DOI: http://doi.org/10.25728/cs.2021.5.6

NON-BLOCKING FAULT-TOLERANT DUAL PHOTON SWITCHES WITH HIGH SCALABILITY

V.S. Podlazov

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

✉ podlazov@ipu.ru

Abstract. This paper continues the construction of a fundamentally new class of system area networks (dual photon networks) with the following features: non-blocking property and static self-routing, high scalability with the maximum achievable speed and a small complexity compared to a full switch, and balancing the scalability-speed and complexity-speed ratios. These networks are implemented in an extended circuit basis consisting of dual photon switches and separate photon multiplexers and demultiplexers. We propose a method for constructing a fault-tolerant dual network with the indicated properties based on networks with the quasi-complete graph and quasi-complete digraph topologies and the invariant extension method with internal parallelization. Also, we propose a method for extending the two-stage dual network designed previously into four-stage and eight-stage dual networks with high scalability while maintaining the original network period and reducing its exponential complexity.

Keywords: photon switch, dual switch, photon multiplexers and demultiplexers, multistage switch, conflict-free self-routing, non-blocking switch, static self-routing, quasi-complete digraph, quasi-complete graph, invariant extension of networks, switching properties, direct channels, scalability and speed.

INTRODUCTION

This paper develops a method for constructing a fundamentally new class of system area networks [1–6], the so-called dual photon networks. They are non-blocking networks with static self-routing [1–3, 5] and can have a given degree of channel fault tolerance [6].

In what follows, we propose a method for constructing non-blocking self-routing photon networks with high scalability. These are dual networks based on a non-blocking dual $p \times p$ switch with a signal period of p cycles [1–3]. For resolving signal conflicts, the dual switch combines the bus method (separation of conflicting signals to different cycles in one channel) and the switch method (separation of conflicting signals to different channels). The dual switch is a non-blocking switch on any input traffic if data bits are transmitted with a signal period of p cycles. The dual switch was developed by the author's colleagues [1, 2] and then applied and named in the joint publications [3–5]. This switch turned out to be a prerequisite for constructing non-blocking networks with high scalability and acceptable complexity.

The dual photon switch transmits signal and control information in parallel at different frequencies for each data bit. This method eliminates the problem of synchronizing signals from different channels.

The photon specifics of such networks consist in using in-bit channel virtualization with feedback links through delay lines with a duration of one cycle and control signals at different frequencies to route individual bits. The separation of information signals to different cycles is accompanied by the separation of the corresponding control information to the same cycles. It is used to route the bits by moving them between different channels without changing the established cycle numbers.

Throughout the paper, the terms "dual switch" and "dual network based on dual switches" imply using bits with a period of p cycles in them. These bits en-

sure the non-blocking property in the first stage of the network; in the other stages, they remain by "inertia" without using the bus method of conflict resolution.

The scalability of dual networks is provided using networks with the quasi-complete graph or digraph topology [4], which are implemented in an extended circuit basis consisting of dual photon switches and separate photon multiplexers and demultiplexers. In [1–3, 5], high scalability was achieved using the invariant extension method of system area networks with many additional demultiplexers and multiplexers.

In the papers [5, 6], a new method for extending dual networks by their internal parallelization without additional devices was developed and applied for the first time. Particularly in [5], a two-stage non-blocking network was constructed. It consists of networks with the quasi-complete digraph topology with $N = p^2$ channels on each stage, whereas the two-stage non-blocking network has N^2 channels in total. On the other hand, a two-stage non-blocking network with $(\sigma - 1)$-channel fault tolerance was constructed in [6]. It consists of networks with the quasi-complete graph topology with $N = p(p-1)/\sigma + 1$ channels, whereas the two-stage fault-tolerant non-blocking network has N^2 channels.

Below, we construct four-stage and eight-stage fault-tolerant networks by developing and applying the generalized method of internal parallelization. In this case, the same degree of network scalability is achieved as in the invariant method, but without using external demultiplexers and multiplexers, and the resulting networks have a significantly smaller complexity.

This paper is organized as follows. Section 1 briefly considers the non-blocking property and channel fault tolerance in modern system area networks. In Section 2, following [6], we introduce the notions of *p*-permutations (crucial for proving the non-blocking property of four- and eight-stage networks) and repeat the proofs of the non-blocking property for two-stage networks. Section 3 presents four-stage non-blocking self-routing switches with one-channel and two-channel fault tolerance and their performance characteristics. The method of internal parallelization from [5, 6] is generalized for four-stage switches.

Section 4 compares the performance characteristics of four-stage non-blocking self-routing switches based on switches with the dual quasi-complete graph and digraph topologies. Finally, in Section 5, we construct eight-stage non-blocking self-routing switches based on switches with the dual quasi-complete graph and digraph topologies. Moreover, the method from Section 3 is generalized for eight-stage switches. Section 6 discusses the properties of these networks compared to other non-blocking networks (particularly their disadvantages and possible ways to overcome them).

In the Conclusions, we analyze the generalized method of internal parallelization, which is the core for constructing dual non-blocking networks with high scalability and low specific complexity. There are three main components of the proposed methodology: a non-blocking dual switch, a switch with the quasi-complete graph or digraph topology based on a dual switch, and the method of internal parallelization.

1. NON-BLOCKING PROPERTY AND FAULT TOLERANCE IN SYSTEM AREA NETWORKS

The problem of constructing non-blocking fault-tolerant system area networks of supercomputers has not been completed solved so far.

A system area network is non-blocking if for any packet permutation, conflict-free paths from sources to sinks can be built in it. A system area network is self-routing if conflict-free paths can be built locally over network nodes without their interaction based on routing information in packets only. Finally, self-routing is static if any source can independently choose conflict-free paths to its sink without interacting with other sources.

The existence of non-blocking networks was proved by Clos [7, 8]. Self-routing procedures for non-blocking Clos networks have not yet been developed. However, these networks can be a qualitative measure for other non-blocking networks.

A network in the form of a two-dimensional generalized hypercube with the quasi-complete digraph topology is non-blocking, e.g., in the YARK and ROSETTA switches used in several networks of different structure: a reconfigurable Clos network [9], a three-dimensional torus [10], and a hierarchy of complete and quasi-complete digraphs [11–13]. Unfortunately, a quasi-complete digraph has a small number of channels $N = p^2$, where p is the degree of internal switches, and a high switching complexity $S \geq N^2$, exceeding the complexities of a complete digraph and a non-blocking Clos network (in the latter case, considerably).

Modern literature widely describes system area networks with the fat tree structure (particularly reconfigurable Clos networks), the generalized hypercube structure, the multidimensional torus structure, and system area networks with a hierarchy of complete and quasi-complete digraphs.

Fat-tree networks are reconfigurable networks [9, 14, 15] with conflict-free transmission only according to predetermined schedules for specific packet permutations. In the case of arbitrary permutations, these networks turn out to be blocking; permutations in them are implemented in several jumps between network nodes. The maximum number of such jumps determines the network diameter. In reconfigurable Close networks, the diameter equals the number of network stages.

Networks with the generalized hypercube structure [16–19] are not even reconfigurable [20, 21]. They can be made such by increasing the number of channels in some dimensions. Generalized cubes have a diameter equal to the number of dimensions or less by 1 in the extended hypercube [17, 18]. Generalized hypercubes with a doubled number of channels in each dimension are reconfigurable networks for two permutations simultaneously. Note that an attempt to use a generalized hypercube as a non-blocking network for a photon computer [22] seems to be a very dubious venture.

For arbitrary permutations, networks with the multidimensional torus structure cannot transmit packets over direct channels [11, 23–25]. They implement permutations in several jumps between network nodes. Multidimensional tori are the simplest, albeit slowest, networks due to their large diameters. For example, the networks considered in [11, 23–25] have diameters measured in tens of jumps.

On the contrary, networks with a hierarchy of complete or quasi-complete digraphs [10, 12, 26] have the smallest diameter of three jumps. Many networks with small diameters have appeared recently [27–32]. All of them have serious problems with balancing network load under channel faults.

Channel fault tolerance is the network's ability to preserve full availability under channel failures while maintaining its original performance characteristics (the non-blocking property, transmission delays, or network diameter).

Apparently, only networks with the quasi-complete graph topology possess channel fault-tolerance in pure form. These networks are isomorphic to such a mathematical object as an incomplete balanced symmetric block design [33]. These networks have an element base of $p\times p$ switches, $1\times p$ demultiplexers, and $p\times 1$ multiplexers and are non-blocking networks with static self-routing. They have direct channels between $N = p(p-1)/\sigma + 1$ network users and σ different channels between any two users [4].

In other networks, the restoration of full network availability under channel faults is accompanied in one way or another by an increase in network transmission delays. For example, under channel faults in a reconfigurable Clos network, the load on the remaining channels grows, increasing the number of conflicts and delays in the transmission of some packets.

The TOFY network with the three-dimensional torus structure [25] uses three more dimensions to create redundant channels. If some network rings fail, their integrity is restored by increasing the network diameter by 1.

Generalized hypercubes with doubled channels in each dimension are one-fault-tolerant networks with a constant diameter [19].

