Abstract

The use of multiprocessor systems is the main method for providing a high computational power. Multistage interconnection networks (MINs) are widely used to connect processors and memory modules in multiprocessor systems. Therefore, the design of an efficient MIN is an essential requirement for the development of multiprocessor systems. In addition, a critical parameter for any efficient interconnection network is reliability. However, the problem in the way of designing high-reliable interconnection networks is high hardware cost. To solve this problem, contribution of this paper is to propose a new approach to improve the reliability of the MINs, called the rearranging links. The proposed approach is implemented on two common MINs namely extra-stage shuffle-exchange network (SEN+) and augmented shuffle-exchange network (ASEN). Meticulous analysis of terminal reliability proves that the proposed approach is an efficient method to improve the reliability of MINs. In addition, performed cost analysis confirms that utilizing it leads to emerge cost-effective MINs.

Keywords: Multiprocessor systems, Multistage interconnection network, Reliability, Cost-effectiveness, Reliability block diagrams

1. Introduction

Trends in fast networks, distributed systems, and multiprocessor computers during the past decade show that parallelism is a perfect solution for high computational power [1, 2]. Parallel processors can be defined as a computer system that is made of multiple processors that are linked together by an interconnection network and the software required for the management of processors working together [3]. Consider a generic high-end parallel architecture. This system is shown in Fig. 1. Here, several processor nodes exist that an interconnection network provides the connection between them. This network is responsible for transferring data between the processor nodes. In this system, each node consists of three components: a (probably multi-core) processor (P), a share of the main memory (M), and a cache hierarchy (C). In addition, the connection between processor nodes and global interconnection network can be created by a network interface (NI). I/O devices such as disks are also other important components of this system. I/O devices are often connected to an I/O bus, which is interfaced to the memory in each processor node via the interconnection. Therefore, it can be argued that the three vital components of a multi-processor computer are processor, memory hierarchy, and interconnection network [4].

Fig. 1. Generic multiprocessor system with distributed memory.

Therefore, multi-processor systems require efficient interconnection networks in order to achieve a desired level of performance. In this regard, the design of an efficient network is a necessary phase for the construction of high-
performance parallel computers. Generally, an interconnection network can be defined as a connection pattern of switching elements and links, which provides communication between processors and memories [5, 6].

Two main groups of interconnection networks include: Static networks and Dynamic networks. Static networks are made up of point-to-point connections that do not change during program execution. At a closer examination, static networks can be divided into two main groups: completely connected networks (CCNs) and limited connected networks (LCNs). In a completely connected network, each node is directly connected to all other nodes in the network. Therefore, there is high-speed transmission of messages from a source to a destination in this type of network. In addition, the direct connection of each node to any other node on the network can lead to a efficient and straightforward routing procedure. However, CCNs have a high cost in terms of number of links required in their design, especially for systems with large size. Therefore, LCNs can be an alternative option. LCNs do not guarantee a direct link between each pair of source-destination. Therefore, the connection between two nodes may be non-directly via some intermediate nodes. This feature makes the path length among the nodes in LCNs be usually longer compared to the case of CCNs. Another point is that as the number of intermediate nodes between a source and a destination increases, the system becomes more susceptible to failure from the viewpoint of reliability. Therefore, another problem in LCNs is in relation to reliability. Examples of LCNs are as follows: ring networks, linear arrays, tree networks, two-dimensional arrays, and cube networks.

Dynamic networks have been implemented by some switching elements, which can switch on different channels according to program requirements. Therefore, dynamic networks are flexible in providing connections between nodes and can be configured on the system requirements. These flexible and reconfigurable connections are a very important factor in many modern applications. That is why dynamic networks are frequently used multiprocessor systems so that dynamic networks are also known as multiprocessor interconnection networks [50]. Dynamic networks include bus network, crossbar network, and multistage interconnection networks (MINs), which are commonly used in multiprocessor systems [7-10]. Two basic parameters are essential in the choice of a suitable interconnection topology for use in large-size systems: cost and performance. Although crossbar network is a high-performance interconnection network, it has a high hardware cost. On the other hand, the bus network is a low-cost topology but its performance is also low. MINs can be seen as a middle ground between bus and crossbar, because MINs can provide acceptable performance at a reasonable hardware cost [11-14]. In other words, dynamic networks (i.e. bus, crossbar, and MINs) can provide a variety of options to satisfy both cost and performance parameters. In addition, MINs provide a good balance between cost and performance compared to bus and crossbar. That is why MINs are commonly used in the field of SIMD (single-instruction multiple-data) and MIMD (multiple-instruction multiple-data) parallel machines. Furthermore, MINs are used in the construction of switching fabric of high-capacity communication processors, including ATM switches, gigabit Ethernet switches, and terabit routers [13, 15].

