known condition for the symmetric and uniform structure [15]; this equivalence supports the correctness of our theory. In addition, our condition reveals counter-examples to an existing nonblocking condition defined for partially non-uniform Clos fabrics [18].
- Section 5: We analyze the nonblocking property of non-uniform networks. Under random link failures, which disrupt network uniformity, we quantitatively demonstrate how the nonblocking property can be controlled by the "margin" of the nonblocking condition. Additionally, we present case studies in which network expansion leads to a loss of uniformity, resulting in a significant degradation in the nonblocking property.
- Section 6: We evaluate the computational efficiency of our ILP formulation. Experiments show that our general condition can, for a large DCN with 32K terminals, be evaluated in 19 min, a practical time in the network design phase; note that the nonblocking condition need not be evaluated for every connection request because it is a long-term network property unchanged by any request.

The rest of this paper is organized as follows. Section 2 provides the problem statement. Section 3 presents the general

nonblocking condition and ILP formulation. Section 4 assesses our condition with reference to past studies. Section 5 analyzes the blocking property of non-uniform networks. Section 6 evaluates the computational efficiency of the ILP formulation. Section 7 details related works and the feasibility of OCS DCNs. Finally, Section 8 concludes this paper.

2. PROBLEM STATEMENT

This section defines the problem studied. Section 2.A models a Clos fabric, and Section 2.B describes the states of the fabric. Section 2.C defines our problem. Notations are given in Table 1. Note that a Clos fabric is also referred to as a three-stage switching network following the traditional literature, hereafter.

A. Structure of a Three-Stage Switching Network

The structure of a three-stage switching network is defined as follows (Fig. 2). A switching network is defined as a set of *switches* connected by directional links. The set of switches is partitioned into three stages; the sets of first-, second-, and third-stage switches are labeled R_1, M, and R_3, respectively.

Table 1. Notations for Network Structure (Top), State (Middle), and Constraints (Bottom)

Symbols	Description
R_1 and R_3	Set of switches in the first or third stage
r_1 and r_3	Number of switches in the first/third stage
r	Number of switches in the first/third stage for a symmetric structure
M	Set of switches in the second stage
m	Number of switches in the second stage
V_1 and V_3	Matrix representing the numbers of links between the first/third and second stages
$v_{1,ik}$ and $v_{3,ik}$	Number of links between first/third-stage switch i and second-stage switch k
v	Number of links between a pair of first/third- and second-stage switches in a regular structure
$\boldsymbol{n}_1$ and $\boldsymbol{n}_3$	Vector representing the numbers of terminals connected to the first/third stage
$n_{1,i}$ and $n_{3,i}$	Number of terminals in first/third-stage switch i
n	Number of terminals in a first/third-stage switch with symmetric regular structure
X	3D array representing the network state (established connections)
x_{ikj}	Number of connections from first-stage switch i via second-stage switch k to third-stage switch j
$x_{1,i}$ and $x_{3,i}$	Total number of connections through first/third-stage switch i
$x_{1,ik}$ and $x_{3,ik}$	Total number of connections through first/third-stage switch i and second-stage switch k
$C(X)$	Capacity constraint for network state X, defined by Eq. (1)
$I_{pq}(X)$	Idle condition on switch pair (p, q) for state X, defined by Eq. (2)
$H_{pq}(X)$	Choking condition on switch pair (p, q) for state X, defined by Eq. (3)
$B_{pq}(X)$	Blocking condition on switch pair (p, q) for state X, defined by Eq. (4)

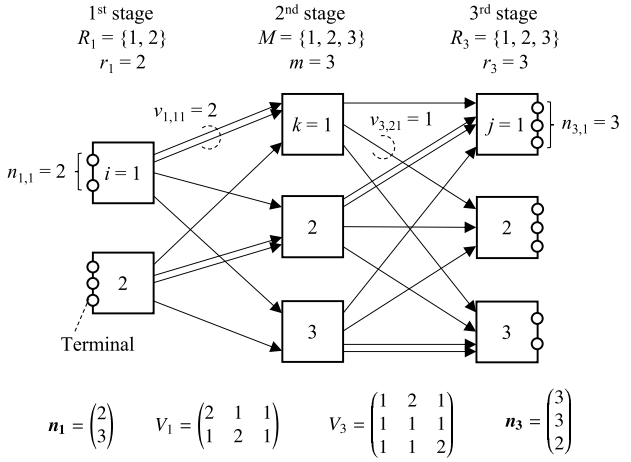


Fig. 2. An example of a three-stage switching network with general structure.

Let $r_1 = |R_1|$, $m = |M|$, and $r_3 = |R_3|$ be the size of these sets. Switches are assumed to be nonblocking.

Links between the first and second stages are represented by the matrix $V_1 \in \mathbb{Z}_{\geq 0}^{r_1 \times m}$, where element $v_{1,ik}$ is the number of links between first-stage switch $i \in R_1$ and second-stage switch $k \in M$. Similarly, links between the third and second stages are represented by the matrix $V_3 \in \mathbb{Z}_{\geq 0}^{r_3 \times m}$.

First- and third-stage switches have *terminals*, which support connections from/to nodes (e.g., ToR switches or servers) outside the network. The terminals in the first stage are represented by vector $\boldsymbol{n}_1 \in \mathbb{Z}_{>0}^{r_1}$, where element $n_{1,i}$ is the number of terminals on the first-stage switch $i \in R_1$. The third stage terminals, $\boldsymbol{n}_3 \in \mathbb{Z}_{>0}^{r_3}$, are defined similarly.

The structure of the three-stage switching network is defined by the tuple $\langle \boldsymbol{n}_1, \boldsymbol{n}_3, V_1, V_3 \rangle$, with several structural properties. First, a network $\langle \boldsymbol{n}_1, \boldsymbol{n}_3, V_1, V_3 \rangle$ is called *symmetric* if it is axisymmetric around the second stage, i.e., $\boldsymbol{n}_1 = \boldsymbol{n}_3 \wedge V_1 = V_3$, where $\wedge$ denotes logical conjunction (AND), indicating that both conditions must hold. Symmetry implies that the first and third stages have an equal number of switches, i.e., $r_1 = r_3$. Additionally, a network is said to be *uniform* on $\boldsymbol{n}_1$ if all first-stage switches have an equal number of terminals, i.e., $\exists n_1 \in \mathbb{Z}_{>0}, \forall i \in R_1, n_{1,i} = n_1$. Similarly, a network is uniform on $\boldsymbol{n}_3$ if $\exists n_3 \in \mathbb{Z}_{>0}, \forall j \in R_3, n_{3,j} = n_3$. Lastly, a network is considered *row-uniform* on V_1 if each first-stage switch has an equal number of links to every second-stage switch, expressed as $\forall i \in R_1[\exists v_{1,i} \in \mathbb{Z}_{>0}, \forall k \in M, v_{1,ik} = v_{1,i}]$. Similarly, row-uniformity on V_3 holds if $\forall j \in R_3[\exists v_{3,j} \in \mathbb{Z}_{>0}, \forall k \in M, v_{3,jk} = v_{3,j}]$.

Figure 3 shows an example of a network whose structure is row-uniform on V_1 and V_3.

Remark 1. If a three-stage switching network is row-uniform on V_1 and V_3, the second-stage switches are structurally indistinguishable.

Definition 1 (Regularity). Three-stage switching network $\langle \boldsymbol{n}_1, \boldsymbol{n}_3, V_1, V_3 \rangle$ is called *regular* if

- the network is uniform on $\boldsymbol{n}_1$ and $\boldsymbol{n}_3$,

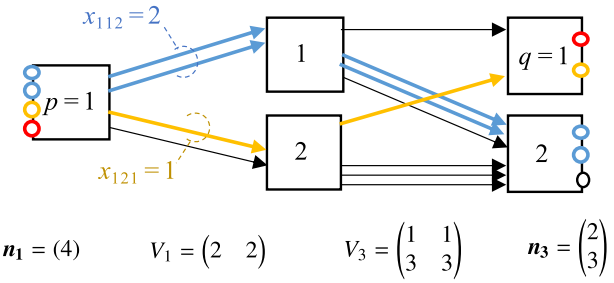


Fig. 3. An example of a three-stage switching network whose structure is row-uniform on V_1 and V_3. The network is in a blocking state. It is also a counter-example of the KL blocking condition [18], as discussed in Section 4.B.

- V_1 and V_3 are completely uniform, i.e., $\exists v \in \mathbb{Z}_{>0}$, $\forall (i, j, k) \in R_1 \times R_3 \times M, v_{1,ik} = v_{3,jk} = v$.

Remark 2. If a three-stage switching network is regular, switches in each stage are structurally indistinguishable.

The regular structure is determined by only six numbers $\langle n_1, r_1, n_3, r_3, v, m\rangle$. If a regular structure is symmetric, it is determined by four numbers $\langle n, r, v, m\rangle$ since $n_1 = n_3 = n$ and $r_1 = r_3 = r$. Figure 1 is an example of the symmetric regular structure given by $\langle n = 2, r = 2, v = 1, m = 3\rangle$.

B. States of a Three-Stage Switching Network

Network states are determined by the connections established on the network. A *connection* is requested by an idle terminal on a first-stage switch to another idle terminal on a third-stage switch. The connection can be established if a path consisting of idle links is found between the terminals. Since we assume per-fiber basis links, each link is occupied by one connection. Established connections are terminated when requested. To identify a connection, we do not need to specify terminals, only the first- and third-stage switches; e.g., the blue connections in Fig. 3 are specified by first-stage switch 1 and third-stage switch 2.

A *blocking state* is defined as follows.

Definition 2 (Blocking state). A network is in a blocking state if both a first-stage switch and a third-stage switch have an idle terminal, but there is no path consisting of idle links between the terminals.

In other words, in a blocking state, a new connection can be requested between a first-stage switch and a third-stage switch, but no connection can be established between them. Figure 3 is an example of a blocking state; first-stage switch p and third-stage switch q have idle terminals (red ones), but there is no idle path between them.

C. Blocking Decision Problem for the General Structure

So far, we have used the term *nonblocking* condition following past work. Hereafter, our discussions will use *blocking* condition instead. This is mainly because the examination of the blocking condition is more straightforward. Of course, the negation of the blocking condition is the nonblocking condition, so this should not cause any confusion.

Strict-sense nonblockingness is defined as follows [35,36]. A three-stage switching network is strict-sense nonblocking if it can always connect each idle terminal in the first stage to an arbitrary idle terminal in the third stage, independently of its current network state and no matter how the connecting paths were selected. The network state is determined by the request sequence and the routing algorithm, so we define *blockingness* in the strict sense as follows.

Definition 3 (Blockingness). A three-stage switching network is blocking if there exists a request sequence and a routing algorithm in which the network state reaches one of the blocking states.

The network in Fig. 3 is blocking because the following request sequence and the routing algorithm lead to a blocking state. A request to third-stage switch $q = 1$ (between the yellow circles in Fig. 3) is followed by two requests to third-stage switch 2 (the blue circles). The request for the yellow circles is routed via the bottom second-stage switch (yellow lines in Fig. 3), while the two requests for the blue circles are routed via the top second-stage switch (blue lines).

Traditionally, the nonblockingness problem simply returns the minimum number m of second-stage switches needed to realize nonblocking, without considering the differences between the second-stage switches. This is because conventional studies assume the row-uniformity of V_1 and V_3 and so do not need to distinguish among the second-stage switches, as described in Remark 1. To deal with the general structure of the three-stage switching network, we introduce the decision version of the blockingness problem BP; i.e., given network structure $\langle \boldsymbol{n}_1, \boldsymbol{n}_3, V_1, V_3\rangle$, we determine whether the network is blocking.

Problem 1 (BP).

- **Input**: three-stage switching network $\langle \boldsymbol{n}_1, \boldsymbol{n}_3, V_1, V_3\rangle$.
- **Output**: yes, if and only if (iff) the network is blocking.

We are allowed to take minutes or hours to solve this problem, because the blocking property is usually evaluated in the network design and construction phase, which usually takes several days. Note that the blocking property need not be evaluated for every connection request because it is a long-term network property unchanged by any new request.

Our blocking problem can be applied recursively to a switching network with three or more stages, similar to past work on nonblockingness [35–37]. This is because a switch in a network can be replaced with another three-stage switching network, as shown in Fig. 4. This switch abstraction can be applied recursively as long as every abstract switch in the recursion is nonblocking (if some abstract switch in a recursion is blocking, the whole network is blocking). Thus, we focus on just three-stage switching networks in this paper.