In networks with a hierarchy of complete or quasi-complete digraphs [10, 12, 26], if some of the channels fail, the network's full availability is restored using bypass paths with a duration of five jumps. In other words, the transmission delay increases by a factor of 5/3.

2. DUAL QUASI-COMPLETE GRAPH, *A*PERMUTATIONS, AND DUAL TWO-STAGE SWITCH

The dual switch SFN_1 with the quasi-complete graph topology, SQG(N_1, p, σ), consists of $N_1 = p(p-1)/\sigma + 1$ dual $p\times p$ switches SSp, N_1 input $1\times p$ demultiplexers, and N_1 output $p\times 1$ multiplexers [6]. They are interconnected using the combinatorial method [4]: there are σ different paths through different dual switches SSp. Figure 1 shows the circuit of an SF4 switch as an SQG (4, 3, 2) graph with one-channel fault tolerance. Two paths are highlighted, connecting two randomly selected inputs and outputs, (2, 4) and (3, 3).

Any dual switch SFN_1 has the same signal period T_1 as the dual switch SSp included in it. For the switch SFN_1, the following performance characteristics are calculated: the switching complexity S_1, expressed in the number of switching points, and the channel complexity, expressed in the number of channels. They are

written exponentially through the number of channels and are called exponential complexities[1]; see Table 1.

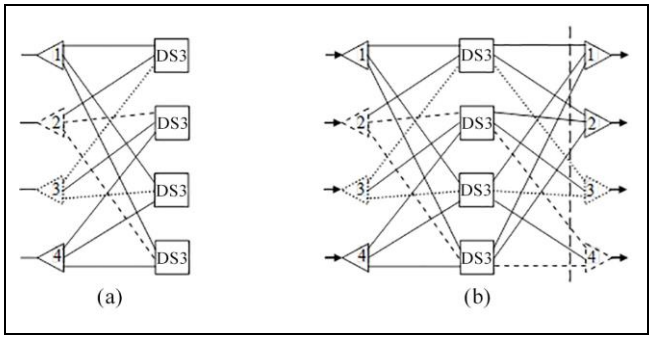

Fig. 1. Dual quasi-complete switch SF4 with signal period of three cycles represented by graph SQG(4, 3, 2): (a) original form with duplex channels, (b) application with simplex channels. Dashed lines and dots indicate different paths between selected inputs and outputs.

Table 1

Performance characteristics of dual switches SFN_1 with one-channel fault tolerance

p	N_1	T_1	S_1	L_1
2	2	2	$24 = N_1^{4.58}$	$8 = N_1^{3}$
4	7	4	$280 = N_1^{2.9}$	$56 = N_1^{2.07}$
6	15	6	$1\,260 = N_1^{2.64}$	$180 = N_1^{1.92}$
8	27	8	$3\,888 = N_1^{2.51}$	$432 = N_1^{1.84}$

For the dual switch SFN_1 with the quasi-complete *digraph* topology [5], the performance characteristics are given in Table 2. They are better than their counterparts from Table 1 but without channel fault tolerance.

Table 2

Performance characteristics of dual switches SFN_1 based on the quasi-complete digraph topology

p	N_1	T_1	S_1	L_1
2	4	2	$48 = N_1^{2.79}$	$16 = N_1^{2}$
4	16	4	$640 = N_1^{2.33}$	$128 = N_1^{1.75}$
6	36	6	$3\,024 = N_1^{2.24}$	$432 = N_1^{1.69}$
8	64	8	$9\,216 = N_1^{2.19}$	$1\,024 = N_1^{1.67}$

We introduce the notion of p-partitions of packets transmitted through some cross-section of a network at the multiplexer inputs. All packets are divided into groups of variable composition, each containing at most p packets. For a common permutation of packets, a 1-partition occurs at the input and output of the switch. A transmission in which a 1-partition occurs at the network input and a p-partition on a given cross-section will be called a p-permutation.

For the dual switch SQG(N_1, p, σ), this cross-section is through the inputs of the output multiplexers and is called the output cross-section. In Fig. 1, the output cross-section is indicated by the vertical dashed line. According to the property of the dual switch DSp, a p-partition occurs on the output cross-section of the dual switch SQG(N_1, p, σ) for any traffic.

Lemma 1 [6]. *The dual switch* SFN_1 *with a signal period of p cycles is a non-blocking switch with static self-routing under any common permutation and has* (σ – 1)*-channel fault-tolerance.*

The papers [5, 6] developed a method for constructing two-stage non-blocking switches with N_1^2 channels. Consider this method on an example of the dual switch SF2 with the dual quasi-complete graph topology SQG (2, 2, 2) (Fig. 2).

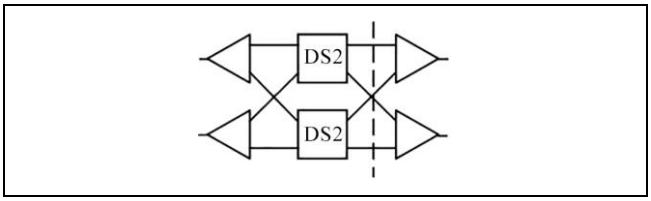

Fig. 2. Dual non-blocking switch SF2 with one-channel fault tolerance.

On its basis, a four-channel two-stage network $N_2$4 is constructed. This network contains two SF2 switches on each stage, connected by exchange links (Fig. 3). The network is blocking on multiplexers of the first stage, highlighted in grey, and loses channel fault tolerance on them.

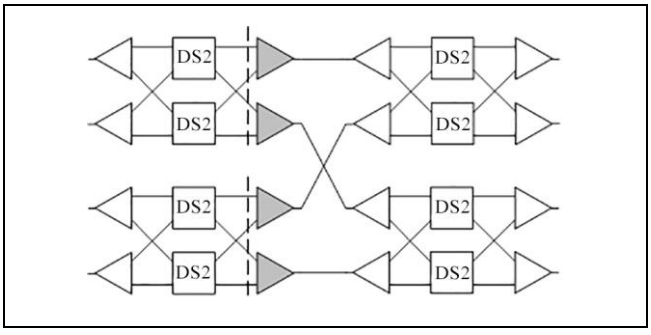

Fig. 3. Dual two-stage blocking network $N_2$4 with exchange links.

[1] This term was introduced by the author.

The network $N_2 4$ can be transformed into a non-blocking switch $S_2 4$ with one-channel fault tolerance by internal parallelization [5, 6]. The second stage of this switch uses two copies of the second stage of the $N_2 4$ network. On the first stage, multiplexers on the cross-sections are eliminated, and their inputs are connected to the inputs of the second stage copies: odd to the first copy and even to the second copy. These links preserve the order of connecting the channels located on the second stage in the network $N_2 4$. The cut-out multiplexers are moved to connect the outputs of the second stage copies, forming the output multiplexers of the switch $S_2 4$ (Fig. 4), which becomes a non-blocking switch with static self-routing and one-channel fault tolerance.

In the general case ($p \geq 2$), the dual switch $SF N_1$ has the dual quasi-complete graph topology $SQG(N_1, p, \sigma)$ with a signal period of p cycles. On its basis, a two-stage blocking network $N_2 N_2$ ($N_2 = N_1^2$) is constructed, where each stage contains N_1 switches $SF N_1$ with exchange links between the stages. For internal parallelization, p copies of the second stage of the network $N_2 N_2$ are formed, and the first stage multiplexers are used to combine the same-name outputs of the second stage copies.

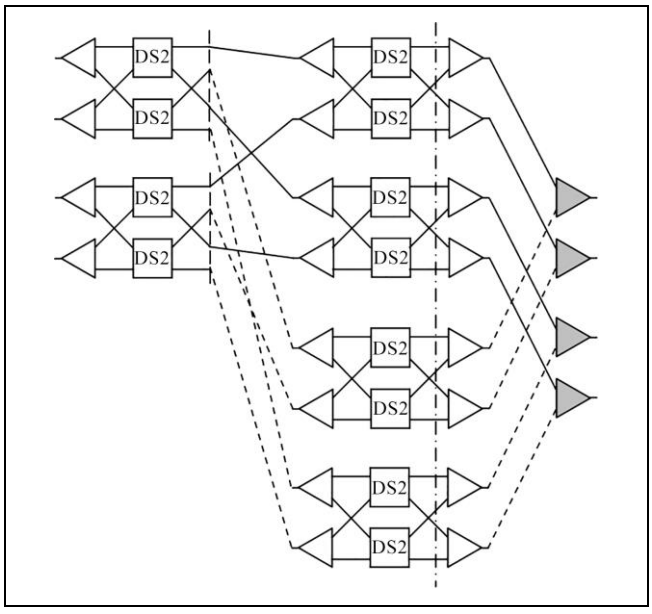

Fig. 4. Dual non-blocking self-routing switch $S_2 4$ with one-channel fault tolerance.

On the cross-sections of the first stage, there are pN_2 inputs to the multiplexers. They are renumbered top-to-bottom by I ($1 \leq I \leq pN_2$), and the inputs $i = I(\mathrm{mod} p)+1$ are connected to the same-name inputs of the ith copy of the second stage, maintaining the same arrangement of the switches $S_1 N_1$ as in the network $N_2 N_2$.