Nowadays, one of the key applications of VLSI (very large scale integration) technology is multiprocessor systems-on-chips (MPSoCs). MPSoCs networks are different from usual chip multiprocessors. Although, chip multiprocessors can put more processors on a single chip using increased transistor densities, they do not try to leverage application needs. In contrast, MPSoCs are custom structures that can balance the constraints of VLSI technology with an application’s needs. In designing these systems, some certain requirements must be considered: low energy consumption, real-time performance that meets deadlines, and low cost. Each of these factors are difficult to meet, and the combination of these will be very challenging. On the other hand, how to design an interconnection network for use in MPSoCs is very effective to meet the basic needs [51]. Although good researches have been done in the design of efficient interconnections networks, it cannot be claimed that this problem has been solved in the field of multi-processor architectures [51]. In this area, the existing works can be divided in two classes [51, 52]: (1) A large number of classic MPSoCs architectures were designed based on bus architecture. However, the performance of the bus network is low, especially when the number of processors in the system is high. (2) Networking, which defines the concept of networks on chips (NoCs). The concept of NoCs includes distributed communication structures and the use of several different paths for data transmission. This can lead to the creation of flexible networks that are programmable and reconfigurable. According to the concepts of dynamic interconnection networks and considering discussions on NoCs, it can be argued that the MINs can also be a suitable option for use in MPSoCs due to their low cost, proper performance, and dynamic reconfigurable structure [53-55].

MINs can be divided into two main groups: single-path MINs (non-fault-tolerant MINs) and Multiple-path MINs (fault-tolerant MINs). Generally, single-path MINs have been made by a number of 2x2 crossbar switches and number of switching stages in an NxN single-path MIN is equal to \( \log_2 N \). It should be noted that the number of switches in each stage is equal to \( \frac{N}{2} \). Let us define the network complexity as the total number of 2x2 switches on
the network. Therefore, the network complexity of an N×N single-path MINs is equal to $\left(\frac{N}{2}\right) (\log_2 N)$. This network complexity is a acceptable complexity compared to the high complexity of the crossbar network that is $N^2$. However, the problem with single-path MINs is that there is only one path between each pair of source-destination. If one of the switches in the path between a given source-destination pair fails, the entire network will fail [13, 14, 16]. As a result, improving the parameters of fault-tolerance and reliability is essential in these networks [14, 16-18]. Fault-tolerance is defined as the ability of a system to continue its operation even if there are some faults in the system, although with a lower performance than normal conditions. The basic idea for improving fault-tolerance and reliability parameters in MINs is to create multiple paths between any source-destination pair. Therefore, multiple-path MINs are usually derived from single-path MINs. To create several different paths between any source-destination pair in multiple-path MINs, the basic idea is to create redundancy in the number of switches and links in the network. This redundancy can be created in different ways: (1) Increasing the number of switching stages [18-21]. (2) Connecting the several MINs in parallel [14, 22-25]. By studying these methods, it is concluded that the common point between these methods is to create redundancy in the number of switching elements and links. Therefore, the main problem of these approaches is that they can lead to increasing network complexity and thus hardware cost. This issue is very important so that can lead to the ineffectiveness of these approaches in practice. That is why one of the most challenging issues in the field of interconnection networks is reliability optimization problem [27]. Generally, the challenge in reliability optimization is to maximize network reliability with regard to cost limits.

Based on the above discussions, the aim of this paper is to provide a new approach to improve the reliability of MINs that meet the following requirements: (1) Providing a higher reliability for a MIN by the implementation of this method, compared to the initial state of the MIN. (2) This method should be cost-effective. To assess the first case, the reliability of the network can be analyzed from three different perspectives: terminal, broadcast, and network. In this paper, the terminal reliability will be analyzed carefully is an important aspect of reliability in MINs. Therefore, we are able to monitor the behavior of the proposed approach in terms of the improvement rate that occurred in the reliability parameter. In addition, like many previous works [14, 22, 23, 25, 26, 28-30], we will analyze the cost of network hardware to assess the second case. In the cost analysis, it will be assumed that the cost of a switch is proportional to the number of gates involved in it. In addition, it should be considered that the number of gates in a switch is almost equal to the number of crosspoint contained in it. For instance, a 2×2 switch has four units of hardware cost and a 3×3 switch has nine units of hardware cost. For the multiplexers and demultiplexers, it will be assumed that that each of m×1 multiplexers or 1×m demultiplexers have m units of hardware cost.

To meet two important parameters of reliability and cost, this paper will present a completely new approach based on the rearranging the links between stages. This new approach is called rearranging links and will be explained in section 3. Moreover, as will be analyzed, this technique can sometimes reduce costs, which is a very good progress in this field. In some other cases, the use of this technique can increase the cost slightly that is affordable.

It should be noted that in most previous works, each proposed method for improving the reliability is implemented only on a MIN [11, 14, 19, 21, 22, 24]. But in this paper, we will implement the proposed method of rearranging links, on two common network of SEN+ [20] and ASEN [24], which are extensively studied in [11, 14, 20, 21, 24, 31].

The rest of this paper is organized as follows: A useful background will be discussed in section 2. The proposed approach of rearranging links will be discussed in section 3. In section 4, performance of the proposed method will be analyzed. Finally, some conclusions will be made in section 5.