3. BLOCKING CONDITION FOR THE GENERAL STRUCTURE

This section derives the blocking condition for the general structure in the form of mathematical programming. First, Section 3.A introduces a network state representation based on a three-dimensional array. Then, Section 3.B formally defines a blocking state using the array representation and derives

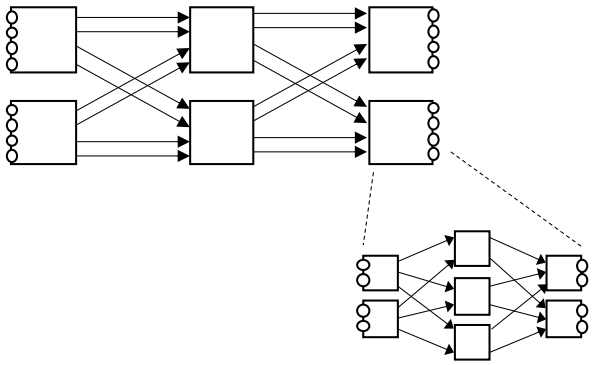


Fig. 4. Recursive abstraction of a three-stage switching network.

the blocking condition for the general structure. Finally, Section 3.C linearizes the blocking condition and casts it in ILP form.

A. State Representation

The state of three-stage switching network $\langle \boldsymbol{n}_1, \boldsymbol{n}_3, V_1, V_3 \rangle$ is determined by connections established on the network as described in Section 2.B. To represent network state, we use three-dimensional array $X \in \mathbb{Z}_{\geq 0}^{r_1 \times m \times r_3}$, where element x_{ikj} is the number of established connections from $i \in R_1$ via $k \in M$ to $j \in R_3$. Initially, a network has no connections, and all the elements are zero: $x_{ikj} = 0, \forall i \in R_1, k \in M, j \in R_3$. The state is called the *initial state*.

In the example network in Fig. 3, we have $x_{112} = 2$ and $x_{121} = 1$, but $x_{ikj} = 0$ for the other paths. For readability, we introduce the following notations:

- $x_{1,i} = \sum_{k \in M} \sum_{j \in R_3} x_{ikj}$ for $i \in R_1$, which represents the total number of connections through first-stage switch i (e.g., $x_{1,1} = 3$ in Fig. 3), and
- $x_{1,ik} = \sum_{j \in R_3} x_{ikj}$ for $(i, k) \in R_1 \times M$, which represents the total number of connections through first-stage switch i and second-stage switch k.

Similarly, $x_{3,j} = \sum_{i \in R_1} \sum_{k \in M} x_{ikj}$ and $x_{3,jk} = \sum_{i \in R_1} x_{ikj}$. In the following, we formulate the blocking condition for the general structure with consideration of the constraints shown in Fig. 5.

Since our aim is to elaborate the blocking condition (Definition 3), we are only interested in the states that can be reached from the initial state given a certain request sequence and routing algorithm. We call such states *reachable states*.

The following proposition describes the four constraints that define the set of all reachable states (the proof is given at the end of this subsection).

Proposition 1 (Reachable states). *Given three-stage switching network* $\langle \boldsymbol{n}_1, \boldsymbol{n}_3, V_1, V_3 \rangle$, *state* $X \in \mathbb{Z}_{\geq 0}^{r_1 \times m \times r_3}$ *is a reachable state iff*

$$x_{1,i} \leq n_{1,i} \qquad \forall i \in R_1, \tag{1a}$$

$$x_{3,j} \leq n_{3,j} \qquad \forall j \in R_3, \tag{1b}$$

$$x_{1,ik} \leq v_{1,ik} \qquad \forall (i,k) \in R_1 \times M, \tag{1c}$$

$$x_{3,jk} \leq v_{3,jk} \qquad \forall (j,k) \in R_3 \times M. \tag{1d}$$

The four constraints of Eq. (1) are collectively called the *capacity constraint*, which is denoted by $C(X)$. Constraint Eqs. (1a) and (1b) indicate that the number of connections is not greater than the number of terminals at a first/third-stage switch, whereas constraint Eqs. (1c) and (1d) indicate that the number of connections is not greater than the number of links between a first/third-stage switch and a second-stage switch, respectively. The state in Fig. 3 satisfies the capacity constraint. Note that this constraint mirrors the *network flow* problem with path variables [38]; Eqs. (1a) and (1b) correspond to the capacity of sources and sinks, respectively, whereas Eqs. (1c) and (1d) are the capacity of links.

Thanks to Proposition 1, we now know if state X is a reachable state. Due to the definitions of the blocking network (Definition 3) and reachable state, a network is blocking if, and only if, the set of reachable states contains a blocking state. Thus, in the next subsection, we represent a blocking state based on state X and provide a blocking condition.

Finally, we prove Proposition 1.

Proof of Proposition 1. First, we show that a reachable state satisfies the capacity constraint. Next, we show that, if a

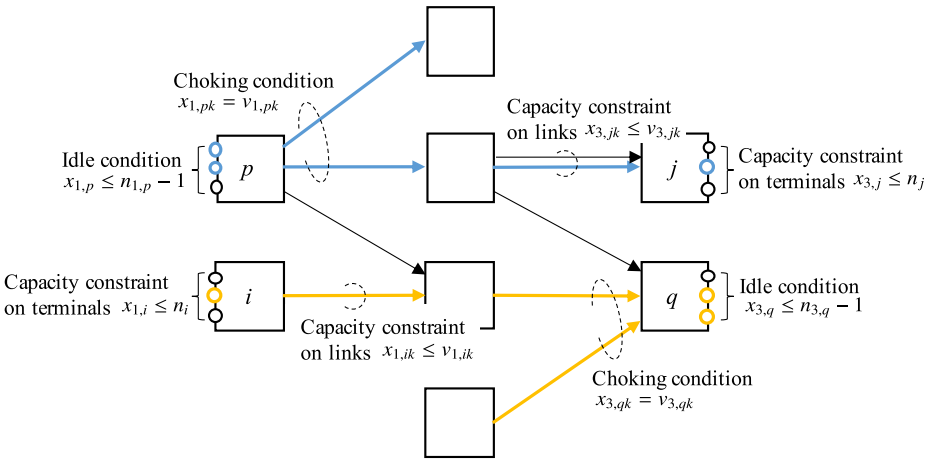


Fig. 5. An illustration of terms used in the description of the blocking condition between first-stage switch p and third-stage switch q.

state satisfies the capacity constraint, then there exist a request sequence and a routing algorithm that lead to the state.

The initial state satisfies the capacity constraint because all elements are zero. Whenever a new connection is established, the capacity constraint is not violated because idle terminals and idle links are used. Whenever an existing connection is terminated, the capacity constraint is also not violated because no element x_{ikj} increases. Thus, every reachable state satisfies the capacity constraint.

Here we construct a request sequence and a routing algorithm for given state $X^* \in \mathbb{Z}_{\geq 0}^{r_1 \times m \times r_3}$. Let X be the current network state. At the start, X is the initial state. As long as current state X is less than target state X^*, that is, $X < X^*$, we repeat the following. Since we have $X < X^*$, there exists $i \in R_1$, $k \in M$, $j \in R_3$ such that $x_{ikj} < x^*_{ikj}$. So we generate a connection request from first-stage switch i to third-stage switch j and route this request via second-stage switch k; element x_{ikj} is incremented by 1. This process terminates in a finite number of steps since a network can hold only a finite number of connections. When the process terminates, current state X is target state X^*, i.e., $X = X^*$. Thus, the generated requests and the path selections are the request sequence and the routing algorithm that lead to given state X^*. □

B. Formulation in Mathematical Programming

This subsection describes blocking states based on state X and provides the blocking condition in the form of mathematical programming. First, we define the blocking states with respect to a pair of first/third-stage switches $(p, q) \in R_1 \times R_3$. Due to Definition 2, in a blocking state, there exists a pair of first/third-stage switches (p, q) such that both first-stage switch p and third-stage switch q have an idle terminal, but there is no path consisting of idle links between them. We denote such a state by a *blocking state on* (p, q). Trivially, state X is blocking if, and only if, there exists $(p, q) \in R_1 \times R_3$ such that state X is a blocking state on (p, q). The following proposition describes the two conditions that define the set of all blocking states on (p, q).

Proposition 2 [Blocking states on (p, q)]. *Given three-stage switching network $\langle \boldsymbol{n}_1, \boldsymbol{n}_3, V_1, V_3 \rangle$ with a pair of first/third-stage switches $(p, q) \in R_1 \times R_3$, state $X \in \mathbb{Z}_{\geq 0}^{r_1 \times m \times r_3}$ is a blocking state on (p, q) iff*

$$x_{1,p} \leq n_{1,p} - 1 \wedge x_{3,q} \leq n_{3,q} - 1, \tag{2}$$

$$x_{1,pk} = v_{1,pk} \vee x_{3,qk} = v_{3,qk}, \forall k \in M, \tag{3}$$

where $\vee$ denotes logical disjunction (OR), indicating that at least one of the conditions must hold.

Proof. Constraint Eq. (2) indicates that switches p and q each have an idle terminal. Constraint Eq. (3) indicates that every second-stage switch $k \in M$ has no idle links to switch p or to switch q. □

Constraint Eq. (2) is called the *idle condition on (p,q)* and is denoted by $I_{pq}(X)$. Constraint Eq. (3) is called the *choking condition on (p,q)* and is denoted by $H_{pq}(X)$. Note that the choking condition is in the form of the complementarity condition [39], which is $\boldsymbol{x} \geq 0$, $\boldsymbol{y} \geq 0$, $\boldsymbol{x}^\intercal \boldsymbol{y} = 0$, where $\boldsymbol{x} = (v_{1,p1} - x_{1,p1}, \ldots)$ and $\boldsymbol{y} = (v_{3,q1} - x_{3,q1}, \ldots)$. As this is a nonlinear condition, we linearize in Section 3.C.

Next, we formulate the *blocking condition on* (p, q), $B_{pq}(X)$. Condition $B_{pq}(X)$ holds if and only if network state X is a reachable state and is also a blocking state on (p, q), which are defined in Propositions 1 and 2, respectively.

Definition 4 [Blocking condition on (p, q)]. Given three-stage switching network $\langle \boldsymbol{n}_1, \boldsymbol{n}_3, V_1, V_3 \rangle$ with a pair of first/third-stage switches $(p, q) \in R_1 \times R_3$ and state $X \in \mathbb{Z}_{\geq 0}^{r_1 \times m \times r_3}$, we define

$$B_{pq}(X) = C(X) \wedge I_{pq}(X) \wedge H_{pq}(X), \tag{4}$$

that is, $B_{pq}(X)$ consists of constraint Eqs. (1)–(3).

Since the idle and choking conditions are defined only by connections to p or q, connections not associated with either can be ignored in the blocking condition, as described in the following corollary.

Corollary 1. *Assume the capacity constraint $C(X)$ holds due to* Eq. (1). *Given network state $X \in \mathbb{Z}_{\geq 0}^{r_1 \times m \times r_3}$ and another state X' that removes connections to switches other than p or q, we have*

$$x'_{ikj} = \begin{cases} x_{ikj} & i = p \vee j = q \\ 0 & \text{otherwise.} \end{cases}$$

Then, we have $B_{pq}(X) \Leftrightarrow B_{pq}(X')$.

Proof. It is obvious from Definition 4. □

Finally, we provide the blocking condition to solve BP (Problem 1).

Theorem 1. *Three-stage switching network $\langle \boldsymbol{n}_1, \boldsymbol{n}_3, V_1, V_3 \rangle$ is blocking iff*

$$\exists (p, q) \in R_1 \times R_3, \exists X \in \mathbb{Z}_{\geq 0}^{r_1 \times m \times r_3}, B_{pq}(X). \tag{5}$$

Proof. From Definition 3, a network is blocking if and only if there exists a network state that is reachable and a blocking state at the same time. Such a state exists if and only if there exist network state X and a pair (p, q) of switches for which the blocking condition on (p, q), $B_{pq}(X)$, holds due to Definition 4. □

Blocking condition $B_{pq}(X)$ is represented as the network flow constraints $C(X)$ with additional conditions $I_{pq}(X)$ and $H_{pq}(X)$. Actually, this is a reasonable formulation because BP is a *meta* problem of the network flow problem (more precisely, the online multicommodity flow problem [40]), i.e., the network flow problem is the problem of *assigning paths on a given network for a given request sequence*, while BP is *determining whether the network flow problem with a given network is feasible for arbitrary request sequences*. Our blocking condition gives an intuitive understanding of this relationship between BP and the network flow problem.