Lemma 2. *The dual switch $S_2 N_2$ has a p-permutation on the indicated cross-section. It is a non-blocking switch with static routing on any common permutation and has $(\sigma - 1)$-channel fault tolerance.*

The switching and channel complexities of the switch $S_2 N_2$ are given by the recursive formulas $S_2 = N_1 S_1 + pN_1 S_1$ and $L_2 = N_1 L_1 + pN_1 L_1$, respectively. The performance characteristics of the switches $S_2 N_2$ for $\sigma = 2$ are presented in Table 3. Note that the exponential complexities decrease compared to Table 1.

Table 3

Performance characteristics of dual switches $S_2 N_2$ with one-channel fault tolerance

p	N_1	$N_2 = N_1^2$	$T_2 = p$	S_2	L_2
2	2	4	2	$N_2^{3.58}$	$N_2^{2.9}$
4	7	49	4	$N_2^{2.37}$	$N_2^{1.97}$
6	15	225	6	$N_2^{2.18}$	$N_2^{1.84}$
8	27	729	8	$N_2^{2.09}$	$N_2^{1.77}$

The performance characteristics of a dual switch $S_2 N_2$ based on the quasi-complete digraph topology [5] are combined in Table 4. They are significantly better than in Table 2, but without channel fault tolerance.

Table 4

Performance characteristics of switches $S_2 N_2$ based on the quasi-complete digraph topology

p	N_1	$N_2 = N_1^2$	$T_2 = p$	S_2	L_2
2	4	16	2	$N_2^{2.29}$	$N_2^{1.95}$
4	16	256	4	$N_2^{1.96}$	$N_2^{1.68}$
6	36	1 296	6	$N_2^{1.89}$	$N_2^{1.63}$
8	64	4 096	8	$N_2^{1.86}$	$N_2^{1.6}$

Also, note that the dual switch $S_2 N_2$ has two stages of output multiplexers containing pN_2 and N_2 multiplexers, respectively. For the purposes of Section 3, we cut the switch $S_2 N_2$ through the inputs of the first stage of multiplexers; see the dash-and-dot line in Fig. 4.

3. FOUR-STAGE FAULT-TOLERANT NON-BLOCKING SELF-ROUTING SWITCH WITH TWO-DIMENSIONAL INTERNAL PARALLELIZATION

In the papers [1–3, 5], the number of channels of a non-blocking switch was further increased using the invariant extension method with external parallelization. This method does not change the signal period. However, it is of little use for fault-tolerant non-blocking switches [6].

In this section, we increase the number of channels without changing the signal period using the generalized method of internal parallelization of the network by constructing four-stage switches S_4N_4 from two-stage switches S_2N_2 with the number of channels $N_4 = N_2^2$ and the signal period $T_4 = T_2 = p$.

The network is constructed using switches $S_2$4 as an example (Fig. 4). First, the two-stage network $N_4$16 is created. Each stage in this network consists of four copies of the $S_2$4 switch, and the stages are interconnected by exchange links (Fig. 5).

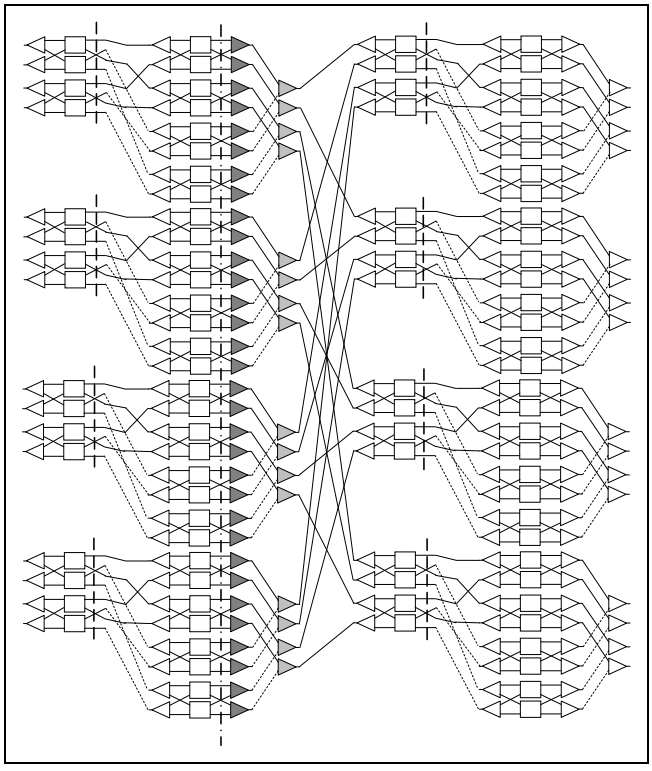

Fig. 5. Blocking dual network $N_4$16 without channel fault tolerance.

In reality, the network $N_4$16 consists of four stages SQG(2, 2, 2), which explains the subscript in the notation. The $N_4$16 network is blocking due to possible signal conflicts on the two stages of the output multiplexers M2 (highlighted in grey). There are two layers of such multiplexers, $W_4 = 48$ in total. In addition, channel fault tolerance is violated on them. The dash-and-dot cut is made through the inputs to the first stage (Fig. 5). For this cut, the notion of a *p*-permutation has been formulated in Section 2.

Then the network $\underline{N}_4$16 is created. It contains the first stage of the network $N_4$16 and two copies of the second stage of the network $N_4$16. One parallel circuit of the first dimension is created in the network $\underline{N}_4$16 (Fig. 6). For this, the outer layer highlighted in pale grey is first cut out with $W_2^* = 16$ multiplexers M2 in total. They remain unconnected for now. Then the multiplexers M2 of the inner layer highlighted in dark grey are cut out, and their odd inputs are routed to the inputs of two second-stage copies of the network $\underline{N}_4$16. In this case, the $W_{4,1} = 16$ cut multiplexers combine the outputs of these two copies.

The remaining $W_{4,2} = 16$ multiplexers M2 highlighted in dark grey are used to create the second parallel circuit of the first dimension in the same way (Fig. 7). Their even inputs are routed to the inputs of two additional copies of the second stage of the network $N_4$16.

Looking ahead, note that Figs. 6 and 7 show the new connections of the multiplexers of the first and second layers. They define the combination of the first dimension circuits into the second dimension circuit.

As a result, two circuits of the first dimension are constructed, each consisting of two switches $S_2$4 connected in parallel. (The connections of the second circuit of the first dimension are not shown in Fig. 8.) The two circuits of the first dimension form a two-dimensional circuit. The outputs of the two-dimensional circuit combine $W_2^* = 16$ multiplexers highlighted in pale grey, forming the outputs of the switch $S_4$16. In Fig. 8, the latter connections are indicated by dotted lines and are shown completely in one copy only due to the lack of space. (They can be found in Figs. 6 and 7.)

The resulting sixteen-channel network consists of 16 copies of the switch $S_2$4 connected in parallel. Their inputs receive sparse alternative direct *p*-permutations implemented without conflict according to a single static schedule; see Lemma 2. (Alternatives *p*-permutations intersect neither in inputs nor outputs.) The paths between sources and sinks in switches $S_2$4 follow two subpaths through different circuits of the first dimension. Therefore, the switch $S_2$4 has one-channel fault tolerance since $p = \sigma = 2$.