2. Background

2.1. General structure of MINs

MINs are used to connect a set of sources (inputs) to a set of destinations (outputs) by a number of switching stages. Generally, If the size of the switches be n×n, then the number of stages in a MIN of size N×N is equal to $\log_2 N$. Also, the number of switches in each stage is $\binom{N}{n}$. Therefore, the network complexity of an N×N MIN is equal to $O(N \log_2 N)$. This complexity is very appropriate with regard to the network complexity of the crossbar that is equal to $O(N^2)$. General structure of a MIN is shown in Fig. 2. Many MINs topologies that all are equivalent in terms of topological, follow this general structure, such as the omega network, the flip network, the indirect binary
n-cube, and the baseline and the reverse baseline networks [15, 19, 32, 33]. In this general structure, switching stages are connected to each other by an interconnection pattern of links, which is known as a permutation network or a shuffle. Different interconnection patterns can be used, but the important point is that an interconnection pattern should guarantee access and connection of all sources to all destinations.

In this paper, we will focus on two common fault-tolerant MINs, SEN+ and ASEN. There are two main reasons for considering these two networks: (1) these are common networks that have been extensively studied in [11, 14, 20, 21, 24, 31, 33]. (2) Previous studies have demonstrated that these two networks can provide a good reliability compared to usual MINs [11, 20, 21, 24, 33]. However, our plan is to show the reality that the reliability of these networks can be upgraded to a higher level by implementing the proposed method on them. In the next subsection, we will be more familiar with the structure of these two networks.

2.2. Structure of SEN+ and ASEN

A SEN+ is a SEN (shuffle-exchange network) with an additional stage. Generally, a SEN+ of size N×N has \((\log_2 N + 1)\) stages, and each stage is composed of \(\left(\frac{N}{2}\right)\) crossbar switches of size 2×2. Fig. 3 shows SEN+ of size 8×8. This network can provide two different paths between each pair of source-destination. In addition, the network complexity of a SEN+ of size N×N is equal to \(\left(\frac{N}{2}\right) \times \left(\log_2 N + 1\right)\).

Fig. 4 shows ASEN (augmented shuffle-exchange network) of size 8×8. Consider a ASEN of size N×N as a general case. In this case, ASEN has \((\log_2 N - 1)\) stages, each stage is made of \(\left(\frac{N}{2}\right)\) crossbar switches. The switches in the final stage are of size 2×2 and the remaining switches in stages 1 through ((log₂ N) − 2) are of size 3×3. In addition, there is one 2×1 multiplexer for each input link of each switch in stage 1 and one 1×2 demultiplexer for each output link of each switch in stage ((log₂ N) − 1). Therefore, the total number of multiplexers and demultiplexers in the input and output stages is equal to \(N\). The network complexity of an ASEN of size N×N is \(\left(\frac{3N}{2}\right)\left(1 + \frac{3}{4}\left((\log_2 N) − 2\right)\right)\).
2.3. State-of-the-art

With a careful review of previous works, two main methods to improve the reliability in MINs can be extracted:

1. Increasing the number of switching stages: This method can lead to an increase in the number of paths between each pair of source-destination [18-21]. Therefore, it increases the hardware cost of network as much as a stage of switches. However, the cost is low and therefore this method can be affordable. Moreover, as the analyses carried out in previous works suggests, this approach can improve the reliability [9, 11, 20, 21]. However, repeated use of this method over a network is not useful and may even lead to reduction in reliability [21]. In fact, this method can improve the reliability to a certain extent. Although this method is a low-cost approach, it cannot be a good solution for improving reliability, due to poor performance. SEN+ is an example of networks that have benefited from this method to improve reliability. That is why we have chosen this network to study and implementation of the proposed method. We will show that the method proposed in this paper resolves performance issues at only a minor cost.

2. Connecting the several MINs in parallel: In this approach, multiple MINs are connected to each other in parallel to improve reliability [14, 22-25]. The analyses carried out in the reported works show that this approach can be more efficient than the first method to improve reliability [11, 14, 22]. However, this approach makes use of multiple networks, a number of multiplexers and demultiplexers, and sometimes increases the size of the switches. Therefore, the main disadvantage of this approach is its high hardware cost. As a result, this approach has a relatively good performance in terms of reliability, but not suitable for the development of cost-effective systems. ASEN is an example of networks that have used this approach to improve reliability. Therefore, we have chosen this network to implementation of the proposed method. As it will be demonstrated later, the method proposed in this paper can reduce the costs in this network and can improve the reliability.

According to the above discussions, it can be deduced that both of these methods are based on creating redundancy in the number of switching elements that in turn leads to increased costs. Although the former method is less expensive as compared to the second approach, it suffers from low performance. On the other hand, although the latter approach is more efficient than the first method, it suffers from high cost. Therefore, in none of the aforementioned approaches, both parameters cost and reliability are fully satisfied. Our aim in this paper is to present a new method that can adequately support these vital parameters. For this purpose, we will present a method which uses an innovative idea which unlike the previous approaches, is not based on creating redundancy in the number of switching elements. Instead, the core idea is based on changes in the connection form of links between different stages, which will be explained further in section 3. In addition to attain the favorable conditions in terms of cost, as will be demonstrated in section 4, this approach can also lead to significant improvement in terms of reliability. Therefore, according to these arguments, we believe that the proposed method would be a proper choice to meet both cost and reliability parameters.