C. Formulation in Integer Linear Programming

The blocking condition $B_{pq}(X)$ defined in Section 3.B includes a complementarity condition, i.e., the choking condition $H_{pq}(X)$. This subsection reformulates $H_{pq}(X)$ in linear programming form to permit the use of a computationally efficient ILP solver. Since $H_{pq}(X)$ is in the form of disjunctive programming [41], we convert it into linear programming as follows.

Lemma 1. *Assume the link capacity constraints (1c) and (1d) hold. The choking condition $H_{pq}(X)$ of Eq. (3) is represented as a linear condition by introducing binary vector $\boldsymbol{y} \in \{0, 1\}^m$, i.e.,*

$$\text{Eq. (3)} \Leftrightarrow \forall k \in M, \exists y_k \in \{0, 1\} \left[x_{1,pk} \geq y_k v_{1,pk} \wedge x_{3,qk} \geq (1 - y_k) v_{3,qk}\right]. \tag{6}$$

Proof. From the definition of state X and the capacity constraints, we have $0 \leq x_{1,pk} \leq v_{1,pk}$ and $0 \leq x_{3,qk} \leq v_{3,qk}$. Thus, we have the following for $\forall k \in M$:

$$\begin{aligned} & x_{1,pk} = v_{1,pk} \vee x_{3,qk} = v_{3,qk} \\ \Leftrightarrow\ & x_{1,pk} \geq v_{1,pk} \vee x_{3,qk} \geq v_{3,qk} \\ \Leftrightarrow\ & \exists y_k \in \{0, 1\},\ x_{1,pk} \geq y_k v_{1,pk} \wedge x_{3,qk} \geq (1 - y_k) v_{3,qk}. \end{aligned}$$

Note that binary variable y_k indicates which equality condition is satisfied. □

Denoting the right-hand side of Eq. (6) by $H^{\mathrm{ILP}}_{pq}(X, \boldsymbol{y})$, we can represent the blocking condition in ILP form.

Theorem 2. *Three-stage switching network $\langle \boldsymbol{n}_1, \boldsymbol{n}_3, V_1, V_3 \rangle$ is blocking iff*

$$\exists (p, q) \in R_1 \times R_3, \exists X \in \mathbb{Z}_{\geq 0}^{r_1 \times m \times r_3}, \boldsymbol{y} \in \{0, 1\}^m,\ B^{\mathrm{ILP}}_{pq}(X, \boldsymbol{y}), \tag{7}$$

where B^{ILP}_{pq} is defined as follows:

$$B^{\mathrm{ILP}}_{pq}(X, \boldsymbol{y}) = C(X) \wedge I_{pq}(X) \wedge H^{\mathrm{ILP}}_{pq}(X, \boldsymbol{y}); \tag{8}$$

that is, $B^{\mathrm{ILP}}_{pq}(X, \boldsymbol{y})$ consists of constraint Eqs. (1), (2), and (6).

Proof. It is obvious from Lemma 1. □

Now the general blocking condition of Theorem 1 is well mapped into the ILP form of Theorem 2, as the derivation process (Lemma 1) involves no approximation. The whole formulation is shown below for readability, i.e., $B^{\mathrm{ILP}}_{pq}(X, \boldsymbol{y}) \Leftrightarrow$ Eq. (9).

The capacity constraint Eq. (1):

$$x_{1,i} \leq n_{1,i} \qquad \forall i \in R_1, \tag{9a}$$

$$x_{3,j} \leq n_{3,j} \qquad \forall j \in R_3, \tag{9b}$$

$$x_{1,ik} \leq v_{1,ik} \qquad \forall (i, k) \in R_1 \times M, \tag{9c}$$

$$x_{3,jk} \leq v_{3,jk} \qquad \forall (j, k) \in R_3 \times M. \tag{9d}$$

The idle condition Eq. (2):

$$x_{1,p} \leq n_{1,p} - 1, \tag{9e}$$

$$x_{3,q} \leq n_{3,q} - 1. \tag{9f}$$

Algorithm 1. BpIlp Algorithm for the General Structure

Input: Three-stage switching network $\langle \boldsymbol{n}_1, \boldsymbol{n}_3, V_1, V_3 \rangle$.
Output: Yes iff the network is blocking.

```
for each p ∈ R_1 do                          //( First-stage switch)
  for each q ∈ R_3 do                        //( Third-stage switch)
    if ILP Eq. (9) has a feasible solution (X, y) then
      return Yes
return No
```

The choking condition Eq. (6):

$$x_{1,pk} \geq y_k v_{1,pk} \qquad \forall k \in M, \tag{9g}$$

$$x_{3,qk} \geq (1 - y_k) v_{3,qk} \qquad \forall k \in M. \tag{9h}$$

ILP Eq. (9) includes integer variables x_{ikj} and binary variables y_k. The number of x_{ikj} is $m(r_1 + r_3) - 1$ because we only consider x_{ikj} for $i = p$ or $j = q$ due to Corollary 1. The number of y_k is m.

There still is a disjunctive condition in Eq. (7), i.e., $\exists (p, q)$. Although we might be able to reformulate it into a single ILP by using some linearizing technique like Lemma 1, here we simply evaluate Eq. (9) for each (p, q) so as not to increase the number of decision variables. The algorithm is named BpIlp and is written in Algorithm 1. BpIlp solves ILP Eq. (9) $O(r_1 r_3)$ times.

4. EVALUATION OF CORRECTNESS AGAINST PAST STUDIES

This section assesses our general condition against past studies. Section 4.A proves the equivalence of our general condition and the classic condition [15]. Section 4.B uses our condition to reveal the incorrectness of a known partially non-uniform condition [18].

A. Equivalence with Clos's Classic Condition

This subsection provides validation by confirming the equivalence of our general condition and Clos's classic condition. Clos gave the first blocking condition, $m < 2n - 1$ [15], for the following symmetric regular structure, which is called SRV1 structure in this paper.

Definition 5 (SRV1 structure). A three-stage switching network has SRV1 structure if

- the network is symmetric and regular;
- the first and third stages have two or more switches, i.e., $r \geq 2$; and
- the number of links between switches is $v = 1$.

The SRV1 structure is determined only by $\langle n, r, m \rangle$.

Substituting the symmetric and regular conditions into Theorem 1, we have the following corollary that gives us the blocking condition for the SRV1 structure.

Corollary 2. *SRV1 structure $\langle n, r, m \rangle$ is blocking iff*

$$\exists (p, q, X) \in R_1 \times R_3 \times \mathbb{Z}_{\geq 0}^{r_1 \times m \times r_3},$$

$$x_{1,i} \leq n \qquad \forall i \in R_1, \tag{10a}$$

$$x_{3,j} \leq n \qquad \forall j \in R_3, \tag{10b}$$

$$x_{1,ik} \leq v = 1 \qquad \forall (i, k) \in R_1 \times M, \tag{10c}$$

$$x_{3,jk} \leq v = 1 \qquad \forall (j, k) \in R_3 \times M, \tag{10d}$$

$$x_{1,p} \leq n - 1 \wedge x_{3,q} \leq n - 1, \tag{10e}$$

$$x_{1,pk} = v = 1 \vee x_{3,qk} = v = 1 \quad \forall k \in M. \tag{10f}$$

Note that due to Remark 2, which characterizes the regular structure, it is sufficient to examine any pair of first- and third-stage switches.

The general condition Eq. (10) of Corollary 2 and the classic condition, $m < 2n - 1$, look different, but as the following theorem shows, these two conditions are equivalent.

Theorem 3. *Let $\langle n, r, m\rangle$ be an SRV1 structure. Then, the condition of Corollary 2 holds if and only if the classic condition holds; that is,*

$$\text{Eq.(10)} \Leftrightarrow m < 2n - 1. \tag{11}$$

Proof. First, we prove the *only if* ($\Rightarrow$) part: we assume condition Eqs. (10) of Corollary 2 and prove the classic condition, $m < 2n - 1$. From the choking condition Eq. (10f), we have $(1 - x_{1,pk}) + (1 - x_{3,qk}) \leq 1$ for all $k \in M$. Summing up these equations for all k, we have $m - x_{1,p} + m - x_{3,q} \leq m$. Combining with the idle condition Eq. (10e), we have $m \leq x_{1,p} + x_{3,q} \leq (n - 1) + (n - 1) = 2n - 2$. Since m is an integer, we finally have the classic condition, $m < 2n - 1$.

Next, we prove the *if* ($\Leftarrow$) part: we assume the classic condition $m < 2n - 1$ and prove condition Eqs. (10) of Corollary 2. We give a constructive proof; we construct a network state x_{ikj} based on m and show that state x_{ikj} satisfies all Corollary 2's condition Eqs. (10). We divide the set M of the second-stage switches into two parts, $M_1 = \{1, 2, \ldots, \lfloor m/2 \rfloor\}$ and $M_3 = \{\lfloor m/2 \rfloor + 1, \ldots, m\}$. Without loss of generality, we can assume $p \neq 1$ and $q \neq 1$ (remember $r \geq 2$ in an SRV1 structure) and define the network state x_{ikj} as follows:

$$x_{ikj} = \begin{cases} 1 & i = p, k \in M_1, j = 1 \\ 1 & i = 1, k \in M_3, j = q \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

In the following, we show that the above defined network state x_{ikj} satisfies all conditions from Eqs. (10a)–(10f).

As preparation, we show $|M_1| \leq n - 1$ and $|M_3| \leq n - 1$ via a case analysis on the parity of m:

- If m is even, we have $|M_1| = |M_3| = m/2$. Together with the classic condition, $m \leq 2n - 2$, we have $|M_1| = |M_3| \leq n - 1$.
- If m is odd, then we can assume $m = 2a + 1$ for some integer a. From the classic condition, we have $2a + 1 < 2n - 1 \Rightarrow a < n - 1$. Thus we have $|M_1| = a < |M_3| = a + 1 \leq n - 1$.

As for condition Eqs. (10a), (10b), and (10e), it is sufficient to show two conditions $x_{1,i} \leq n - 1$, $\forall i \in R_1$ and $x_{3,j} \leq n - 1$, $\forall j \in R_3$. We prove the former condition, $x_{1,i} \leq n - 1, \forall i \in R_1$; the proof of the latter one, $x_{3,j} \leq n - 1$, $\forall j \in R_3$, is ditto and omitted. We perform case analysis on i:

- If i equals 1, from the network state's definition Eq. (12) and $|M_3| \leq n - 1$, we have

$$x_{1,1} = \sum_{k \in M} \sum_{j \in R_3} x_{1kj} = \sum_{k \in M_3} x_{1kq} = |M_3| \leq n - 1.$$

- If i equals p, from the network state's definition Eq. (12) and $|M_1| \leq n - 1$, we also have

$$x_{1,p} = \sum_{k \in M} \sum_{j \in R_3} x_{pkj} = \sum_{k \in M_1} x_{pk1} = |M_1| \leq n - 1.$$

- If i neither equals 1 nor p, then we have $x_{1,i} = 0$.

Thus, we have $x_{1,i} \leq n - 1$, $\forall i \in R_1$.

As for condition Eqs. (10c) and (10d), the proof is similar. Thus, we prove only condition Eq. (10c), $x_{1,ik} \leq 1$, and omit the proof of condition Eq. (10d). Calculating $x_{1,ik}$ based on definition Eq. (12), we have

$$x_{1,ik} = \begin{cases} \sum_{j \in R_3} x_{1kj} = x_{1kq} \leq 1 & i = 1, \\ \sum_{j \in R_3} x_{pkj} = x_{pk1} \leq 1 & i = p, \\ 0 & \text{otherwise.} \end{cases}$$

Hence, we have $x_{1,ik} \leq 1$.

As for the last condition Eq. (10f), any second switch, $k \in M$, belongs to either M_1 or M_3, due to the definition of M_1 and M_3. If $k \in M_1$, we have $x_{1,pk} = \sum_{j \in R_3} x_{pkj} = 1$. If $k \in M_3$, we have $x_{1,qk} = \sum_{i \in R_1} x_{ikq} = 1$. Thus, we have the condition Eq. (10f), and, finally we have all the conditions of Corollary 2. □

B. Limitations of the Known Partially Non-uniform Condition

Our general blocking condition reveals counter-examples to the blocking condition presented in [18]. The blocking condition is called the KL condition here. The KL condition is claimed to be the necessary and sufficient blocking condition for the structure row-uniform on (V_1, V_3). The KL condition is defined based on combinatorial arguments as follows:

$$m \leq \max_{(p,q) \in R_1 \times R_3} \left\{ \left\lfloor \frac{\min\left\{ n_{1,p} - 1, \sum_{j \in R_3 \setminus \{q\}} n_{3,j} \right\}}{v_{1,p}} \right\rfloor + \left\lfloor \frac{\min\left\{ n_{3,q} - 1, \sum_{i \in R_1 \setminus \{p\}} n_{1,i} \right\}}{v_{3,q}} \right\rfloor \right\}, \tag{13}$$

where $v_{1,p} = v_{1,pk} \forall (p, k) \in R_1 \times M$ and $v_{3,q} = v_{3,qk} \forall (q, k) \in R_3 \times M$. Since [18] deals with multirate

connections, we need to restrict it to its single-rate variant, as $\beta = b = B = w = 1$ in [18]. Actually, the KL condition is neither a necessary nor a sufficient condition, as the following counter-examples show.