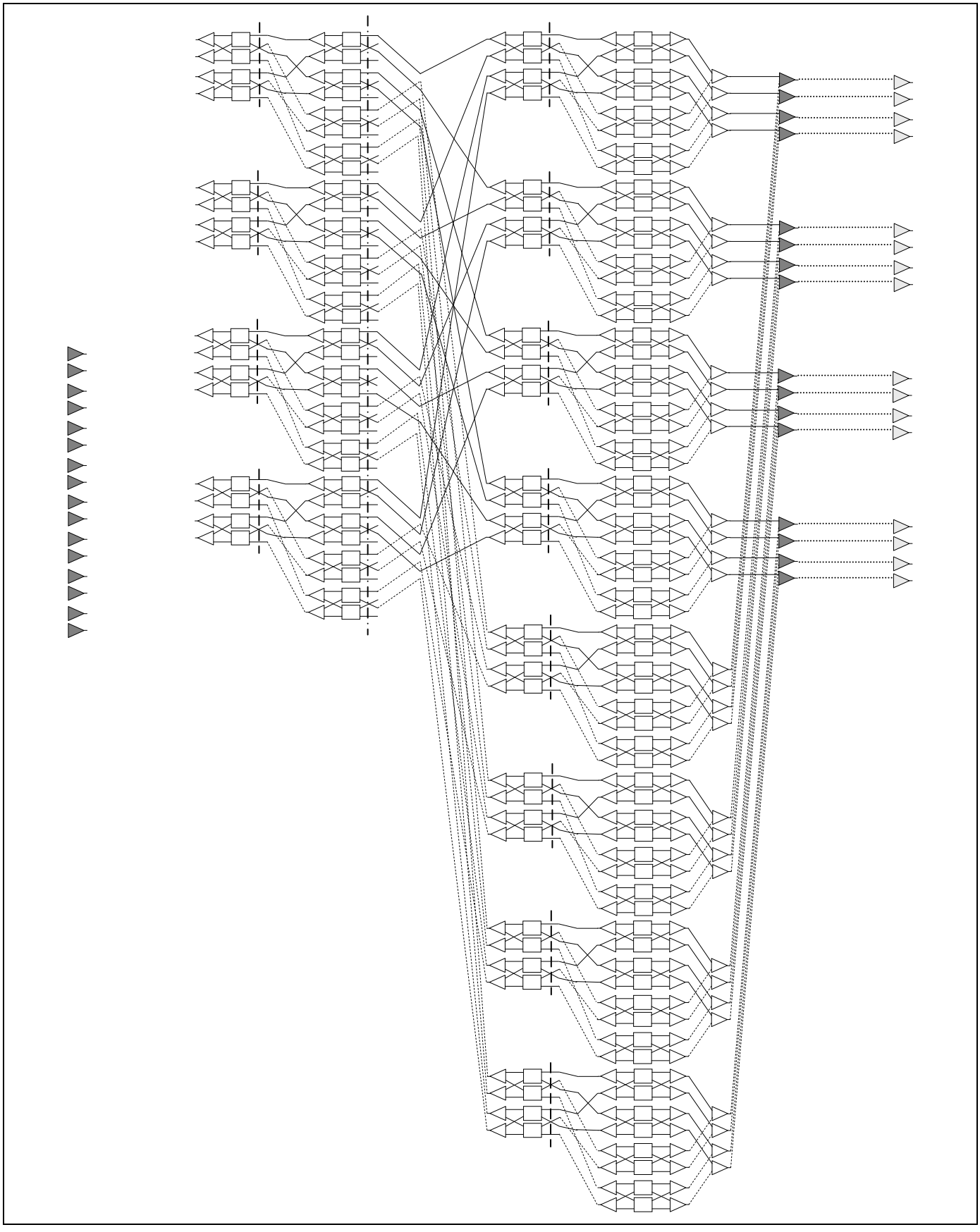

Fig. 6. Constructing the first circuit of the first dimension. Multiplexers of network $N_4 16$ not used in this circuit are shown on the left.

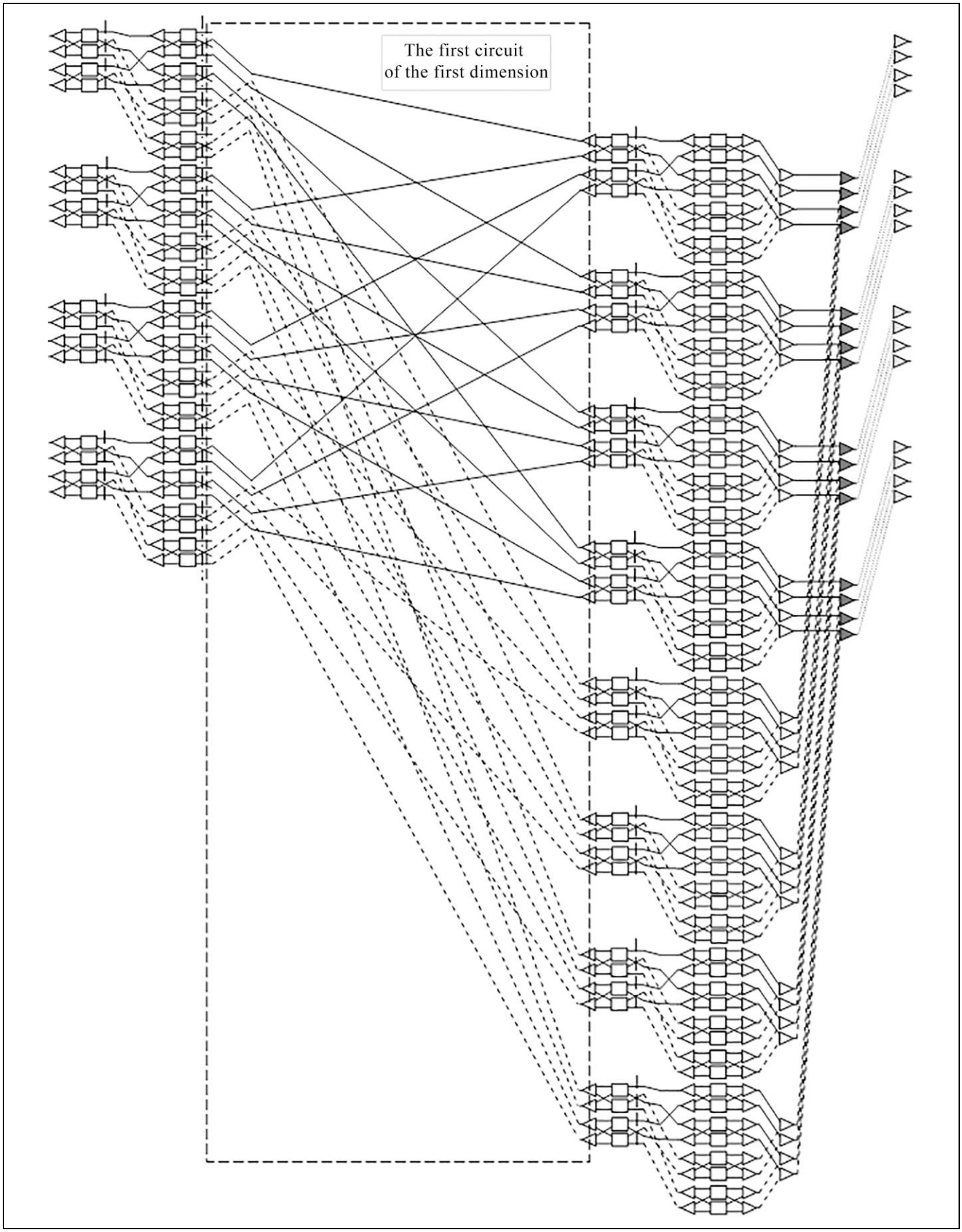

Fig. 7. Constructing the second circuit of the first dimension using the multiplexers not included in the first circuit of the first dimension.

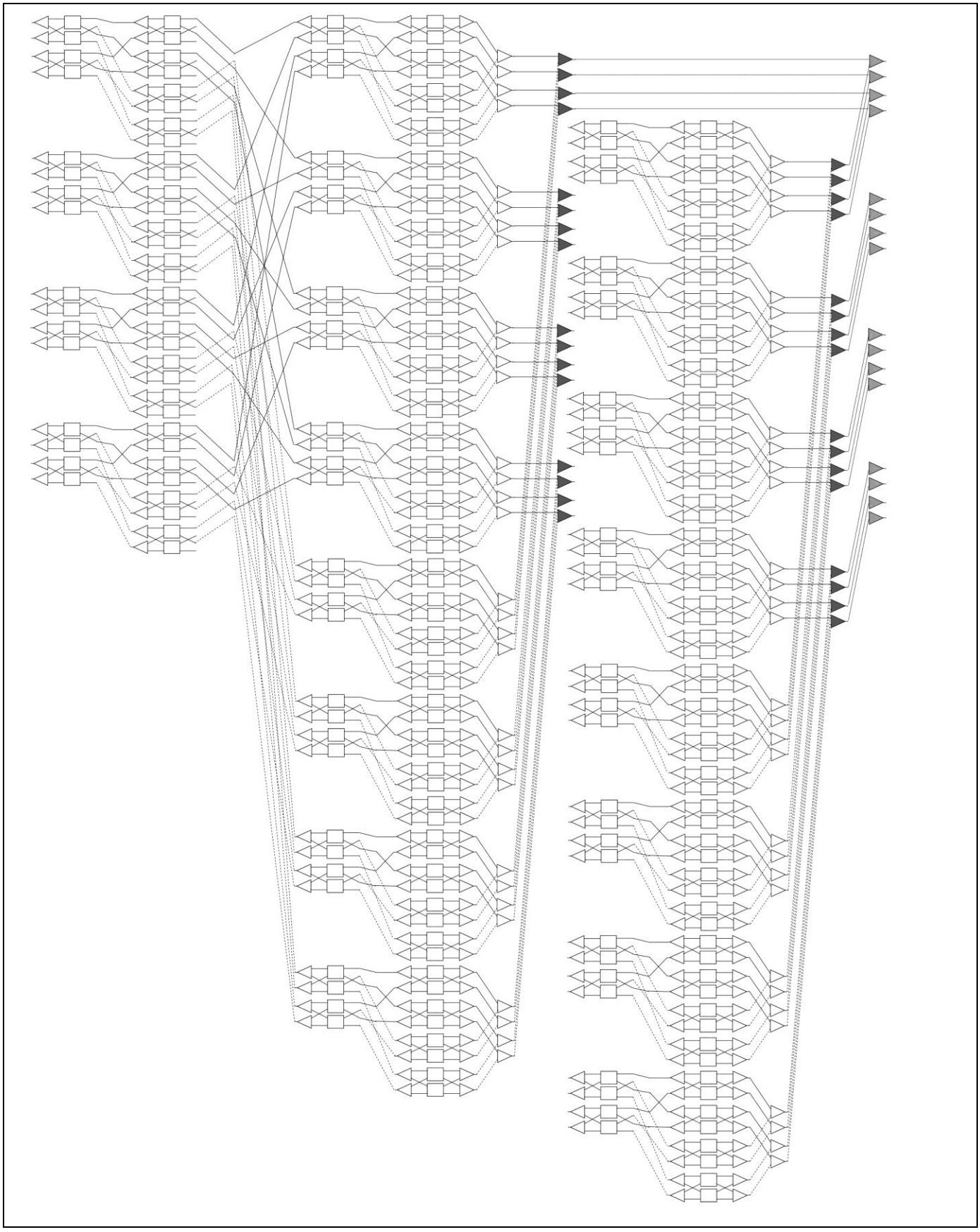

Fig. 8. Non-blocking dual switch $S_4$16 with one-channel fault tolerance.