Clearly, contribution of this paper can be explained as follows: (i) a new method to improve the reliability of MINs will be presented. (ii) The proposed method is implemented on two well-known networks of SEN+ and ASEN [9, 11, 14, 18-25, 33, 39], thus emerging two new topologies. (iii) MINs’ reliability should be accurately evaluated. However, precise modeling of interconnection networks reliability is extremely complex [9, 11]. That is why great efforts have been made so far in the analysis of various parameters on the network, but less attention has been paid to the analysis of reliability. In this paper, reliability block diagrams (RBDs will be used to conduct a detailed
analysis of reliability. RBDs are a visual and accurate method that can be used to analyze the reliability in a variety of systems [34]. In addition, many researchers believe that the use of RBDs is a necessary phase in the analysis of reliability [35-37]. Although, RBDs have been used in some previous works to analyze the reliability of the MINs, almost all of them have used simple series-parallel RBDs to analyze the reliability of MINs [11, 22, 33, 38, 39]. However, this approach cannot lead in many cases to a detailed analysis of the reliability in fault-tolerant MINs. In this paper, in order to provide a more accurate analysis, we will consider a RBD As much as possible close to the network structure from the standpoint of reliability. Then, in the case of complex systems, we will use a technique called decomposition method [40-44] for determining the reliability that will be explained more in section 4.1.1.

3. Rearranging links: the proposed approach to improve the reliability of MINs

We initially try to explain the proposed method, rearranging links, in subsection 3.1 and then, we will implement this method on ASEN and SEN+ in subsections 3.2 and 3.3, respectively.

3.1. Rearranging links

The rearranging links improves MINs in several steps as follows:

**Step 1** Assuming a given source and destination of the network, switches of stage (i) that are connected to the same switches in stage (i+1) should be connected to switches in stage (i+1) that end to the associated destination. It is noted that the selected switches from stage (i) should be connected to the given source.

**Step 2** Increasing the size of either multiplexers or de-multiplexers or both. This option is mandatory when the network is out of the general structure. In the general structure, each network source should be able to access all the network destinations. Also, this step can be used for homogenizing the path lengths (defined as the number of switches between a source-destination pair) between a specific network source and all destinations.

According to step 1, it is clear that the proposed method is based on the rearranging links between stages of the network. Therefore, it does not make any increased costs. Also, according to step 2, the proposed method slightly increases the cost because of using the multiplexers. However, as it will be shown later, this approach in some cases can reduce the costs. In the next subsections, we will be more familiar with this method by implementing it on two popular networks of SEN+ and ASEN. Moreover, as it is shown in section 4, the proposed method has a tremendous positive impact on both basic parameters reliability and cost.

Also, in order to create a clear view of the proposed method, the method as algorithm is given in Algorithm 1.

---

**Algorithm 1: Rearranging links**

**Input**
- \(n\): Number of stages
- \(i\): Stage number
- \(S\): A given source in the network
- \(D\): A given destination in the network

**Output**
- Improved network

```plaintext
for S and D
    for i=1 to n
        if COUNT_SWITCH (i) = 2
            CONNECT (SWITCH (i), SWITCH (i+1)) and CONNECT (~SWITCH (i), ~SWITCH (i+1));
        if COUNT_SWITCH (i) > 2
            CONNECT (SAME_SWITCH (i), SWITCH (i+1)) and CONNECT (~SWITCH (i), ~SWITCH (i+1));
    end for
end for
```

Also, each of the ancillary operations in the Algorithm 1 namely COUNT_SWITCH (i), SWITCH (i), SAME_SWITCH (i), CONNECT (S (m), S (n)), and ADD_MUX () are shown separately in Algorithms 2 through 6, respectively.

---

**Fig. 5.** Algorithm 1: Rearranging links.
Algorithm 2: COUNT_SWITCH (i)

Inputs:
- N: Number of sources and destinations
- n: Number of stages
- i: Stage number
- j: Switch number
- S: A given source in the network
- D: A given destination in the network
- Switch(i,j) // j-th switch in stage i

Outputs:
- Count(i) // Number of switches connected to the source S and destination D in stage i

{ 
  for i = 1 to n
    for j = 1 to N/2
      Count(i) = 0;
      if Switch(i,j) is connected to S and D
        Count(i) = Count(i) + 1;
    end for
  end for
  Return (Count(i));
}

Fig. 6. Algorithm 2: COUNT_SWITCH (i).

Algorithm 3: SWITCH (i)

Inputs:
- N: Number of sources and destinations
- n: Number of stages
- i: Stage number
- j: Switch number
- S: A given source in the network
- D: A given destination in the network
- Switch(i,j) // j-th switch in stage i

Outputs:
- SC(i) // Set of switches connected to the source S and destination D in stage i

{ 
  for i = 1 to n
    for j = 1 to N/2
      SC(i) = 0;
      if Switch(i,j) is connected to S and D
        Switch(i,j) is added to the set of SC(i);
    end for
  end for
  Return (SC(i));
}

Fig. 7. Algorithm 3: SWITCH (i).
Most of the previous works [11, 20, 22-25] often focused on just one network. In contrast, in this paper to provide a comprehensive approach with a more detailed argument, the procedure for two networks ASEN and SEN+ is presented separately. In what follows, the proposed method, rearranging links, will be implemented on the ASEN and SEN+ in subsections 3.2 and subsection 3.3, respectively. In each subsection how the rearranging links method is implemented will be exactly discussed.
3.2. Implementation the rearranging links on the ASEN

In this section, the proposed method, rearranging links, will be implemented on ASEN and the new network called ASEN-RL is emerged.