1. Counter-Example of a Blocking Network

A counter-example is given in Fig. 3. The network is blocking because it has the blocking state indicated by the colored connections. The KL blocking condition, however, incorrectly determines that the network is nonblocking.

The reason for the false decision is as follows. Assume $p = 1$ and $q = 1$ in Eq. (13); this means that the pair of idle terminals at p and q (the red ones in Fig. 3) are used for the blocking test. For Fig. 3, the first term of Eq. (13) becomes 1, which indicates that there exists *one* second-stage switch such that all links between p and the top second-stage switch can be occupied (or *choked*) by the blue connections. The second term is 0, which indicates that there exists *no* second-stage switch such that all the links between the switch and q can be choked. The second term is, unfortunately, incorrect because the yellow connection chokes all the links between the bottom second-stage switch and q. Since all the second-stage switches can be choked by either p or q, there is a blocking state between p and q.

The reason for the false second term is as follows. Although connections to q could be established from p (the yellow one), the KL condition excludes such connections, as shown in $R_1 \backslash \{p\}$ in the second term.

It is worth noting that the BpIlp algorithm provides an example of the blocking state at Line 3 in Algorithm 1, which helps us find counter-examples.

2. Counter-Example of a Nonblocking Network

Another counter-example is given in Fig. 6. The KL condition Eq. (13) indicates that this network is blocking, but actually it is not. The reason for the false decision is as follows. Assume $p = 1$ and $q = 1$ in Eq. (13); note that the red terminals at p and q must be idle for the blocking test. For Fig. 6, the first term of Eq. (13) is 2, which indicates that the blue connections from p choke all links to the *two* top second-stage switches. The second term is 1, which indicates that connections to q can choke all links from the remaining (bottom) second-stage switch. However, the second term is incorrect. The links cannot be choked and a connection between the red terminals is not blocked (it can be established through the bottom second-stage switch).

The reason for the false second term is as follows. Although just one connection can be established from the bottom second-stage switch to q in Fig. 6, the KL condition incorrectly decides that two connections can be established. This is because the second term of the KL condition ignores the number of links between the first and second stages, i.e., the second term includes $v_{3,q}$ but not $v_{1,i}$.

As seen in the complicated details of the counter-examples, the KL condition, based on the combinatorial arguments, combines several different constraints into a single inequality, which may lead to inaccuracies in certain network structures. Given the difficulty of fixing the condition in its current form, we formulated the blocking condition in the form of mathematical programming in Section 3.

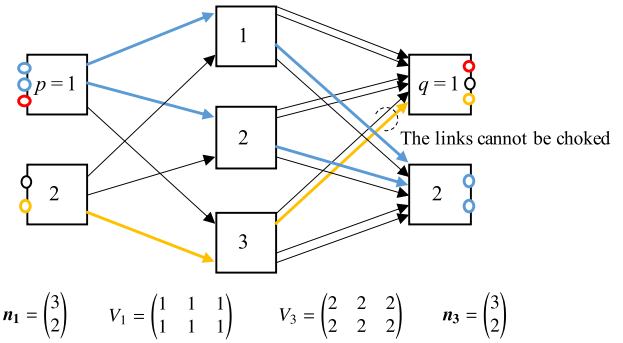


Fig. 6. A counter-example of the KL blocking condition [18].

5. NETWORK DESIGN IMPLICATIONS ARISING FROM THE GENERAL BLOCKING CONDITION

This section analyzes the blocking property using our general blocking condition and offers design and operational implications. Section 5.A examines the blocking property under random link failures. Section 5.B investigates the blocking property under DCN evolution as well as the cost benefits of using different-sized switches.

A. Blocking Property under Random Link Failures

This section examines the blocking property under random link failures. Starting with a symmetric regular nonblocking network, we randomly remove inter-switch links and examine the probability that the network becomes a blocking structure. First, we describe the initial network structure. Let the port count of each switch be N, considering $N \times N$ switches. In this subsection, we configure each stage with $r = m = 32$ switches, each having $N = 256$ ports. Using the known nonblocking condition for symmetric regular networks [42],

$$m \geq 2\left\lfloor \frac{n-1}{v} \right\rfloor + 1, \tag{14}$$

we determine the remaining parameters, $n = 128$ and $v = 8$. There are $mrv = 8192$ links between the first and second stages, and there are also 8192 links between the third and second stages.

Since the structure of $n = 128$ is at the threshold of nonblocking, the network is likely to become blocking with even a few link failures. To make the network tolerant against link failures, we decrement the number of available terminals, n, with the loss of network size nr, i.e., the total number of terminals in the first or third stage. By decrementing n, the maximum number of connections to be established from a first/third-stage switch decreases, making the network less prone to blocking. For the network without decrement, i.e., $\Delta n = 0$, the network size is $nr = 4096$. With each decrement of n, the network size nr is reduced by $r = 32$. We conducted 32 trials for each combination of decrement Δn and link failure rate, so as to estimate the probability of the network becoming blocking.

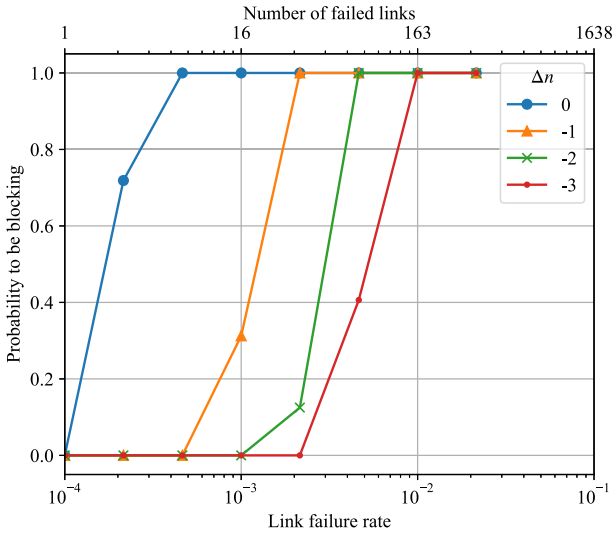


Fig. 7. Probability of a network becoming blocking versus the link failure rate (the number of failed links). The initial network has a symmetric regular structure of $(n, r, v, m) = (128, 32, 8, 32)$. The inter-switch links fail randomly.

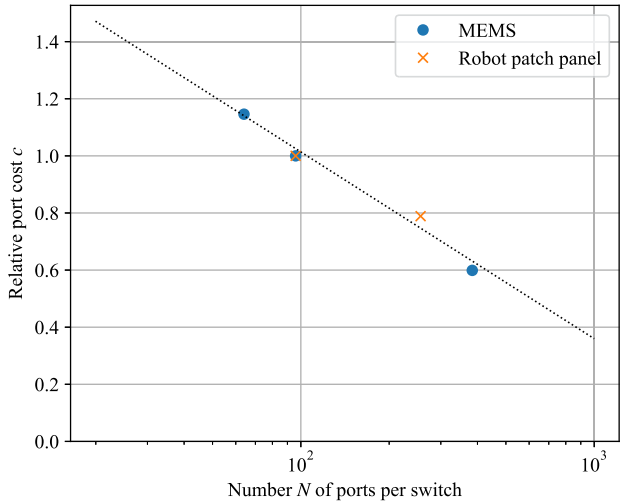


Fig. 8. Relative port cost c versus switch size N based on actual market products. The port cost is normalized by 96-port switches. The dotted line is a logarithmic approximation, $c(N) = -0.284 \cdot \ln(N) + 2.3216$.

Figure 7 demonstrates the probability of the network becoming blocking against the link failure rate, or equivalently the number of failed links. First, we describe the network without terminal decrement ($\Delta n = 0$). The network can withstand a single link failure. However, with three link failures, the network would become blocking with 75% probability. With seven link failures, the network would be almost certain to become blocking. We give a brief analysis of the three link failure case; if the failures occur across the first and third stages (i.e., both in V_1 and V_3), which is the 75% probability, the network is unable to find a commonly available second-stage switch, resulting in a blocking structure.

By decrementing n, networks are less likely to be blocking. To understand this behavior, we conduct a simple analysis. According to the known nonblocking condition for regular networks, Eq. (14), each first/third-stage switch must have inter-switch links that are approximately twice the number of terminals, i.e., $mv \gtrsim 2n$. Therefore, reducing the number of terminals by one provides a margin equivalent to two additional inter-switch links. In other words, if two inter-switch links fail on the same first- or third-stage switch, the margin is used up. Thus, we consider the probability of multiple link failures occurring on the same first-stage switch when f links fail between the first and second stages (V_1). This problem resembles the classic "birthday paradox," where the probability is given by $1 - \prod_{i=0}^{f-1}(r-i)/r$. As the failure rate is 0.002 for $\Delta n = -1$, corresponding to 35 failed links in the entire network (or $f \approx 18$ for V_1), the probability of multiple link failures at a single switch is 99.8%. Therefore, the two-link margin is certainly used up and the probability of the network becoming blocking approaches 1.

The results imply that we can enhance the fault tolerance of the nonblocking network by sacrificing network size nr. Even if failures are not immediately fixed, preemptively reducing the network size allows for continued operation as a nonblocking network. Note that our general blocking condition enables the evaluation of the trade-off between network size and fault tolerance, which provides valuable insights for optimizing network design and operational strategies.

B. Blocking Property for DCN Evolution

This subsection studies the blocking property under DCN evolution. First, to justify using different-sized switches for DCN evolution, Section 5.B.1 examines the economic benefits. Section 5.B.2 investigates the blocking property of DCNs expanded with larger switches.

1. *Economic Efficiency of DCN Evolution Using Larger Switches*

First, we show the relationship between switch size and per-port cost based on actual market prices. Switch size (number of ports) is defined by the number of Tx and Rx pairs in accordance with industry conventions. Since the authors are not allowed to disclose the absolute prices, the analysis is conducted based on relative costs. Figure 8 illustrates the port cost per switch size for micro-electromechanical systems (MEMS) switches and robotic patch panels. For both series, port costs are normalized by that of 96-port switches. As the switch size increases, the port cost decreases for both series because switches have common management components that do not depend on the port counts. Interestingly, the port costs for both the MEMS and robot series decrease at a similar rate. To quantify this trend, we applied a logarithmic fit to the combined data from both series, resulting in the port cost c as a function of the number N of ports: $c(N) = -0.284 \cdot \ln(N) + 2.3216$, with a correlation coefficient of $R^2 = 0.9797$. Note that the difference in port cost when doubling the switch size, $c(N) - c(2N)$, is approximately 0.197, regardless of the initial switch size.

Consider the following DCN evolution scenario. Initially, the network is composed of switches of uniform size N. Then, the network is expanded, twofold or fourfold, with larger

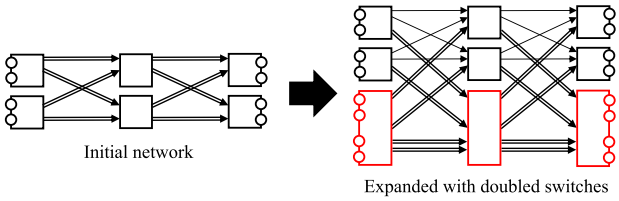


Fig. 9. Network evolution with and without larger (doubled) switches. The initial network consists of four-port switches and has the structure of $(n, r, v, m) = (2, 2, 2, 2)$. The network is expanded with the red doubled eight-port switches in the right.