In the general case ($p > 2$ and $\sigma \geq 2$), the network N_4N_4 is first constructed. It consists of two stages with N_2 switches S_2N_2 in each, connected by exchange links. This network has $N_4 = N_2^2$ channels, and signals can conflict at output multiplexers Mp. Therefore, it is a blocking network without channel fault tolerance.

The first stage of the network N_4N_4 has two layers of output multiplexers Mp, $W_4 = N_2V_2 = N_4(p+1)$ in total. The first inner layer of multiplexers Mp contains $W_{4,1} = pN_4$ multiplexers Mp, which together have p^2N_4 inputs.

Then the network $\underline{N}_4N_4$ is created. It contains the first stage of the network N_4N_4 and p^2 copies of the second stage of the network N_4N_4. The network $\underline{N}_4N_4$ is constructed with the following structure: copies of the second stage of the network N_4N_4 form p circuits of the first dimension, and all together, they form the circuit of the second dimension.

The network $\underline{N}_4N_4$ contains p^2N_2 switches S_2N_2, which have p^2N_4 inputs in total. Further, both layers of multiplexers Mp in the first stage of the network N_4N_4 are cut out, and the inputs of the first layer of multiplexers Mp are connected to the inputs of switches S_2N_2. This is possible since the latter and former have the same number of inputs.

We divide the p^2 copies of the second stage of the network N_4N_4 in the network $\underline{N}_4N_4$ into p groups, denoting by G ($1 \leq G \leq p$) the group number and by I ($1 \leq I \leq p$) the copy number. In fact, G is the circuit number of the first dimension, and I is the copy number in the first dimension circuit.

In addition, we introduce the following notations: J ($1 \leq J \leq N_2$) is the number of the switch S_2N_2 in the first dimension circuit, and K ($1 \leq K \leq N_2$) is the input number of each such switch S_2N_2. Thus, the input of any switch S_2N_2 is given by the composite number G, I, J, K.

Also, we denote by i ($0 \leq i \leq p-1$) the mod$p$ input number of each multiplexer Mp in their first layer. Each switch in the first stage of the network N_4N_4 contains pN_2 such multiplexers Mp. We divide them into N_2 groups, denoting by g ($1 \leq g \leq N_2$) the group number and by j ($1 \leq j \leq p$) the number of the multiplexer Mp in the group. Let k ($1 \leq k \leq N_2$) denote the number of the switch S_2N_2 in the first stage of the network N_4N_4. Thus, the input of any multiplexer Mp in the first layer is given by the composite number i, g, j, k.

An arbitrary input of the multiplexer Mp with the number i, g, j, k is connected to the input of the switch S_2N_2 with the number G, I, J, K, where $G = i + 1$, $I = j$, $J = g$, and $K = k$. As a result, p^2N_2 switches S_2N_2 are connected in parallel, and their inputs receive sparse disjoint direct p-permutations.

The cut-out multiplexers Mp of the first layer combine the outputs of the first dimension circuits. The cut-out multiplexers Mp of the second layer combine the outputs of the p circuits of the first dimension, forming the outputs of the second-level circuit (the outputs of the switch S_4N_4).

Switches S_4N_4 have the following property.

Lemma 3. *The dual switch* S_4N_4 *is a non-blocking switch with static routing on any common permutation for any p. It has* $(\sigma - 1)$*-channel fault tolerance.*

P r o o f. The first statement is based on using the switch S_2N_2 and Lemma 2. The second statement is based on the non-blocking property of the switch S_2N_2 and the fact that the p-permutation on the cross-section consists of sparse 1-permutations separated to different channels and cycles.

According to the assignment $G = i+1$, different inputs of one multiplexer in the first layer are connected to different one-dimensional circuits (different copies of the second stage of the network N_4N_4), and the inputs of different multiplexers are connected to the inputs of different switches S_2N_2 in the second stage of the network N_4N_4. In other words, there are σ different paths of switches S_2N_2 along p different paths in the switch S_4N_4. Due to $p \geq \sigma$, the latter switch is therefore $(\sigma - 1)$-channel fault-tolerant. ♦

As a result, the non-blocking self-routing switch S_4N_4 with $N_4 = N_2^2$ channels and $(\sigma - 1)$-channel fault tolerance has the performance characteristics shown in Tables 5 and 6. They are calculated using the recursive formulas $S_4 = N_2S_2 + p^2N_2S_2$ and $L_4 = N_2L_2 + p^2N_2L_2$. The switch S_4N_4 has four layers of output multiplexers Mp, $V_4 = N_4(p^4 - 1)/(p - 1)$ in total.

Note the further decrease in the exponential switching and channel complexities of the switch S_4N_4 (Table 4) compared to the switch S_2N_2 (Table 2).

Abandoning the requirement of channel fault tolerance, we can construct a non-blocking self-routing switch S_4N_4 based on the switch with the quasi-complete digraph topology. Its performance characteristics are presented in Table 7. Compared to the fault-tolerant modifications, it has more channels and smaller complexity. In addition, the entire set of switches has switching and channel complexities less than those of a two-stage switch based on a switch with the complete graph topology (Table 3) and less than those of a switch with the complete graph topology (*switchboard*).

4. EIGHT-STAGE NON-BLOCKING SWITCHES WITH FOUR-DIMENSIONAL INTERNAL PARALLELIZATION BASED ON THE GRAPH AND DIGRAPH TOPOLOGIES

The method of extending two-stage switches S_2N_2 into four-stage switches S_4N_4 can be generalized to construct eight-stage switches S_8N_8 from four-stage switches S_4N_4.

First, the network N_8N_8 is constructed. It consists of two stages with N_4 switches S_4N_4 in each, connected by exchange links. This network has $N_8 = N_4^2$ channels, and signals can conflict at output multiplexers Mp of the first stage. Therefore, it is a blocking network without channel fault tolerance.

The first stage of the network N_8N_8 has four layers of output multiplexers Mp, $W_8 = N_4V_4 = N_8(p^4 - 1)/(p - 1)$ in total. The first inner layer of multiplexers Mp contains $W_{8,1} = p^3N_8$ multiplexers Mp, which together have p^4N_8 inputs.

Then the network $\underline{N_8}N_8$ is created. It contains p^4 copies of the second stage of the network N_8N_8. The network $\underline{N_8}N_8$ is created with the following structure.

The second stage copies of the network N_8N_8, p in total, form the circuit of the first dimension, and p circuits of the first dimension form the second dimension circuit. Similarly, the third dimension circuit consists of p second dimension circuits, and the fourth dimension circuit consists of p third dimension circuits.

The network $\underline{N_8}N_8$ contains p^4N_4 switches S_4N_4, which have p^4N_8 inputs in total. Further, all four layers of multiplexers Mp in the first stage of the network N_8N_8 are cut out, and the inputs of the first layer of multiplexers Mp are connected to the inputs of switches S_4N_4. This is possible since the latter and former have the same number of inputs.

We divide the p^4 copies of the second stage of the network N_8N_8 in the network $\underline{N_8}N_8$ into p^3 groups, denoting by G ($1 \le G \le p^3$) the group number and by I ($1 \le I \le p$) the number in the group. In fact, G is the circuit number of the first dimension, and I is the copy number of S_4N_4 in the first dimension circuit.

Table 5

Performance characteristics of dual switches S_4N_4 with one-channel fault tolerance

p	N_1	$N_4 = N_1^4$	$T_4 = p$	S_4	L_4
2	2	16	2	2 720 = $N_4^{2.85}$	1 120 = $N_4^{2.53}$
3	4	256	3	238 080 = $N_4^{2.23}$	69 120 = $N_4^{2.01}$
4	7	2 401	4	8 000 132 = $N_4^{2.04}$	1 795 948 = $N_4^{1.85}$
5	11	14 641	5	1.35E+08 = $N_4^{1.95}$	24 743 290 = $N_4^{1.77}$
6	15	65 536	6	1.41E+09 = $N_4^{1.9}$	2.18E+08 = $N_4^{1.73}$
7	21	194 481	7	8.64E+09 = $N_4^{1.88}$	1.16E+09 = $N_4^{1.71}$
8	27	531 441	8	4.45E+10 = $N_4^{1.86}$	5.25E+09 = $N_4^{1.7}$

Table 6

Performance characteristics of dual switches S_4N_4 with two-channel fault tolerance

p	N_1	$N_4 = N_1^4$	$T_4 = p$	S_4	L_4
3	3	81	3	75 330 = $N_4^{2.56}$	21 870 = $N_4^{2.27}$
4	5	625	4	2 082 500 = $N_4^{2.26}$	467 500 = $N_4^{2.03}$
5	7	2 401	5	22 161 230 = $N_4^{2.17}$	4 057 690 = $N_4^{1.95}$
6	11	14 641	6	3.15E+08 = $N_4^{2.04}$	48 754 530 = $N_4^{1.85}$
7	15	50 625	7	2.25E+09 = $N_4^{1.99}$	3.01E+08 = $N_4^{1.8}$
8	19	130 321	8	1.09E+10 = $N_4^{1.96}$	1.29E+09 = $N_4^{1.78}$