**Step 1)** Assume the source and destination 000. Possible paths between the source and destination 000 are highlighted in Fig. 11. In stage 1, switches SE\(_{0,1}\) and SE\(_{2,1}\) are connected to the same switches (SE\(_{0,2}\) and SE\(_{2,2}\)) in stage 2, which are highlighted with dash line in Fig. 11. Also, switches SE\(_{1,1}\) and SE\(_{3,1}\) are connected to the same switches (SE\(_{1,2}\) and SE\(_{3,2}\)) in stage 2, which are highlighted in Fig. 11. Therefore, according to the discussion in section 3.1, switches of SE\(_{0,1}\) and SE\(_{2,1}\) or SE\(_{1,1}\) and SE\(_{3,1}\), which are connected to the same switches in stage 2, should be connected to switches that are leading to the specific destination 000. The two switches of SE\(_{0,1}\) and SE\(_{2,1}\) are picked. On the other hand, in the second stage, both switches of SE\(_{0,2}\) and SE\(_{1,2}\) are leading to the given destination 000. Therefore, two switches SE\(_{0,1}\) and SE\(_{2,1}\) should be connected to two switches of SE\(_{0,2}\) and SE\(_{1,2}\). As a result, both switches of SE\(_{1,1}\) and SE\(_{3,1}\) should be also connected to two switches of SE\(_{2,2}\) and SE\(_{3,2}\). Fig. 12 depicts ASEN at the end of this step.

![Fig. 11. Paths between source and destination 000 in 8x8 ASEN.](image1)

![Fig. 12. 8x8 ASEN after step 1 of the rearranging links.](image2)

**Step 2)** as discussed in section 3.1, step 2 of the rearranging link method is used in two cases: (1) when the network is out of the general structure. (2) When path lengths from a source to different destinations are not identical.
The ASEN is plagued by the second problem after the first step of the rearranging links. For example, consider the source and destination 000 in Fig. 12. For this case, the four paths can be used: 000→SE\(_{0,1}\)→SE\(_{0,2}\)→100, 000→SE\(_{0,1}\)→SE\(_{1,2}\)→000, 000→SE\(_{0,2}\)→SE\(_{2,1}\)→SE\(_{1,2}\)→000. As it can be observed, all path lengths equals to 2. Now, consider the source 000 and destination 100 in Fig. 12. For this case, the four paths can be used: 000→SE\(_{0,1}\)→SE\(_{1,1}\)→SE\(_{2,2}\)→100, 000→SE\(_{0,1}\)→SE\(_{1,1}\)→SE\(_{2,2}\)→100, 000→SE\(_{0,2}\)→SE\(_{3,1}\)→SE\(_{3,2}\)→100. As it can be observed, all path lengths are 3. Therefore, in Fig. 12, path lengths from a given source to different destinations are not identical. In this case, the problem can be resolved by increasing the size of multiplexers from 2×1 to 4×1. The resulting network is shown in Fig. 13.

To establish a clearer vision of the proposed technique in ASEN network, values of different operations in the proposed method provided on Algorithm 1 are summarized in table 1.

| Table 1. Values of different operations in the proposed method for ASEN. |
|------------------|------------------|
| \(S\) | \(000\) |
| \(D\) | \(000\) |
| COUNT\_SWITCH (i) | 
| i=1 | 4 |
| i=2 | 2 |
| SWITCH (i) | 
| i=1 | \(\{SE_{0,1}, SE_{1,1}, SE_{2,1}, SE_{3,1}\}\) |
| i=2 | \(\{SE_{0,2}, SE_{1,2}\}\) |
| SAME\_SWITCH(i) | 
| i=1 | \(\{(SE_{0,1}, SE_{2,1}), (SE_{1,1}, SE_{3,1})\}\) |
| i=2 | - |
| CONNECT (S (1), S (2)) | \((SE_{0,1}, SE_{2,1})\) are connected to \((SE_{0,2}, SE_{1,2})\) and \((SE_{1,1}, SE_{3,1})\) are connected to \((SE_{2,2}, SE_{3,2})\) |
| ADD\_MUX () | Using N 4×1 multiplexers |

In Fig. 13, to better understand, the connections to the source 000 are highlighted. It should be noted that in Fig. 12 the link (loop) between switches SE\(_{0,1}\) and SE\(_{1,1}\) as well as SE\(_{2,1}\) and SE\(_{3,1}\) in stage 1 were used to redirect the request to access a specific destination. While in Fig. 13, all the sources are directly connected to all the necessary switches to access all destinations. Consequently, these links (loops) can be removed as shown in Fig. 13, which this can lead to a tremendous reduction in the network complexity and the hardware cost. Due to this point, the final network, ASEN-RL is shown in Fig. 14.