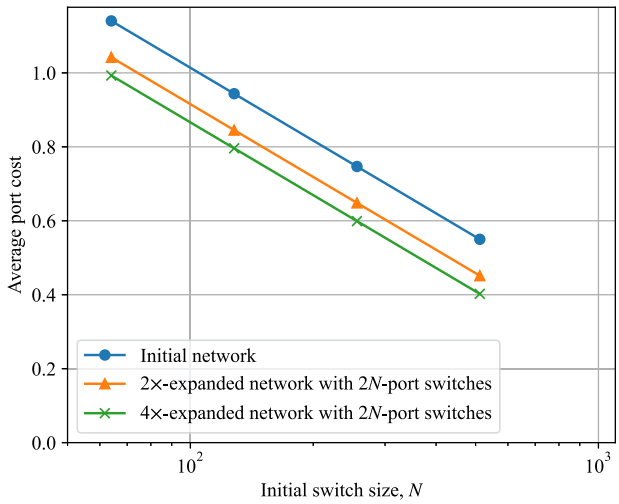


Fig. 10. Average port cost of a network expanded with doubled switches.

switches of size $2N$ (Fig. 9). Finally, the average port cost of the expanded network is calculated. The number of switches in the initial network is given by $r_1 + r_3 + m$. To double the network size, $(r_1 + r_3 + m)/2$ of $2N$-port switches are added. Similarly, to quadruple the network size, $3(r_1 + r_3 + m)/2$ of $2N$-port switches are added. Figure 10 shows the average port cost for the expanded networks as a function of the initial switch size N. For the initial switch size of $N = 512$, doubling the network reduces the average port cost by 17.9%, while quadrupling it results in a 26.8% reduction.

Finally, by referencing the absolute port cost described in [33], we investigate the potential cost savings for a large-scale DCN. According to [33], the port cost for a robotic patch panel with approximately 1000 ports is \$100. Assuming the port cost for a 1024-port switch is adjusted to \$100, the cost function is corrected to $c^*(N) = 283 \cdot c(N)$. Considering a network of Google Jupiter scale $nr = 32{,}768$ terminals, we compare the total switch cost for two scenarios: (i) a DCN composed entirely of 512-port switches and (ii) a DCN where half of the network is expanded with 1024-port switches. For simplicity, we assume that the initial network is symmetric and regular with $r_1 = r_3 = m$. As discussed in Section 5.A, we set $n \approx N/2$ to ensure the nonblocking condition for a symmetric regular network. In scenario (i), with $n = N/2 = 256$, the required number of switches per stage is $r = m = 32{,}768/n = 128$. The total switch cost per stage is $N \cdot r \cdot c^*(512)$, resulting in the total network switch cost of 30,622,906. In scenario (ii), where 512-port switches and 1024-port switches each handle 16,384 terminals, we have $N = 512$, $n = 256$, $r = m = 16{,}384/n = 64$, $N' = 1024$, $n' = 512$, and $r' = m' = 16{,}384/n' = 32$, where primes denote the 1024-port switches. [Here, r and r' do not represent the number of first/third-stage switches but rather the number of switches of a specific size; e.g., in Fig. 9 (right), we have $r = 2$ and $r' = 1$. Although this constitutes a notation abuse, we use r and r' in this manner in Section 5.B for the sake of simplicity. The same applies to m and m'.] The switch cost per stage is $N \cdot r \cdot c^*(N) + N' \cdot r' \cdot c^*(N')$, resulting in the total switch cost of 25,141,853. This represents a large cost reduction of \$30,622,906 − \$25,141,853 = \$5,481,053 (−17.9%). Thus, using larger switches in evolving a DCN can result in substantial cost savings. However, as discussed in Section 5.B.2, under certain conditions, network expansion can make it prone to blocking.

2. *Inefficient DCN Evolution Due to Blocking*

This subsection investigates the blocking property of a network expanded using larger switches. When adding switches to expand a network, the network size (number of terminals) should remain proportional to the additional resources; e.g., assuming the original network consists of 32 128-port switches in each stage, if we added 16 256-port switches to each stage, the total network size should double. However, to maintain the nonblockingness of the network, the network size cannot match the additional switches in some cases.

First, we describe the network expansion method employed in this subsection. The method follows that of Section 5.B.1, but we need more detail to evaluate the blocking property. The initial network has a symmetric regular structure. Larger switches are then added to the network; here, an equal number of switches is added to each stage, i.e., $r_1' = r_3' = m'$, where the primes denote the additional switches. Links between the first and second stages, as well as between the third and second stages, are set proportional to switch size; if some ports remain unused with the proportional distribution, more links are added as evenly as possible to avoid leaving any ports between stages unused. The ratio of ports used as terminals at a first/third-stage switch is kept approximately equal, i.e., $n/N \approx n'/N'$; in Fig. 9 (right), we have $n/N = n'/N' = 1/2$. However, if the network would become blocking due to the expansion, we decrement n and n' to restore the nonblocking property.

In this subsection, we consider a *threefold* network expansion. Scaling by a factor of three, rather than two, is not uncommon in DCN evolutions due to constraints such as available space, power capacity, or equipment procurement. The following two expansion scenarios are examined:

(i) $N' = 2N$: The size of the additional switches is twice that of the initial switches. The number of additional switches equals the initial switch count, i.e., $r' = r$, resulting in $1 + 2 \times 1 = 3\times$ expansion.

(ii) $N' = N$: The size of the additional switches is the same as the initial switches. The number of additional switches is twice the initial switch count, i.e., $r' = 2r$.

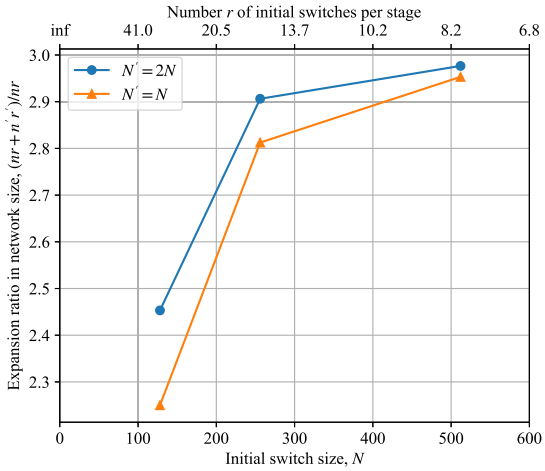
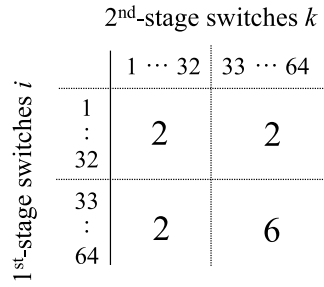


Fig. 11. Expansion ratio versus initial switch size N (number r of switches per stage). The initial network consists of r N-port switches per stage and offers $nr = 2048$ terminals. Switches with 2.0× or 1.0× ports are added to increase the total port count up to 3×. However, the network size might be constrained to less than 3× to maintain the nonblocking property.

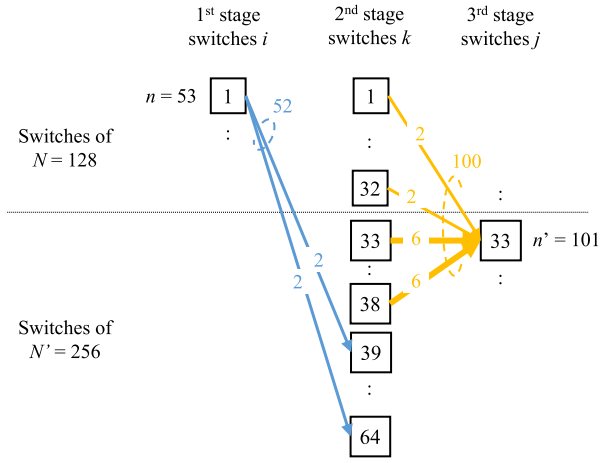
We exemplify scenario (i), as follows. The initial network has $r = m = 32$ switches of $N = 128$ ports per stage. As described in Section 5.B.1, to maintain the nonblocking condition, we have $n/N \approx 1/2$, so the initial network size is $nr = 64 \times 32 = 2048$. We then add $r' = m' = 32$ switches of $N' = 256$ ports to each stage, hopefully increasing the network size by $n'r' = 128 \times 32 = 4096$. The total network size would be $2048 + 4096 = 6144$; if the network is blocking, n and n' would be decremented until nonblockingness is recovered.

Figure 11 illustrates the expansion ratio in the network size for different initial switch sizes. With smaller initial switches, the network cannot be expanded to the expected size, 3×; e.g., for a network with initial switches of $N = 128$, the network size is expanded to less than 2.5×. This is due to the nonblocking constraint and a brief analysis is given using Fig. 12. The inter-switch link matrices V_1 and V_3 are configured as shown in Fig. 12(a); the links are distributed proportionally to switch size with no unused ports. In this link configuration, connections can be established as in Fig. 12(b). Now, we focus on a 128-port first-stage switch of $i = 1$ and a 256-port third-stage switch of $j = 33$. They have one unused terminal each, allowing for a new connection. However, there is no commonly available second-stage switch, so the new connection would be blocked. In this way, when a network consists of smaller switches, each switch pair has only a few links, making the network prone to blocking. To restore the nonblocking property, the network has to have fewer terminals, thereby limiting the network size, as shown in the left side of Fig. 11. On the contrary, with large and fewer switches, networks can be expanded to almost the expected size, as shown in the right side of Fig. 11.

While we have presented that the reasonable expansion method does not lead to the expected increase in network size, there is room to improve the expansion method, such as varying the n/N ratio across switches or configuring inter-switch links unevenly. The key takeaway from this subsection is that



(a) Matrix V_1 representing the link counts between the first and second stages. V_3 is ditto.



(b) Possible blocking state. First-stage switch of $i = 1$ has 52 blue connections, while third-stage switch of $j = 33$ has 100 yellow connections. They have one idle terminal each, but cannot connect them because of the lack of a commonly available second-stage switch. For connection paths, only the links incident on the two switches are depicted.

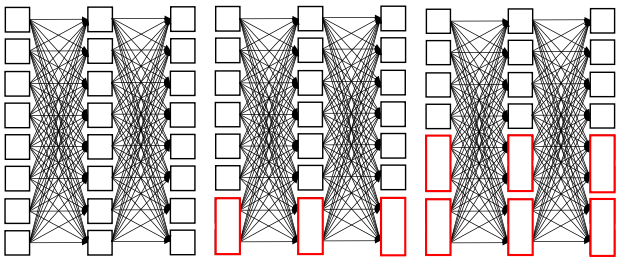
Fig. 12. Analysis of the lowest expansion ratio in Fig. 11, i.e., $N'/N = 2.0$ and $N = 64$. Under the inter-switch link matrices (a), the blocking state (b) could arise.

nonblocking network expansion is not a trivial process, and this insight has only become apparent through our general blocking condition.

6. EVALUATION OF COMPUTATIONAL PERFORMANCE

This section experimentally evaluates the computational performance of the BpIlp algorithm. Section 6.A demonstrates the impact of non-uniformity of network structure under two scenarios: DCN evolution employing different-sized switches and failures of terminals and links. Section 6.B shows the scalability of BpIlp against network size nr. Up to here, BpIlp has been evaluated against nonblocking networks to prevent BpIlp from breaking the loops, but in Section 6.C, the evaluation is performed with blocking networks. Since no nonblocking conditions for the general structures are known, as discussed in Sections 1 and 7.A, no existing algorithms are available for comparison to BpIlp.

Algorithm BpIlp was implemented in Python 3.11. The ILP of Eq. (9) was modeled with PuLP 2.8 [43] and solved with



(a) Coverage of doubled switches: 0 **(b)** Coverage of doubled switches: 0.25 **(c)** Coverage of doubled switches: 0.5

Fig. 13. Network configurations without and with doubled switches. (a) A uniform network of $(r, v, m) = (8, 1, 8)$, which includes no doubled switches. (b) A network with red switches with doubled radixes; the doubled switches cover 25% of the network. (c) A network with red doubled switches covering 50% of the network.

Gurobi Optimizer 10.0 [44]. All runs were performed on a single thread on an Apple M1. The experiments were conducted mainly with networks with the size of $nr = 32{,}768$ terminals, a typical size of the Google Jupiter DCN, where 2048 ToR switches each have 16 uplinks [13,45].

A. Impact of Non-uniformity

This subsection investigates the impact of non-uniformity on the computation time of the BpIlp algorithm.

1. Evolving Networks with Non-uniform Switch Radixes

We consider DCN expansion by configuring each stage with switches of two different sizes. We use Fig. 13 to help understand the structure of non-uniform networks used in this evaluation. Instead of using uniform-radix switches as depicted in Fig. 13(a), we use larger ones (doubled original size) only in part, making the network non-uniform, as shown in Figs. 13(b) and 13(c). We refer to the proportion of the network using the doubled switches as the *coverage of doubled switches*; e.g., Fig. 13(b) illustrates a network where doubled switches cover 25% of the network, while Fig. 13(c) shows a network with 50% coverage. Introducing large switches does not change the total number of terminals and links in a network.