Table 7

Performance characteristics of dual switches S_4N_4 based on the digraph topology

p	N_1	$N_2 = N_1^2 = p^4$	$N_4 = N_2^4 = p^8$	$T_4 = p$	S_4	L_4
2	4	16	256	2	43 520 = $N_4^{1.93}$	17 920 = $N_4^{1.77}$
3	9	81	6 561	3	6 101 730 = $N_4^{1.78}$	1 771 470 = $N_4^{1.64}$
4	16	256	65 536	4	2.18E+08 = $N_4^{1.73}$	49 020 928 = $N_4^{1.60}$
5	25	625	390 625	5	3.61E+09 = $N_4^{1.71}$	6.6E+08 = $N_4^{1.58}$
6	36	1 296	1 679 616	6	3.62E+10 = $N_4^{1.70}$	5.59E+09 = $N_4^{1.57}$
7	49	2 401	5 764 801	7	2.56E+11 = $N_4^{1.69}$	3.43E+10 = $N_4^{1.56}$
8	64	4 096	16 777 216	8	1.4E+12 = $N_4^{1.68}$	1.66E+11 = $N_4^{1.55}$

In addition, we introduce the following notations: J ($1 \le J \le N_4$) is the number of the switch S_4N_4 in the first dimension circuit, and K ($1 \le K \le N_4$) is the input number of each such switch S_4N_4. Thus, the input of any switch S_4N_4 is given by the composite number G, I, J, K.

Also, we denote by i ($0 \le i \le p-1$) the mod$p$ input number of each multiplexer Mp in their first layer. Each switch in the first stage of the network N_8N_8 contains p^3N_4 such multiplexers Mp. We divide them into N_4 groups, denoting by g ($1 \le g \le N_4$) the group number and by j ($1 \le j \le p$) the number of the multiplexer Mp in the group.

Let k ($1 \le k \le N_4$) denote the number of the switch S_4N_4 in the first stage of the network N_8N_8. Thus, the input of any multiplexer Mp in the first layer is given by the composite number i, g, j, k. For this purpose, all layers of multiplexers Mp in the first stage of the network N_8N_8 are cut out. The first inner layer contains $W_{8,1} = N_4p^3$ multiplexers Mp, and their inputs are connected to the inputs of switches S_4N_4 in the network $\underline{N_8}N_8$.

An arbitrary input of the first layer multiplexer Mp with the number i, g, j, k is connected to the input of the switch S_4N_4 with the number G, I, J, K, where $G = i + 1$, $I = j$, $J = g$, and $K = k$. As a result, p^4N_4 switches S_4N_4 are connected in parallel, and their inputs receive sparse disjoint direct p-permutations.

The cut-out multiplexers Mp of the first layer combine the outputs of the p^3 first dimension circuits. The cut-out multiplexers Mp of the second layer combine the outputs of the first dimension circuits with the same number G, forming the outputs of the second-level circuits with this number. The cut-out multiplexers Mp of the third layer combine the outputs of the second dimension circuits with the same numbers G, forming the outputs of the third-level circuits with this number. The cut-out multiplexers Mp of the fourth layer combine the outputs of the third dimension circuits, forming the outputs of the switch S_8N_8.

Switches S_8N_8 have the following property.

Lemma 4. *The dual switch S_8N_8 is a non-blocking switch with static routing on any common permutation for any p. It has $(\sigma - 1)$-channel fault tolerance.*

Proof. The first statement is based on using the switch S_4N_4 and Lemma 3. The second statement is based on the non-blocking property of the switch S_4N_4 and the fact that the p-permutation on the cross-section consists of sparse 1-permutations separated to different channels and cycles.

Channel fault-tolerance holds since the paths between sources and sinks in the switch S_4N_4 are through different circuits of each dimension and $p \ge \sigma$.

The resulting non-blocking self-routing switch S_8N_8 includes $N_8 = N_4^2$ channels and is $(\sigma - 1)$-channel fault-tolerant. ♦

The switch S_8N_8 has eight layers of output multiplexers Mp, $V_8 = N_8(p^8 - 1)/(p - 1)$ in total.

Tables 8–10 present the performance characteristics of the fastest modifications of the switch S_8N_8 for $\sigma = 2$ and $\sigma = 3$. The switching and channel complexities are calculated by the recursive formulas $S_8 = N_4S_4 + p^4N_4S_4$ and $L_8 = N_4L_4 + p^4N_4L_4$, respectively. Note that the switching and channel complexities of the switch S_8N_8 based on a digraph can be made significantly less than those of a lumped switch with the complete graph topology.

Table 8

Performance characteristics of dual switches S_8N_8 with one-channel fault tolerance

p	N_1	$N_8 = N_1^8$	$T_8 = p$	S_8	L_8
2	2	256	2	739 840 = $N_8^{2.44}$	304 640 = $N_8^{2.28}$
3	4	65 536	3	4.998E+09 = $N_8^{2.01}$	1.451E+09 = $N_8^{1.9}$
4	7	5 764 801	4	4.937E+12 = $N_8^{1.88}$	1.108E+12 = $N_8^{1.78}$

Table 9

Performance characteristics of dual switches S_8N_8 with two-channel fault tolerance

p	N_1	$N_8 = N_1^8$	$T_8 = p$	S_8	L_8
3	3	6 *561*	3	500 341 860 = $N_8^{2.28}$	145 260 540 = $N_8^{2.14}$
4	5	390 625	4	3.345E+11 = $N_8^{2.06}$	7.509E+10 = $N_8^{1.94}$
5	7	5 764 801	5	3.331E+13 = N_8^{2}	6.099E+12 = $N_8^{1.89}$

Table 10

Performance characteristics of dual switches S_8N_8 based on the digraph topology

p	N_1	$N_8 = N_1^8$	$T_8 = p$	S_8	L_8
2	4	65 536	2	189 399 040 = $N_8^{1.72}$	77 987 840 = $N_8^{1.64}$
3	9	5 764 801	3	3.283E+12 = $N_8^{1.64}$	9.531E+11 = $N_8^{1.57}$
4	16	4.29E+09	4	3.678E+15 = $N_8^{1.62}$	8.256E+14 = $N_8^{1.55}$

5. ANALYSIS OF THE RESULTS. PRACTICAL DEVELOPMENT OF THE CONSTRUCTED NETWORKS

This paper has proposed a method for constructing a new class of non-blocking self-routing photon networks with high scalability. These are the so-called dual networks based on a non-blocking dual $p\times p$ switch with a signal period of p cycles.

The dual switch is used as an integral part of the non-blocking self-routing $N_1\times N_1$ switch S_1N_1 with the quasi-complete graph or digraph topology. In the former case, the number of channels is $N_1 = p(p - 1)/\sigma + 1$, and $(\sigma - 1)$-channel fault tolerance can be provided. In the latter case, the number of channels is $N_1 = p^2$, which can be even increased. The switch with the quasi-complete (di)graph topology consists of N_1 dual $p\times p$ switches together with N_1 $1\times p$ demultiplexers Dp and $p\times 1$ multiplexers Mp without delay lines. The switching complexity of S_1N_1 is given by $S_1 = N_1(S_0 + 2p)$. The signal period T_1 of the switch S_1N_1 equals that of the dual switch: $T_1 = p$.

Switches S_1N_1 form two stages to construct a blocking $N_2\times N_2$ network N_2N_2 with $N_2 = N_1^2$ channels. Each stage consists of N_1 $N_1\times N_1$ switches, and the channels between the stages are built using exchange links. The network N_2N_2 is transformed into a non-blocking self-routing two-stage switch S_2N_2 by one-dimensional internal parallelization.

If the $N_1\times N_1$ switch is based on a quasi-complete graph, the switch S_2N_2 has $(\sigma - 1)$-channel fault tolerance since $p \geq \sigma$. The switching and channel complexities of the switch S_2N_2 are given by the recursive formulas $S_2 = N_1S_1+pN_1S_1$ and $L_2 = N_1L_1 + pN_1L_1$, respectively. By construction, the signal period T_2 of the switch S_2N_2 equals that of the switch S_1N_1: $T_2 = T_1 = p$.

If the $N_1\times N_1$ switch is based on a quasi-complete graph, the four-stage switch S_4N_4 has $(\sigma - 1)$-channel fault tolerance since $p \geq \sigma$. The switching and channel complexities of the switch S_4N_4 are given by the recursive formulas $S_4 = N_2S_2+p^2N_2S_2$ and $L_4 = N_2L_2+p^2N_2L_2$, respectively. By construction, the signal period T_4 of the switch S_4N_4 equals that of the switch S_2N_2: $T_4 = T_2 = p$.

Similarly, switches S_4N_4 form two stages to construct a blocking $N_8\times N_8$ network N_8N_8 with $N_8 = N_4^2 = N_1^8$ channels. Each stage consists of N_4 switches S_4N_4, and the channels between the stages are built using exchange links.