Consider an ASEN-RL of size N×N. In this general case, ASEN-RL consists of \((\log_2 N) - 1\) stages and each stage has \(\frac{N}{2}\) crossbar switches. All the crossbar switches are of size 2×2. There is one multiplexer of size 4×1 for each input link of switch in stage 1 and one demultiplexer of size 1×2 for each output link of switch in stage \((\log_2 N) - 1\). Therefore, there are \(N\) multiplexers in the input stage and \(N\) demultiplexers in the output stage in an ASEN-RL of size N×N. The network complexity of an N×N ASEN-RL is \(\frac{N}{2} (3 + ((\log_2 N) - 1))\).
MINs are organized in a number of stages that each of these stages are made up of several small-size crossbar switches. A crossbar switch suitable for distributed control in the environment a MIN is shown in Fig. 15 [8]. Each incoming packet to the switch contains a tag that indicates the destination of the packet. The selector of the input port checks this tag, and then it considers the appropriate output port based it. If the arbiter of this output port is free then the connection is established. However, if it is busy, then the connection could not be established. It should be noted that all selectors can work concurrently and asynchronously.

In fact, a switch of size 2×2 can be in one of two states of straight or exchange. Fig. 16 shows these two states. Let the upper input and output lines be labeled i and the lower input and output lines be labeled j. (1) straight: input i to output i, input j to output j; (2) exchange: input i to output j, input j to output i.

Routing tag consists of binary digits that are used to control the connection through different switching stages from source to the destination. Let the source S and destination D be represented in binary as:

\[ S = s_n \ldots s_1, n = \log_2 N \]
\[ D = d_n \ldots d_1, n = \log_2 N \]
Routing for ASEN-RL can be implemented as self-routing. In a self-routing procedure, crossbar switches check their incoming data and set themselves accordingly. In addition, this approach does not require any central routing hardware [8, 45].

**Self-routing procedure for ASEN-RL:**

In this scenario, it is assumed that the sources and switches are able to detect the faults of those switches that are connected to them. So far, several techniques have been proposed in the literature to detect faults [46-49].

1. For connecting a given source-destination pair to each other, there are two basic paths (two basic paths are defined according to bit $d_n$), one of which must be selected. If the selected basic path is currently not available due to a failure or occupied switch in stage 1, then the next basic path should be selected. If this path is not also available, then the network fails.

2. In any basic path, bits $d_n$ to $d_2$ are used for routing in the intermediate stages between multiplexers and demultiplexers. In the other words, bits $d_n$ to $d_2$ are used for routing in switches located in stages of 1 to $(\log_2 N - 1)$, respectively. In each of these intermediate switches, bit 0 is meant to establish a connection with upper output and bit 1 is meant to establish a connection with lower output. Fig.17 shows this concept.

![Fig. 17. Self-routing in each switch.](image)

In any basic path, if the first main path is not available due to a failure or busy switch in stage 2, then the switch in stage 1 uses its another output. Therefore, request will be sent to the successor switch in the next stage and this switch will continue the routing, using the same routing tag. If the new path is not also available, then the next basic path will be used and this new basic path will follow the same routing trend used in the first basic path. Finally, if no possible paths in all basic paths are available, then the request will fail.

3. In the end, a request that has reached to demultiplexer will be sent to the upper or lower destination based on least significant bit of the tag (i.e. $d_1$).

For example, a successful ASEN-RL routing scheme for permutation $P = (01234567)$ is shown in Fig. 18. In this figure, the 3-bit destination index on each input of the switching elements with the appropriate bit used for controlling the switches have been highlighted, so bits that are used to control routing in each case can be viewed.

![Fig. 18. A successful ASEN-RL routing scheme for permutation $P = (01234567)$](image)
3.3. Implementation the rearranging links on the SEN+

In this section, the proposed method, rearranging links, will be implemented on SEN+ emerging the new network called the SEN-RL.

**Step 1** In Fig. 3, the source and destination 000 is assumed. Existing paths between these two nodes are highlighted in Fig. 19. As it can be observed, in the first stage, only the switch of SE₀₁ is connected to the source 000. But we need two switches connected to the source for implementation of the first step of the rearranging links. Therefore, links between the first and second stage remain unchanged. In stage 2, only two switches of SE₀₂ and SE₁₂ are connected to the source 000. Consequently, according to the first step of the rearranging links, two switches should be connected to both switches of SE₀₃ and SE₂₃ of the stage 3 that will end to the destination 000. Therefore, both switches of SE₂₂ and SE₁₂ should also be connected to switches of SE₁₃ and SE₂₃. This time, we consider the stages 3 and 4. In stage 3, only two switches of SE₀₃ and SE₂₃ are connected to the source 000. However, there is only one switch (SE₀₄) leads to the desired destination in the last stage. However, we need two switches in the last stage to connect two switches of SE₀₃ and SE₂₃ of the penultimate stage to them. Therefore, the links between stages 3 and 4 remain unchanged. The final structure of the SEN+ after the first step of rearranging links is depicted in Fig. 20.

![Fig. 19. Paths between source and destination 000 in 8x8 SEN+.](image1)

![Fig. 20. 8x8 SEN+ after step 1 of the rearranging links.](image2)

**Step 2** as previously discussed, step 2 of the rearranging links method is used in two cases: (1) when the network is out of the general structure. (2) When path lengths from a source to different destinations are not identical.