We first describe how to configure a uniform network like Fig. 13(a), which is used as a baseline. As discussed in Section 3.C, the computation time of BpIlp depends on r_1, r_3, and m. To focus on the impact of non-uniformity, we do not change r or m, where $r = r_1 = r_3$ assuming symmetric networks. To analyze a Google Jupiter scale network with $nr = 32{,}768$, and a typical switch size of up to a thousand ports, we use the following parameter sets: $(n, r, v, m) = (256, 64, 32, 16)$, $(512, 64, 64, 16)$, or $(1024, 64, 128, 16)$; note that a symmetric regular network is determined by four parameters $\langle n, r, v, m\rangle$, as noted in Section 2.A. These networks are configured to be nonblocking according to the known nonblocking condition for symmetric regular networks, Eq. (14). Note that we restrict r and m to be an even number to allow for expansion through replacement with *doubled* switches.

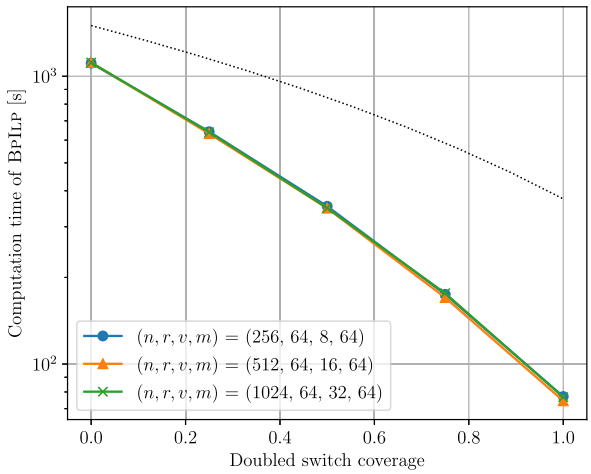


Fig. 14. Computation times of the BpIlp algorithm versus the doubled switch coverage in a network. When the coverage is zero, the structural parameters are $(n, r, v, m) = (256, 64, 32, 16)$, $(512, 64, 64, 16)$, and $(1024, 64, 128, 16)$. The dotted line indicates the slope of r^2.

Figure 14 plots the computation time of BpIlp against doubled switch coverage. As coverage increases, the numbers of switches, r and m, decrease (Fig. 9); they are halved at 1.0. We observe that the computation time of BpIlp decreases faster than r^2, where the dotted line indicates the slope of r^2 (the slope is arbitrarily shifted for comparison with the solid lines). This is because BpIlp loops r^2 times for a nonblocking network, but the ILP time in each loop also decreases with r and m, as will be elaborated below. As observed from Fig. 14, the computation time remains unchanged even with different values of n and v. This accords with the discussion in Section 3.C that the computation time of BpIlp is determined by r and m. Although networks have a non-uniform structure in the intermediate region of the horizontal axis, there is no bump in computation time associated with the non-uniformity. This indicates that networks whose stages include two different-sized switches incur no significant penalty with regard to the computation time of BpIlp. Even for Jupiter-scale (large) networks, BpIlp finishes within 19 min at most. This is a practical time frame in the network design and inspection phases, as mentioned in Section 2.C.

Figure 15 presents the computation times of individual ILPs evaluated during the measurements in Fig. 14. The ILP has $m(r_1 + r_3) - 1$ integer decision variables (x's) and m binary decision variables (y's), as described in Section 3.C. As the coverage of doubled switches reaches 1.0, both r and m are halved, reducing the number of decision variables to roughly 1/4. As shown in Fig. 15, the ILP computation time decreases as the doubled switch coverage increases. This decrease explains why the computation time of BpIlp decreases faster than r^2 in Fig. 14.

In addition to Fig. 14, we introduce another measurement, Fig. 16, where only the first and third stages or only the second stage are expanded with doubled switches while the remaining stages use small switches. The computation times are normalized by that of the baseline network where all stages consist of small switches (i.e., the doubled switch coverage is zero in

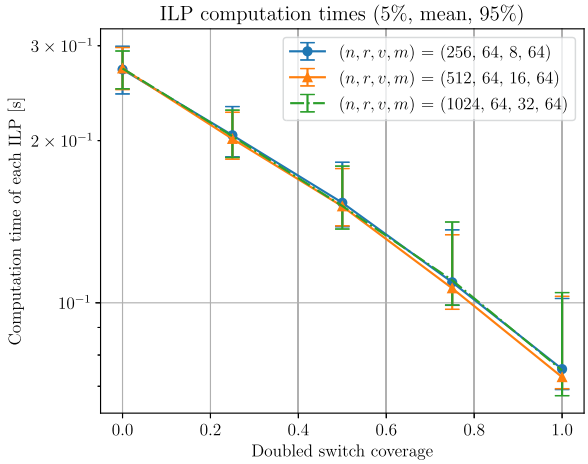


Fig. 15. Computation times of each ILP evaluated in Fig. 14. The 5%, mean, and 95% times are shown.

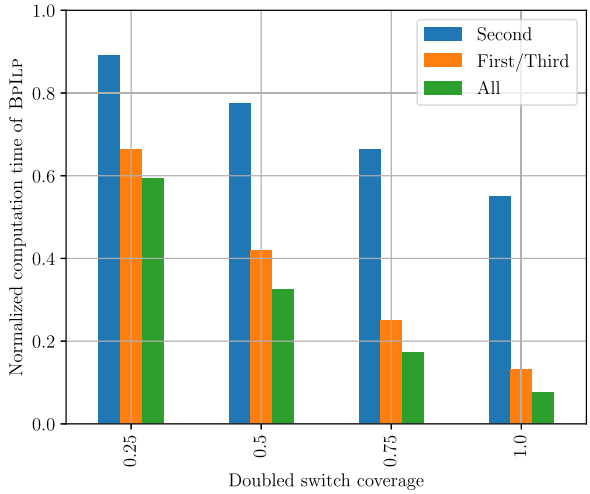


Fig. 16. Normalized computation time of the BpIlp algorithm for the network evolution scenarios. The computation time is normalized by that of the baseline network where all stages consist of small switches. The parameters in the baseline network are set to $(n, r, v, m) = (512, 64, 16, 64)$.

Fig. 14). The parameters in the baseline network are set to $(n, r, v, m) = (512, 64, 16, 64)$. The "All" scenario in Fig. 16 corresponds to the case where the doubled switch coverage is one in Fig. 14. Adding doubled switches only to the first/third stages significantly reduces the computation time, approaching the All scenario. This is because the first/third scenario reduces the number of first/third-stage switches, r, which quadratically reduces the loops in BpIlp. In contrast, adding doubled switches only to the second stage does not greatly reduce the computation time. Although this scenario reduces the number m of second-stage switches, it does not reduce the loops at all.

2. Failures of Terminals and Links

We consider networks whose uniformity can be lost due to terminal and link failures; although terminals are regarded as control parameters in Section 5.A, they randomly fail to

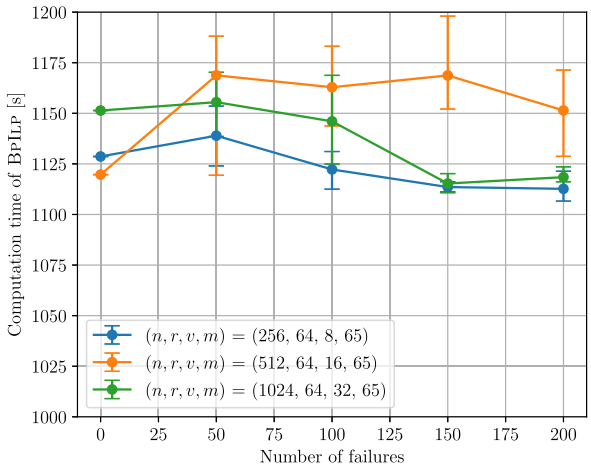


Fig. 17. Computation times of the BpIlp algorithm versus the number of failed terminals and links. When the failure number is zero, the structural parameters are $(n, r, v, m) = (256, 64, 32, 16)$, $(512, 64, 64, 16)$, and $(1024, 64, 128, 16)$. The minimum, the mean, and the maximum computation times are shown for eight trials.

further skew the network structure in this section. Similar to Section 6.A.1, we use a symmetric and regular network as a baseline. To ensure nonblockingness under link failures, we choose a slightly larger number of second-stage switches, i.e., $m = 2\lfloor (n-1)/v \rfloor + 3$. The experiments were conducted with the parameter sets of Section 6.A.1. In the experiments, terminals and links randomly fail up to the specified number of failed components (terminals and links are chosen without distinction), and the blocking property of the network is evaluated. For each number of failed components, we conducted eight trials and plotted the minimum, average, and maximum of the computation times.

Figure 17 plots the computation time of BpIlp against the number of failures. We randomly remove up to 200 terminals and links; removing more links could make the networks blocking. Similar to Section 6.A.1, the computation time remains unchanged for different values of n and v; the fluctuation is within 8%. The computation time matches the case of no doubled switches in Fig. 14, which has almost the same parameter set. Notably, the computation time is unaffected by the number of failures. Although the network structure becomes quite non-uniform under high failure numbers, we find no clear increase in the computation time.

B. Scalability

This subsection evaluates the scalability of the BpIlp algorithm. We use symmetric regular networks in this evaluation because Section 6.A finds little impact of non-uniformity on the computation time. In the following plots, given (n, r, v), the remaining m is determined using the known nonblocking condition, $m = 2\lfloor (n-1)/v \rfloor + 1$ [42].

Figure 18 plots the computation time of the BpIlp algorithm against network size nr. Figure 18(a) is plotted for $n = 512$, while Fig. 18(b) is plotted for $v = 64$. The computation time increases slightly faster than r^2 in all settings, where the dotted lines indicate the slope of r^2. This is because BpIlp

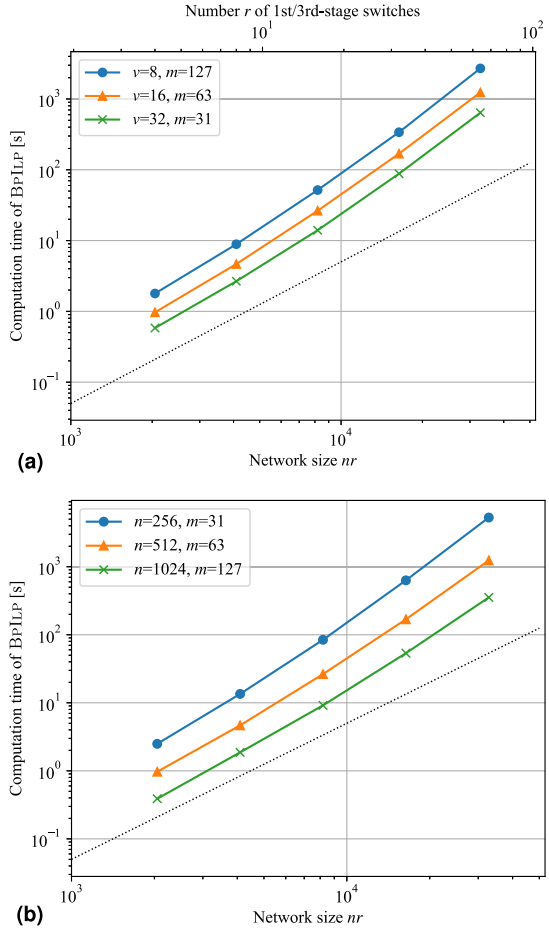


Fig. 18. Computation time of the BpIlp algorithm versus network size. The dotted lines indicate the slope of r^2. (a) $n = 512$ and (b) $v = 64$.

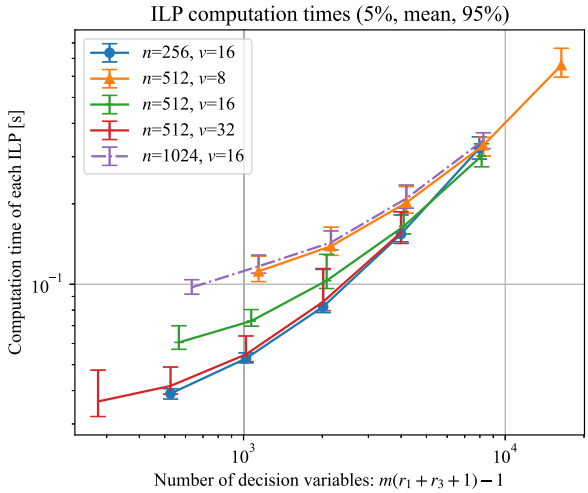


Fig. 19. Computation times for each ILP evaluated in Fig. 18. The 5%, mean, and 95% times are shown.

loops r^2 times for a nonblocking network, but each ILP time increases with network size. In Fig. 18(a), the computation time increases with m because the number of decision variables in an ILP increases with it (Section 3.C). In Fig. 18(b), the computation time is inversely correlated with n because r is inversely proportional to n for each nr (the blue line has larger r than the green line). Unlike Fig. 18(a), the computation time does not increase with m. This is because the number of decision variables also depends on r, and r has a stronger impact on the computation time than m.