The network N_8N_8 is transformed into a non-blocking self-routing two-stage switch S_8N_8 by four-dimensional internal parallelization.

If the $N_1\times N_1$ switch is based on a quasi-complete graph, then the eight-stage switch S_8N_8 has $(\sigma - 1)$-channel fault tolerance as well. The switching and channel complexities of the switch S_8N_8 are given by the recursive formulas $S_8 = N_4S_4 + p^4N_4S_4$ and $L_8 = N_4L_4 + p^4N_4L_4$, respectively. By construction, the signal period T_8 of the switch S_8N_8 equals that of the switch S_4N_4: $T_8 = T_4 = p$.

The performance characteristics of the switches S_2N_2, S_4N_4, and S_8N_8 have several degrees of freedom. First of all, the number of channels grows with increasing the base p, and the speed decreases. In addition, the exponential complexity decreases with increasing the base p, and the speed can be traded for complexity. Also, more channels due to increasing the number of stages reduce the exponential complexity.

The proposed method allows constructing non-blocking self-routing networks with a self-similar structure. The switch S_2N_2 consists of dual switches S_1N_1 with the dual graph or digraph topology and uses one-dimensional internal parallelization. In turn, the switch S_4N_4 consists of switches S_2N_2 and uses two-dimensional internal parallelization. Finally, the switch S_8N_8 is composed of switches S_4N_4 and uses four-dimensional internal parallelization. All these switches inherit the basic properties of the switch S_1N_1, such as the non-blocking property under static self-routing and channel fault tolerance (if necessary), but with significantly less complexity.

The high scalability of non-blocking switches can also be achieved by repeated application of the invariant extension method to the switch S_1N_1 with the digraph topology based on a conventional $p\times p$ switch. Such extended switches have a signal period of one cycle but increased complexity. Table 11 compares the switching complexity of dual switches S_4N_4 and S_8N_8 and extended switches S_1N_1. Clearly, the switching complexity of dual switches is by several orders of magnitude lower.

Note that the dual switch resolves conflicts by the bus method only in the first stage of the switch S_1N_1. All other conflicts in all stages are prevented using internal parallelization, and the dual switches in them are used as common $p\times p$ switches. Therefore, it seems reasonable to use the dual switch in its original form [1–3] (the multiplexer–demultiplexer pair): its switching complexity is p times less. This approach will reduce the switching complexity of the dual switches S_4N_4 and S_8N_8 by several times (1.5–4.5).

Note that for small p, the complexity of the fault-tolerant switches S_2N_2 and S_4N_4 is greater than that of the complete graph; for large p, it is smaller. In this

Table 11

Complexity analysis: dual switches vs. extended digraphs

Switching complexities of non-blocking four-stage switches ($S_{4,\,D}$) and extended switches based on the quasi-complete digraph topology (S_{PO})				
p	N_4	Dual switch S_4N_4 $S_{4,\,D}$	Extended switch S_1N_1 S_{PO}	Ratio $S_{PO}/S_{4,\,D}$
2	256	46 080 = $N_4^{1.94}$	261 120 = $N_4^{2.25}$	5.67
3	6 561	6 298 560 = $N_4^{1.78}$	129 120 480 = $N_4^{2.12}$	20.5
4	65 536	22 282 400 = $N_4^{1.74}$	11 453 071 360 = $N_4^{2.09}$	514
Switching complexities of non-blocking eight-stage switches ($S_{8,\,D}$) and extended switches based on the quasi-complete digraph topology (S_{PO})				
p	N_8	Dual switch S_8N_8 $S_{8,\,D}$	Extended switch S_1N_1 S_{PO}	Ratio $S_{PO}/S_{8,\,D}$
2	65 536	18 939 9040 = $N_8^{1.72}$	17 179 607 040 = $N_8^{2.12}$	85.7
3	43 046 721	3.283E+12 = $N_8^{1.64}$	5.55822E+15 = $N_8^{2.06}$	1 640
4	4.29E+09	3.678E+15 = $N_8^{1.62}$	4.91906E+19 = $N_8^{2.04}$	13 107

case, the complexity of switches S_8N_8 is significantly less than that of the complete graph for any p. We emphasize the performance characteristics of the switch S_8N_8 for $p = 2$ and $\sigma = 1$. With $N_8 = 65\,536$ channels and half the speed, its switching complexity is comparable to that of a five-stage non-blocking Clos network based on a 64-channel YARC router [9] with $N = 32\,768$ channels constructed as a non-blocking network [7, 8]. The complexity of this non-blocking Clos network is estimated as $S = N^{1.73}$. However, this network has no parallel static or dynamic self-routing procedures. The other switches S_8N_8 with $p > 2$ and $\sigma = 1$ have even lower switching complexity and higher scalability but with lower speed.

The p-times reduced speed of the switches S_2N_2, S_4N_4, and S_8N_8 can be compensated by different protocols. It is possible to choose processors with p independent ports, divide packets into p parts and transmit them in parallel. The high scalability of these switches supports such an operation mode, albeit by reducing the number of users by p times and increasing the network complexity. An alternative is to apply the parallel-serial method for transmitting packets over p lines, as in the PCI Express protocol, without reducing the number of users.

A shortcoming of the switches S_1N_1, S_2N_2, S_4N_4, and S_8N_8 is their optimization for the conflict-free implementation of arbitrary permutations. What will be their behavior on arbitrary traffic? To determine it, we can assign the one-channel property to the multiplexers in the output stages. When receiving several input packets, such a multiplexer passes only one packet and blocks the others. The blocked packets not acknowledged by sinks are re-transmitted by the sources.

A considerable disadvantage of the proposed method is the need for parallel transmission of signal and control information, which significantly increases the required bandwidth. This disadvantage is not fatal for photon switches since an optical cable can simultaneously carry hundreds of different frequencies. However, this disadvantage can be generally eliminated with bit synchronization of signals from different channels. This can be done using the method [34, 35], locating the mutual arrangement of sources and sinks and the corresponding transmission delays from the sources. In this case, control information for dual switches and demultiplexers can be transmitted, as usual, in the form of sets of bits in the packet header.

A pleasant bonus of bit synchronization is the ability to construct an arithmetic logic unit (ALU) in the channel at each sink's input using network means. Such ALUs were developed for computing in the common channel [36]. For implementing them, it is necessary to transmit through the channel the digit values in the two-signal form: along two lines with active signals for values 0 and 1 in each. In the network ALU, an operation is performed over the number arriving through the channel and the number at the sink; the result is formed in the channel after the ALU. In the channel, it is possible to perform addition, multiplication, and any bitwise logical operations, including finding the maximum (minimum).

CONCLUSIONS

This paper has proposed a method for constructing non-blocking fault-tolerant photon networks with high scalability, considered in [5], but with much less complexity. This method is based on three main components:

- A *p*-channel dual switch with a signal period of *p* cycles, which turns out to be non-blocking on any input traffic (a prerequisite for constructing more complex non-blocking networks).
- A switch with the quasi-complete graph or digraph topology and a dual switch inside. As a result, the non-blocking property is maintained, and channel fault tolerance and higher scalability during cascading are provided compared to a pure dual switch.
- Internal parallelization to maintain the non-blocking property by preventing conflicts and maintaining fault tolerance, which provides high scalability when cascading non-blocking networks.

In the paper [5], scaling was implemented by cascade application of the invariant extension method with additional external multiplexers and demultiplexers. In this paper, scaling has been implemented by cascading smaller non-blocking networks and applying the generalized internal parallelization method at each cascading step.

The cascading of a non-blocking network with N channels is performed by constructing a blocking network with N^2 channels. This network consists of two stages with exchange links with N original non-blocking networks in each. Interlocks in this two-stage network occur at the output multiplexers of the first stage. These interlocks are prevented by separating the conflicting channels to multiple copies of the second stage and moving the multiplexers to the outputs by the second stage part responsible for packet routing. No conflicts occur in this part of the network since it consists of copies of non-blocking subnetworks that route sparse permutations. Sparse permutations are united into a complete permutation on a network with N^2 channels by the moved stages of multiplexers without conflict.

During the first cascading [5], internal parallelization is performed using p second stage copies and a one-layer stage of output multiplexers. When constructing a non-blocking network with N^4 channels, the second cascading is performed using p^2 second stage copies and a two-layer stage of the output multiplexers. When constructing a non-blocking network with N^8 channels, the third cascading is performed using p^4 second stage copies and a four-layer stage of the output multiplexers. Thus, we have designed non-blocking two-, four-, and eight-stage networks with stages consisting of non-blocking dual networks with the quasi-complete graph or digraph topology.

During each cascading, internal parallelization maintains the signal period and reduces the specific complexity of the non-blocking network. In particular, we have constructed non-blocking networks with a specific complexity not exceeding that of the theoretical non-blocking Clos network.

This method can be a fundamental base for constructing practical non-blocking switches with high scalability, static self-routing, and channel fault tolerance.