As it is shown in section 3.2, ASEN was faced with the latter problem. But, here SEN+ is caught by the former problem. For example, consider the source 000 and destination 100 in the Fig. 20. Clearly no path exists between the two nodes source and destination for communicating. This means that the network is out of the general structure. In this case, we can resolve this issue by adding some 2x1 multiplexers to the network. The final structure, SEN-RL is shown in Fig. 21.

In order to a clearer vision of the proposed technique in SEN+ network, values of different operations in the proposed method provided on Algorithm 1 are summarized in table 2.
Table 2. Values of different operations in the proposed method for SEN+.

<table>
<thead>
<tr>
<th>S</th>
<th>D</th>
<th>COUNT_SWITCH (i)</th>
<th>SWITCH (i)</th>
<th>SAME_SWITCH (i)</th>
<th>CONNECT (S(1), S(2))</th>
<th>ADD_MUX ()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>000</td>
<td></td>
<td>i=1</td>
<td>i=1</td>
<td>(SE₀,₁)</td>
<td>Using 2x1 multiplexers</td>
</tr>
<tr>
<td></td>
<td>000</td>
<td></td>
<td>i=2</td>
<td>i=2</td>
<td>{SE₀₁₂, SE₁₁₂}</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>i=3</td>
<td>i=3</td>
<td>{SE₀₂₃}</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>i=4</td>
<td>i=4</td>
<td>{SE₀₄}</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>i=1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>i=2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>i=3</td>
<td>i=3</td>
<td>{(SE₀₃, SE₂₃)}</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>i=4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Consider a SEN-RL of size N×N. In this general case, SEN-RL is comprised of (\(\frac{\log_2 N + 1}{2}\)) stages and each stage made up of (\(\frac{N}{2}\)) switching elements of size 2×2. There is one 2×1 multiplexer for each input link of switch in stage 1. Therefore, an SEN-RL of size N×N has N multiplexers of size 2×1 in the input stage. The network complexity of an N×N SEN-RL is equal to (\(\frac{N}{2}\))(1 + ((\(\log_2 N\)) + 1)). In this network, there are four paths between each source-destination pair.

Routing tag for SEN-RL:

Routing tag consists of binary digits that are used to control the connection through different switching stages from source to the destination. Let the source S and destination D be represented in binary as:

\[ S = s_n \ldots s_1, n = \log_2 N \]
\[ D = d_n \ldots d_1, n = \log_2 N \]

Since the number of stages in SEN-RL is equal to ((\(\log_2 N\)) + 1); for a true self-routing, we consider the number of bits of destination routing tag equal to the number of stages. Thus, the tag D = d_n d_n ... d_1 will be used for the self-routing.

Self-routing procedure for SEN-RL:
For each source-destination pair, the bit $d_n$ determines the appropriate multiplexer for starting the connection. Then bits $d_{n-1}$ to $d_1$ are used for routing across stages of 0 to $(\log_2 N) + 1$, respectively. For each entry in each switch, bit 0 means to establish connection with the upper output and bit 1 means to establish connection with the lower output. If the switch in stage 2 or 3 is not available, then other output of switch in stage 1 or 2 can be used to deliver the request to successor switch in stage 2 or 3. For example, Fig. 22 shows a successful SEN-RL routing scheme for permutation $P = (01234567)_{(63051274)}$.

![Fig. 22. A successful SEN-RL routing scheme for permutation $P = (01234567)_{(63051274)}$.]

4. Performance analysis

In this section, we will analyze the performance of the proposed method of rearranging links by analyzing the performance of the ASEN-RL and SEN-RL and compare them with the performance of ASEN and SEN. In this paper, our focus is on two crucial performance parameters of reliability and cost. Therefore, first, we will assess the reliability in section 4.1 and then the cost parameter will be analyzed in section 4.2.

4.1. Reliability

A large number of researchers in the field of interconnection networks have admitted that reliability is an essential parameter for a network [9, 11, 14, 20-25, 38, 39, 60]. In fact, the reliability analysis is a mathematical description of a system that can provide detailed information about system performance. In this paper, reliability will be analyzed from the perspective of terminal. Terminal reliability is defined as the probability of a successful connection between a given source-destination pair in the network.

Typically, a system is made up of a number of components. Reliability of the system is a function of the reliability of its components that is defined according to the type relationships between components. Also, each component can be in two different modes: working or failed. Reliability Block Diagram (RBD) is a visual method that shows how the relationship between different components of the system in terms of reliability. Therefore, the RBDs are an accurate method that can be used to model various systems. After modeling system reliability by RBDs, the reliability equation can be extracted for analysis of reliability [41, 44]. In a RBD, each component of the system is shown by a block and each block is connected to some other blocks based on the relationship between them in terms of reliability. Regarding the relationship between the components of the system in terms of reliability, they can be organized in four different structures: series, parallel, series-parallel, and complex (non-series-parallel). For more information about these different systems, it can be referred to [40, 41, 44]

4.1.1. Terminal reliability

Since terminal reliability is defined as the probability of successful connection between a source-destination pair, it can be analyzed by considering a specific source-destination pair in the network. In a MIN, for a successful connection between a given source-destination pair, the presence of at least one fault-free path between them is essential.
In this paper, like many previous works [25, 38, 39, 56-59], the following assumptions were used in the reliability analysis:

1. In the field of MINs, the researchers believe that the switches are the more unreliable and therefore more prone to failure compared to links [38]. That is why in most of the previous works, switch fault model is used to analyze the reliability [11, 14, 20-23, 25, 38, 39, 56-59]. Therefore, the switch fault model will be used to analyze the reliability in this paper. In the switch fault model, it is assumed that each of switching components such as switching elements, multiplexers, and demultiplexers may fail.