Figure 19 presents the computation times for individual ILPs evaluated during the measurements in Fig. 18. The ILP has $m(r_1 + r_3) - 1$ integer decision variables (x's) and m binary decision variables (y's), and the total number of decision variables is plotted on the horizontal axis. Figure 19 shows that the computation time increases with the number of decision variables. This increase explains why the computation time of BpIlp increases faster than r^2 in Fig. 18.

C. Blocking Networks

This subsection evaluates the computation time for blocking networks. We modify symmetric-regular nonblocking networks to create blocking networks by removing switches or links. We consider three scenarios:

- (A) removal of a second-stage switch and its links, i.e., $m \leftarrow m - 1$ and a column is removed from V_1 and V_3;
- (B) removal of the first link between the first and second stages, i.e., $v_{1,11} \leftarrow v_{1,11} - 1$; and
- (C) removal of the last link between the first and second stages, i.e., $v_{1,r_1 m} \leftarrow v_{1,r_1 m} - 1$.

BpIlp sequentially examines each pair of first- and third-stage switches, as shown in Algorithm 1. Therefore, if the first pair is found to be blocking, as in (A) and (B), BpIlp would finish without looping. On the other hand, if only the final pair is found to be blocking, as in (C), BpIlp would loop almost as many times as in the nonblocking network.

Figure 20 plots the computation times for nonblocking networks in scenarios (A), (B), and (C). The computation time for the corresponding regular nonblocking network is shown for reference. Figure 20 presents results that are consistent with the above scenarios. For (A) and (B), BpIlp finishes in almost the same time as the single ILP evaluation shown in Fig. 19. For (C), the computation time is close to that of the corresponding nonblocking network.

7. RELATED WORKS

Section 7.A discusses past studies on nonblocking conditions. Section 7.B reviews the literature related to OCS DCNs and their feasibility in terms of insertion loss.

A. Nonblocking Conditions

Nonblocking conditions have been studied extensively since Clos presented the nonblocking condition for a three-stage switching network with symmetric regular structures in 1953 [35–37], e.g., assuming asymmetric network structures [16,17,24], multirate or asynchronous-transfer-mode networks [46,47], optical WDM networks [23,48–51], and elastic optical networks [22,25,52,53]. From the perspective

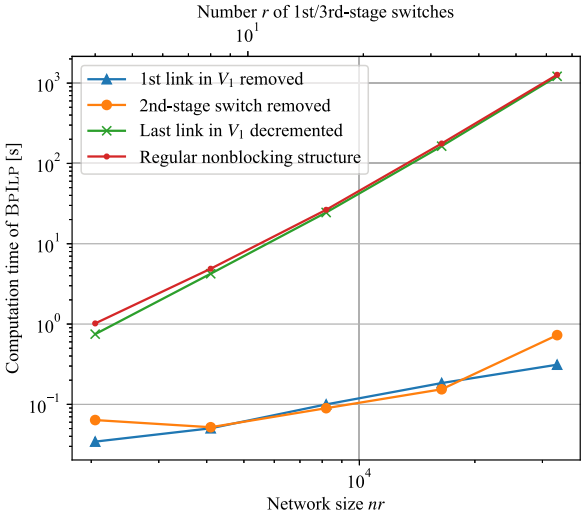


Fig. 20. Computation times of the BpIlp algorithm for blocking networks with $n = 512$, $v = 16$, and $m = 63$.

of network structure, several studies show nonblocking conditions for some "general" structures [16–20]. However, as discussed in Section 1, no existing work presents necessary and sufficient nonblocking conditions for the general structure. Although [18] presents the nonblocking condition for partially uniform structures, the condition is incorrect, as described in Section 4.B. References [19,20] study nonblocking conditions for the general structure but only show sufficient conditions.

There are several classes of nonblocking properties, e.g., rearrangeable, wide-sense, and strict-sense nonblocking [35–37]. A rearrangeable nonblocking network can always establish a connection by rearranging established connections, but connection rearrangement interrupts established connections. A wide-sense nonblocking network can always establish a connection under a specific path-selection policy, but a network without specifying a path-selection policy is more manageable for network operators. The SNB condition provides a theoretical foundation for switching networks including the Clos fabric [51] and has widely influenced their design [54]. It is still being extensively studied as evidenced by the recent literature [22–25,50,52,53]. In addition, the SNB condition is used as a bound between blocking and nonblocking network structures; e.g., [9,55,56] employ the SNB condition as a subproblem in designing a Clos fabric that must guarantee an admissible blocking probability. In this paper, we focus on SNB conditions without relying on connection rearrangement or any specific path-selection policy.

For the sake of clarity, we have so far described the main issue of our study as non-uniform switch radixes in a stage. However, even Clos fabrics with uniform switch radixes at every stage, but with non-uniform distribution of inter-switch links, have prevented past studies from evaluating their blocking properties. Reference [57] indicates that expanding a Clos fabric to avoid blocking and rearrangement of active connections breaks the uniformity of inter-switch links. The general nonblocking condition proposed in this paper can evaluate the blocking property under non-uniform inter-switch links as well (e.g., $N' = N$ in Fig. 11).

The book [58] delves into interconnection networks, including Clos fabrics, and examines reliability modeling methods. While blocking affects short-term path establishment success, reliability impacts long-term availability. These two factors are essential for building efficient and robust DCNs.

B. Optical Datacenter Networks

The saturation of Moore's law has led to a decline in the energy efficiency and performance of DCNs [3]. For instance, Meta's DC power usage saw a fourfold increase from 2016 to 2020, reaching 7.2 TWh in 2020 [59]. In a similar vein, the power consumption of Google's DCs rose 22% per year from 2017 to 2021, hitting 18.6 TWh in 2021 [60]. Networking, which recently accounted for about 5% of the total power consumption in DCs [61], is projected to constitute 27% by 2029 [62]. From a performance standpoint, the yearly growth in bisection bandwidth at Google's DC was approximately 125% from 2004 to 2012, but this rate decreased to around 26% post-2012 [45]. Optical switching is recognized as a promising solution to overcome the challenges of stagnant energy efficiency and performance.

Traditional DCNs relying on EPSs frequently experience significant latency. The 99th percentile of latency in packet switching can be two to three orders of magnitude greater than the median latency, primarily due to queueing and retransmission delays [8]. Moreover, average packet drop rates in DCNs could be high; e.g., a study by Microsoft [63] reported packet drop rates within or between pods in their DCNs to be in the range of 10^{-4} to 10^{-5}. However, recent advancements in AI training systems, including large language models (LLMs), demand not only high-capacity networks but also lossless and low-latency communication [5,30]. Circuit-switched networks, which establish dedicated circuits for connections, can meet these requirements without being affected by other connections, so networks proposed for AI training systems often adopt OCSs [5,30,31,33]. OCS DCNs for AI training have already been deployed by Google and Meta [5,33]. Currently, OCSs are integrated only in the spine stage, but they will be layered to form an entire Clos fabric, expanding the optical-circuit domain for better transmission bandwidth and energy efficiency [6,7]. The research community has been actively conducting studies focused on multi-stage OCS DCNs [8–11].

OCS technology encompasses a variety of devices such as robotic patch panels [64], MEMS [7,65], and silicon photonic switches [31]. Commercial robotic patch panels have 1008 ports at most, and MEMS switches with 1296 ports have been demonstrated [66,67]. These technologies are functionally similar from a network layer's perspective, as they all represent an input port signal connected to an output port without optical-electrical-optical (OEO) signal conversion or the decoding/buffering of in-flight packets. OCSs offer advantages over EPSs in terms of cost and power savings, transparency, and seamless upgrade due to their bandwidth-agnostic nature. To the best of our knowledge, no single OCS can accommodate the tens of thousands of terminals required to support an entire

large-scale DCN, such as Google Jupiter. Thus, in this paper, we studied Clos networks consisting of several OCSs.

We briefly discuss the feasibility of Clos fabrics with OCSs in terms of insertion loss. Consider OCSs with worst-case loss of 1.0 dB [64]. Operational signals would pass through at most three OCSs in the Clos fabric, resulting in a worst-case loss of 3.0 dB. Additionally, signals would traverse two long (e.g., 1 km) fibers in a building, incurring 0.7 dB loss, assuming a propagation loss of 0.35 dB/km. The total loss, 3.7 dB, is roughly one half of the loss budget of CWDM8 transceivers, 7.0 dB, a parameter often used in DCNs [5]. Thus, OCSs are feasible for forming a three-stage Clos fabric.

As noted in Section 1, DCNs are often composed of switches from different generations, resulting in a Clos fabric with non-uniform stages [12–14]. Inventory shortages due to supply chain damage may also prevent us from constructing uniform fabrics. In addition, Clos fabrics could be non-uniform due to failures of terminals and links. As described in Section 7.A, research issues in this field also include Clos fabrics with non-uniform inter-switch links [57]. For these factors, this paper studied the nonblocking condition for a Clos fabric with general structure.

8. CONCLUSIONS

This study presented the blocking condition for Clos fabrics whose stages involve non-uniform switch radixes. The condition is given in ILP form, and the associated algorithm named BpIlp evaluates the blocking property of a given network. The general condition revealed counter-examples to a known blocking condition defined for partially uniform Clos fabrics. We analyzed the nonblocking property under random link failures and during DCN evolution; we revealed that the nonblocking property of non-uniform networks could affect available network size. Numerical experiments demonstrated that BpIlp can evaluate a Jupiter-scale DCN with 32K terminals in the feasible 19 min.

In future works, we shall study more efficient algorithms that can evaluate the nonblockingness of partially uniform structures. Future work also includes exploring parallelization techniques for our algorithm to enhance computational efficiency and further reduce completion times. We will study more complex network models on the general structure, e.g., multirate connections where the flow rates could be represented using time slots or bandwidth, following past work [17,18].