REFERENCES

1. Barabanova, E.A., Vytovtov, K.A., and Podlazov, V.S., Multistage Switches for Optical and Electronic Supercomputer Systems, *Proceedings of the 8th National Supercomputer Forum* (*NSCF–2019*), *Pereslavl-Zalessky*, 2019. URL: http://2019.nscf.ru/TesisAll/02_Apparatura/037_BarabanovaEA.pdf. (In Russian.)
2. Barabanova, E.A., Vytovtov, K.A., Vishnevsky, V.M., and Podlazov, V.S., The New Principle for the Construction of Optical Information Processing Devices for Information-Measuring Systems, *Sensors and Systems*, 2019, no. 9, pp. 3–9. (In Russian.)
3. Barabanova, E., Vytovtov, K., Podlazov, V., and Vishnevskiy, V. Model of Optical Non-blocking Information Processing System for Next-generation Telecommunication Networks, *Proceedings of the 22nd International Conference on Distributed Computer and Communication Networks: Control, Computation, Communications* (*DCCN–2019*), Moscow, 2019. Communications in Computer and Information Science, vol. 1141, Cham: Springer, pp. 188–198. DOI: 10.1007/978-3-030-36625-4_16.
4. Karavai, M.F. and Podlazov, V.S., An Invariant Extension Method for System Area Networks of Multicore Computational Systems. An Ideal System Network, *Automation Remote Control*, 2010, vol. 71, no. 12, pp. 2644–2654.
5. Barabanova, E.A., Vytovtov, K.A., and Podlazov, V.S., Two-Stage Dual Photon Switches in an Extended Scheme Basis, *Control Sciences*, 2021, no. 1, pp. 69—81.
6. Barabanova, E.A., Vytovtov, K.A., Podlazov, V.S., Non-blocking Fault-Tolerant Two-Stage Dual Photon Switches, *Control Sciences*, 2021, no. 4, pp. 67–76.
7. Clos, C., A Study of Non-locking Switching Networks, *Bell System Tech. J.*, 1953, vol. 32, pp. 406–424.
8. Benes, V.E., *Mathematical Theory of Connecting Networks and Telephone Traffic*, New York: Academic Press, 1965.
9. Scott, S., Abts, D., Kim, J. and Dally, W., The Black Widow High-radix Clos Network, *Proc. of the 33rd International Symposium on Computer Architecture* (*ISCA'2006*), Boston, 2006. https://www.researchgate.net/publication/4244660_The_Black Widow_High-Radix_Clos_Network.
10. De Sensi, D., Di Girolamo, S., McMahon, K.H., Roweth, D., and Hoefler, T., An In-Depth Analysis of the Slingshot Interconnect, *arXiv: 2008.08886v1*, August 20, 2020. https://www.researchgate.net/publication/343786515_An_In-Depth_Analysis_of_the_Slingshot_Interconnect.
11. Alverson, R., Roweth, D., and Kaplan, L., The Gemini System Interconnect, *Proceedings of the 18th IEEE Symposium on*

High Performance Interconnects, Santa Clara, CA, 2009, pp. 83–87.
12. Alverson, R., Roweth, D., Kaplan, L., and Roweth, D., Cray XC® Series Network. http://www.cray.com/Assets/PDF/products/xc/CrayXC30Networking.pdf.
13. Kim, J., Dally, W. J., Scott, S., and Abts, D., Technology-Driven, Highly-Scalable Dragonfly Topology, *Proceedings of the 35th Annual International Symposium on Computer Architecture (ISCA'2008)*, Beijing, 2008, pp. 77–88. http://users.ece.gatech.edu/~sudha/academic/class/Networks/Lectures/4%20-%20Topologies/papers/dragonfly.pdf.
14. *Mellanox OFED for Linux User Manual. Rev 2.3-1.0.1*, Mellanox Technologies, 2014. https://dlcdnets.asus.com/pub/ASUS/mb/accessory/PEM-FDR/Manual/Mellanox_OFED_Linux_User_Manual_v2_3-1_0_1.pdf.
15. Pipenger, N., On Rearrangeable and Non-blocking Switching Networks, *J. Comput. Syst. Sci.*, 1978, vol. 17, pp. 307–311.
16. Bhuyan, L.N. and Agrawal, D.P., Generalized Hypercube and Hyperbus Structures for a Computer Network, *IEEE Trans. on Computers*, 1984, vol. C-33, no. 4, pp. 323–333.
17. Tzeng, N. and Wei, S., Enhanced Hypercubes, *IEEE Trans. Computers*, 1991, vol. 40, no. 3, pp. 284–294.
18. Efe, K., A Variation on the Hypercube with Lower Diameter, *IEEE Trans. Computers*, 1991, vol. 40, no. 11, pp. 1312–1316.
19. Kim, J. and Dally, W.J., Flattened Butterfly Topology for On-Chip Networks, *IEEE Computer Architecture Letters*, 2007, vol. 6, no. 2, pp. 37–40.
20. Gu, Q.P., and Tamaki, H., Routing a Permutation in Hypercube by Two Sets of Edge-Disjoint Paths, *J. of Parallel and Distributed Comput.*, 1997, vol. 44, no. 2, pp. 147–152.
21. Lubiw, A. Counterexample to a Conjecture of Szymanski on Hypercube Routing, *Inform. Proc. Let.*, 1990, vol. 35(2), pp. 57–61.
22. Stepanenko, S., Structure and Implementation Principles of a Photonic Computer, *EPJ Web of Conferences*, vol. 224, 2019. DOI: https://doi.org/10.1051/epjconf/201922404002.
23. Zhabin, I.A., Makagon, D.V., Polyakov, D.A., Simonov, A.S., Syromyatnikov, E.L., and Shcherbak, A.N., First Generation of Angara High-Speed Interconnection Network, *Science Intensive Technologies*, 2014, no. 1, pp. 21–27. (In Russian.)
24. Stegailov, V., Agarkov, A., Biryukov, V., et al., Early Performance Evaluation of the Hybrid Cluster with Torus Interconnect Aimed at Molecular Dynamics Simulations, *Proceedings of the International Conference on Parallel Processing and Applied Mathematics*, Cham: Springer, 2017, pp. 327–336.
25. Ajima, Y., Inoue, T., Hiramoto, S., and Shimiz, T., Tofu: Interconnect for the K Computer, *Fujitsu Scientific & Technical Journal*, 2021, vol. 48, no. 3, pp. 280–285. https://www.researchgate.net/publication/265227674_Tofu_Interconnect_for_the_K_computer.
26. Arimili, B., Arimilli, A., Chung, V., et al., The PERCS High-Performance Interconnect, *Proceedings of the 18th IEEE Symposium on High Performance Interconnects*, New York, 2009, pp. 75–82.
27. Kathareios, G., Minkenberg, C., Prisacari, B., et al., Cost-Effective Diameter-Two Topologies: Analysis and Evaluation, *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC'15)*, 2015, pp. 1–11.
28. Besta, M. and Hoefler, T., Slim Fly: A Cost Effective Low-Diameter Network Topology, *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC'14)*, 2014, pp. 348–359
29. Flajslik, M., Borch, E., and Parker, M.A., Megafly: A Topology for Exascale Systems, in *High Performance Computing*, Yokota, R., Weiland, M., Keyes, D., and Trinitis, C., Eds., Cham: Springer, 2018, pp. 289–310.
30. Ahn, J.H., Binkert, N., Davis, A., et al., Hyperx: Topology, Routing, and Packaging of Efficient Large-Scale Networks, *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC'09)*, 2009, pp. 1–11.
31. Domke, J., Matsuoka, S., Ivanov, I.R., et al., Hyperx Topology: First At-scale Implementation and Comparison to the Fat-Tree, *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC'19)*, New York, Association for Computing Machinery, 2019.
32. Singla, A., Hong, C.-Y., Popa, L., and Godfrey, P.B., Jellyfish: Networking Data Centers Randomly, *The 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, 2012, San Jose, CA, USENIX, pp. 225–238.
33. Hall, M., *Combinatorial Theory*, Waltham: Blaisdell Publishing Company, 1967.
34. Stetsyura, G.G., Computer Network with the Fast Distributed Reorganization of Its Structure and Data Processing During Their Transmission, *Control Sciences*, 2017, no. 1, pp. 47–56. http://pu.mtas.ru/archive/Stetsyura_117.pdf (In Russian.)
35. Stetsyura, G.G., The Computer Clusters with Fast Synchronization of Messages and with Fast Distributed Computing by the Network Hardware, *Control Sciences*, 2020, no. 4, pp. 61–69. (In Russian.)
36. Prangishvili, I.V., Podlazov, V.S., and Stetsyura, G.G., *Lokal'nye mikroprocessornye vychislitel'nye seti. Glava 6* (Local Microprocessor Computing Networks. Chapter 6), Moscow: Nauka, 1984. (In Russian.)

This paper was recommended for publication by V.M. Vishnevsky, a member of the Editorial Board.

Received March 25, 2021, and revised August 12, 2021.
Accepted August 24, 2021.

Author information

Podlazov, Viktor Sergeevich. Dr. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ podlazov@ipu.ru

Cite this article
Podlazov, V.S. Non-blocking Fault-Tolerant Dual Photon Switches with High Scalability. *Control Sciences* **5,** 61–76 (2021). http://doi.org/10.25728/cs.2021.5.6

Original Russian Text © Podlazov, V.S., 2021, published in *Problemy Upravleniya*, 2021, no. 5, pp. 70–87.

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com