REFERENCES

1. M. Noormohammadpour and C. S. Raghavendra, "Datacenter traffic control: understanding techniques and tradeoffs," IEEE Commun. Surv. Tutorials **20**, 1492–1525 (2017).
2. T. Hoefler, D. Roweth, K. Underwood, *et al.*, "Data center Ethernet and remote direct memory access: issues at hyperscale," Computer **56**, 67–77 (2023).
3. T. N. Theis and H.-S. P. Wong, "The end of Moore's law: a new beginning for information technology," Comput. Sci. Eng. **19**, 41–50 (2017).
4. L. Poutievski, O. Mashayekhi, J. Ong, *et al.*, "Jupiter evolving: transforming Google's datacenter network via optical circuit switches and software-defined networking," in *ACM SIGCOMM 2022 Conference* (2022), pp. 66–85.
5. H. Liu, R. Urata, K. Yasumura, *et al.*, "Lightwave fabrics: at-scale optical circuit switching for datacenter and machine learning systems," in *ACM SIGCOMM 2023 Conference* (2023), pp. 499–515.
6. X. Zhao, A. Vahdat, and H. Liu, "Implementation of a large-scale multi-stage non-blocking optical circuit switch," US patent 9,210,487 (11 March 2015).
7. R. Urata, H. Liu, K. Yasumura, *et al.*, "Mission Apollo: landing optical circuit switching at datacenter scale," arXiv (2022).
8. K.-I. Sato, "Optical switching will innovate intra data center networks [Invited Tutorial]," J. Opt. Commun. Netw. **16**, A1–A23 (2023).
9. H. Taka, T. Inoue, and E. Oki, "Design model of twisted and folded Clos network with multi-step grouped intermediate switches guaranteeing admissible blocking probability," J. Opt. Commun. Netw. **16**, 328–341 (2024).
10. K. Anazawa, T. Inoue, T. Mano, *et al.*, "Efficient fiber-inspection and certification method for optical-circuit-switched datacenter networks," J. Opt. Commun. Netw. **16**, 788–799 (2024).
11. W. Li, G. Yuan, C. Wu, *et al.*, "A reconfigurable optical network for distributed deep learning," in *Opto-Electronics and Communications Conference (OECC)* (IEEE, 2023).
12. J. C. Mogul and J. Wilkes, "Physical deployability matters," in *22nd ACM Workshop on Hot Topics in Networks* (2023), pp. 9–17.
13. S. Zhao, R. Wang, J. Zhou, *et al.*, "Minimal rewiring: efficient live expansion for Clos data center networks," in *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI)* (2019), pp. 221–234.
14. Y. Zhao, X. Zhang, H. Zhu, *et al.*, "Klotski: efficient and safe network migration of large production datacenters," in *ACM SIGCOMM 2023 Conference* (2023), pp. 783–797.
15. C. Clos, "A study of non-blocking switching networks," Bell Syst. Tech. J. **32**, 406–424 (1953).
16. M. Collier and T. Curran, "The strictly non-blocking condition for three-stage networks," in *Teletraffic Science and Engineering* (Elsevier, 1994), Vol. **1**, pp. 635–644.
17. W. Kabacinski, "Non-blocking asymmetrical three-stage multirate switching networks," in *International Conference on Communication Technology* (IEEE, 1998), Vol. **1**, pp. 59–63.
18. W. Kabacinski and F. K. Liotopoulos, "Multirate non-blocking generalized three-stage Clos switching networks," IEEE Trans. Commun. **50**, 1486–1494 (2002).
19. F. K. Liotopoulos and S. Chalasani, "Strictly nonblocking operation of 3-stage Clos switching networks," in *ATM Networks: Performance Modelling and Evaluation* (1996), Vol. **3**, pp. 269–286.
20. Y. Xuan and C.-T. Lea, "Discrete-bandwidth nonblocking networks," IEEE Trans. Commun. **61**, 4334–4342 (2013).
21. H. Q. Ngo, A. Le, and Y. Wang, "A linear programming duality approach to analyzing strictly nonblocking d-ary multilog networks under general crosstalk constraints," J. Comb. Optim. **21**, 108–123 (2011).
22. W. Kabaciński, M. Michalski, and R. Rajewski, "Strict-sense nonblocking WSW node architectures for elastic optical networks," J. Lightwave Technol. **34**, 3155–3162 (2016).
23. W. Kabaciński, J. Kleban, M. Michalski, *et al.*, "Strict-sense nonblocking networks with k degrees of freedom," Opt. Switch. Netw. **22**, 18–25 (2016).
24. G. Danilewicz, "Asymmetrical space-conversion-space SCS1 strict-sense and wide-sense nonblocking switching fabrics for continuous multislot connections," IEEE Access **7**, 107058 (2019).
25. W. Kabaciński, M. Michalski, and R. Rajewski, "Optimization of strict-sense nonblocking wavelength-space-wavelength elastic optical switching fabrics," Opt. Switch. Netw. **33**, 76–84 (2019).
26. H. Q. Ngo, A. Rudra, A. N. Le, *et al.*, "Analyzing nonblocking switching networks using linear programming (duality)," in *IEEE INFOCOM* (IEEE, 2010).
27. T. Mano, T. Inoue, K. Mizutani, *et al.*, "Increasing capacity of the Clos structure for optical switching networks," in *IEEE Global Communications Conference (GLOBECOM)* (IEEE, 2019).

28. T. Mano, T. Inoue, K. Mizutani, *et al.*, "Redesigning the nonblocking Clos network to increase its capacity," IEEE Trans. Netw. Serv. Manage. **20**, 2558–2574 (2023).
29. V. E. Beneš, *Mathematical Theory of Connecting Networks and Telephone Traffic* (Academic, 1965).
30. W. M. Mellette, R. Das, Y. Guo, *et al.*, "Expanding across time to deliver bandwidth efficiency and low latency," in *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI)* (2020).
31. M. Khani, M. Ghobadi, M. Alizadeh, *et al.*, "SIP-ML: high-bandwidth optical network interconnects for machine learning training," in *ACM SIGCOMM 2021 Conference* (2021), pp. 657–675.
32. P. Cao, S. Zhao, D. Zhang, *et al.*, "Threshold-based routing-topology co-design for optical data center," IEEE/ACM Trans. Netw. **31**, 2870–2885 (2023).
33. W. Wang, M. Khazraee, Z. Zhong, *et al.*, "TopoOpt: co-optimizing network topology and parallelization strategy for distributed training jobs," in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI)* (2023), pp. 739–767.
34. M. Y. Teh, S. Zhao, P. Cao, *et al.*, "Enabling quasi-static reconfigurable networks with robust topology engineering," IEEE/ACM Trans. Netw. **31**, 1056–1070 (2022).
35. F. K.-M. Hwang, *The Mathematical Theory Of Nonblocking Switching Networks* (World Scientific, 2004).
36. W. Kabacinski, *Nonblocking Electronic and Photonic Switching Fabrics* (Springer, 2005).
37. J. Y. Hui, *Switching and Traffic Theory for Integrated Broadband Networks* (Springer, 1990).
38. B. Korte and J. Vygen, "Network flows," in *Combinatorial Optimization: Theory and Algorithms*, 5th ed. (Springer, 2006), pp. 157–189.
39. R. W. Cottle, J.-S. Pang, and R. E. Stone, *The Linear Complementarity Problem* (SIAM, 2009).
40. C. Chekuri, S. Khanna, and F. B. Shepherd, "The all-or-nothing multicommodity flow problem," in *36th Annual ACM Symposium on Theory of Computing* (2004), pp. 156–165.
41. E. Balas, *Disjunctive Programming* (Springer, 2018).
42. A. Jajszczyk, "On nonblocking switching networks composed of digital symmetrical matrices," IEEE Trans. Commun. **31**, 2–9 (1983).
43. https://coin-or.github.io/pulp/.
44. https://www.gurobi.com/solutions/gurobi-optimizer/.
45. A. Singh, J. Ong, A. Agarwal, *et al.*, "Jupiter rising: a decade of Clos topologies and centralized control in Google's datacenter network," ACM SIGCOMM Comput. Commun. Rev. **45**, 183–197 (2015).
46. S. C. Liew, M.-H. Ng, and C. W. Chan, "Blocking and nonblocking multirate Clos switching networks," IEEE/ACM Trans. Netw. **6**, 307–318 (1998).
47. J. S. Turner and R. Melen, "Multirate Clos networks," IEEE Commun. Mag. **41**(10), 38–44 (2003).
48. A. Rasala and G. Wilfong, "Strictly nonblocking WDM cross-connects," SIAM J. Comput. **35**, 449–485 (2005).
49. H. Q. Ngo, D. Pan, and C. Qiao, "Constructions and analyses of nonblocking WDM switches based on arrayed waveguide grating and limited wavelength conversion," IEEE/ACM Trans. Netw. **14**, 205–217 (2006).
50. J. Lin, T. Chang, Z. Zhai, *et al.*, "Wavelength selective switch-based Clos network: blocking theory and performance analyses," J. Lightwave Technol. **40**, 5842–5853 (2022).
51. J. Lin, Z. Chang, L. Zong, *et al.*, "From small to large: Clos network for scaling all-optical switching," IEEE Commun. Mag. **61**(12), 136–141 (2023).
52. G. Danilewicz, W. Kabaciński, and R. Rajewski, "Strict-sense non-blocking space-wavelength-space switching fabrics for elastic optical network nodes," J. Opt. Commun. Netw. **8**, 745–756 (2016).
53. M. T. H. Al-Musawi, A. A. A. Wahab, S. S. N. Alhady, *et al.*, "The three-stage non-blocking switch for elastic optical networks," in *11th International Conference on Robotics, Vision, Signal Processing and Power Applications: Enhancing Research and Innovation through the Fourth Industrial Revolution* (Springer, 2022), pp. 234–240.
54. A. Jajszczyk, "Nonblocking, repackable, and rearrangeable Clos networks: fifty years of the theory evolution," IEEE Commun. Mag. **41**(10), 28–33 (2003).
55. H. Taka, T. Inoue, and E. Oki, "Design of twisted and folded Clos network with guaranteeing admissible blocking probability," IEEE Netw Lett. **5**, 265–269 (2023).
56. H. Taka, T. Inoue, and E. Oki, "Twisted and folded Clos-network design model with two-step blocking probability guarantee," IEEE Netw Lett. **6**, 2576–3156 (2023).
57. T. Inoue, T. Mano, and T. Uno, "Cost-effective live expansion of three-stage switching networks without blocking or connection rearrangement," in *IEEE INFOCOM Conference on Computer Communications* (IEEE, 2023).
58. N. K. Goyal and S. Rajkumar, *Interconnection Network Reliability Evaluation: Multistage Layouts* (Wiley, 2020).
59. M. Sahini and J. Athavale, "Preparing Meta for growing power demand: thermal perspective," in *Open Computer Summit* (2021).
60. Google, "Environmental report" (2022).
61. L. A. Barroso, U. Hölzle, and P. Ranganathan, *The Datacenter as a Computer: Designing Warehouse-Scale Machines* (Springer, 2019).
62. "2015 International Technology Roadmap for Semiconductors (ITRS)" (Semiconductor Industry Association, 2015).
63. C. Guo, L. Yuan, D. Xiang, *et al.*, "Pingmesh: a large-scale system for data center network latency measurement and analysis," in *ACM Conference on Special Interest Group on Data Communication* (2015), pp. 139–152.
64. A. S. Kewitsch, "Large scale, all-fiber optical cross-connect switches for automated patch-panels," J. Lightwave Technol. **27**, 3107–3115 (2009).
65. M. Stepanovsky, "A comparative review of MEMS-based optical cross-connects for all-optical networks from the past to the present day," IEEE Commun. Surv. Tutorials **21**, 2928–2946 (2019).
66. R. Ryf, J. Kim, J. Hickey, *et al.*, "1296-port MEMS transparent optical crossconnect with 2.07 petabit/s switch capacity," in *Optical Fiber Communication Conference (OFC)* (2001), paper PD28.
67. S. B. Yoo, "Prospects and challenges of photonic switching in data centers and computing systems," J. Lightwave Technol. **40**, 2214–2243 (2021).

Takeru Inoue is a Distinguished Researcher at Nippon Telegraph and Telephone Corporation (NTT) Laboratories, Japan. He received the B.E. and M.E. degrees in engineering science and the Ph.D. degree in information science from Kyoto University, Japan, in 1998, 2000, and 2006, respectively. In 2000, he joined NTT Laboratories. From 2011 to 2013, he was an ERATO Researcher with the Japan Science and Technology Agency, where his research focused on algorithms and data structures. Currently, his research interests widely cover the design and control of communication networks. He serves as an Associate Editor of the *IEEE Transactions on Network and Service Management*. He is a Senior Member of IEEE. Dr. Inoue has been the recipient of several prestigious awards, including the Best Paper Award of the Asia-Pacific Conference on Communications in 2005, the Best Paper Award of the IEEE International Conference on Communications in 2016, the Best Paper Award of the IEEE Global Communications Conference in 2017 and 2023, the Best Paper Award of the IEEE Reliability Society Japan Joint Chapter in 2020, the IEEE Asia/Pacific Board Outstanding Paper Award in 2020, and the IEICE Paper of the Year in 2021, and was selected as one of the best papers at the European Conference on Optical Communication in 2023.

Toru Mano received the B.E. and M.E. degrees from the University of Tokyo in 2009 and 2011, respectively, and the Ph.D. degree in computer science and information technology from Hokkaido University in 2020. He joined Nippon Telegraph and Telephone (NTT) Network Innovation Laboratories, Japan, in 2011, where he is a Senior Research Engineer. His research interests are network architectures, network optimization, and softwarization of networking. He was a recipient of the IEICE Paper of the Year in 2021 and was selected as one of the best papers at the European Conference on Optical

Communications 2023. He is a member of IEICE and the Operations Research Society of Japan.

Kazuya Anazawa is a researcher at Nippon Telephone and Telegraph (NTT) Corporation Network Innovation Laboratories, Japan. He received his B.E. (2016) and M.E. (2018) degrees in computer science and engineering from the University of Aizu. In 2018, he joined NTT Network Innovation Laboratories. His research interests include optical network design and autonomous control of optical networks.

Takeaki Uno is a Professor at the National Institute of Informatics, Japan. He received the Ph.D. degree (Doctor of Science) from the Department of Systems Science, Tokyo Institute of Technology, Japan, 1998. He was an assistant professor in the Department of Industrial and Management Science at the Tokyo Institute of Technology from 1998 to 2001. His research topic is discrete algorithms, especially enumeration algorithms, algorithms on graph classes, and data mining algorithms. He received the Young Scientists' Prize of The Commendation for Science and Technology in Japan in 2010.