2. Switching components can be in one of these two states: healthy or faulty.

3. It assumes that in the event of a fault in a switch, the fault in the switches is stable. These types of faults are also known as static faults.

4. A faulty switching components cannot perform any of the connection functions.

5. It will be assumed that the reliability of a switch of size 2x2 (SE2) is equal to $r$. Therefore, the reliability of other switching components can be calculated based on $r$ by considering the number of gates in them. Also, it should be noted that the hardware complexity of a component is directly proportional to the number of gates [22, 25]. According to these discussions, the reliability of each switching components are calculated as follows:

\[
\begin{align*}
  r_{MUX2} &= r_{DEMUX2} = r^{\frac{1}{2}} \\
  r_{MUX4} &= r_{SE2} = r \\
  r_{SE3} &= r^{\frac{9}{4}}
\end{align*}
\]

According to the SEN+ structure (shown in Fig. 3), its RBD for terminal reliability is depicted in Fig. 23. In the RBDs, switching elements of size n x n is shown as SE_n and an n x 1 MUX and 1 x n DMUX are shown as MUX_n and DM_n, respectively. The RBD in Fig. 23 is composed of three blocks A, B, and C. Reliability for each of these blocks is obtained by the equations (4) and (5).

\[
\begin{align*}
  R(A) &= R(B) = r & \text{(4)} \\
  R(C) &= 1 - (1 - r^{(\log_2 N - 1)})^2 & \text{(5)}
\end{align*}
\]

The terminal reliability of the N x N SEN+ is calculated as follows:

\[
R_t(SEN+) = r^2[1 - (1 - r^{(\log_2 N - 1)})^2] & \text{(6)}
\]

Similar to SEN+, according to the SEN-RL structure (shown in Fig. 21), its RBD for terminal reliability is depicted in Fig. 24.
In the RBD of Fig. 24, first, we calculate the reliability of parts A and B, and then the reliability of the entire network can be calculated according to them. For part A, we have:

\[ R(A) = r^2(1 - (1 - r^{(\log_2 N) - 3})^2) \]  \hspace{1cm} (7)

According to the equation (7), we have:

\[ R(B) = 1 - (1 - r^2(1 - (1 - r^{(\log_2 N) - 3})^2))^2 \]  \hspace{1cm} (8)

Now, given that reliability of part B is calculated, RBD in Fig. 24 can be seen as a series system made up of four blocks: one multiplexer block, two 2x2 switch blocks, and block B. As a result, the reliability of the entire network can be calculated by multiplying the reliability of the blocks on each other. Therefore, according to the equation (8), we have:

\[ R_{t}(SEN - RL) = r^5(1 - (1 - r^2(1 - (1 - r^{(\log_2 N) - 3})^2))^2) \]  \hspace{1cm} (9)

Next network, which will be examined, is the ASEN. Since this is an intractable MIN from the viewpoint of reliability, its reliability is not analyzed sufficiently exact in previous works (such as [11]). Therefore, ASEN has particular importance in terms of reliability analysis. The main reason for the increased complexity of this network is the use of links (loops) between some pairs of switches for improving fault-tolerance capability. Use of these loops causes the network to show complex variable behavior for different network sizes. Therefore, the modeling of the network is difficult to careful analysis of reliability even for a small network size 8×8. However, in this paper, we will provide very precise analysis of the reliability of the network for network sizes 8×8 and 16×16 by modeling it with complex (non-series-parallel) RBDs. It should be noted that the analysis conducted in this paper is very valuable for future researches to precise analyze the reliability of the MINs. However, the calculation of reliability is also valuable for larger sizes. In this case, it is possible to derive useful reliability equation for N×N ASEN based on the terminal reliability equations of 8×8 and 16×16 ASEN.

The terminal reliability RBD for 8×8 ASEN is shown in Fig. 25.
By calculating the reliability of the series blocks specified in the RBD of Fig. 25, the diagram can be plotted in Fig. 26. In Fig. 26, the reliability of each block is written inside it. The RBD is from the complex type; therefore, to analyze it, the decomposition method can be used. For this purpose, we need to pick one block as the key block. The key block is highlighted in Fig. 26. Therefore, we have:

\[
R_t(ASEN) = R_{key}(R_{ASEN}|R_{key}) + R_{key}(R_{ASEN}|R_{key})
\]  \hspace{1cm} (10)

On the other hand, we have:

\[
R_{key}(R_{ASEN}|R_{key}) = (1 - r^{\frac{11}{2}})(r^{\frac{11}{2}} - 1 - r^{\frac{3}{2}})(1 - r^{\frac{13}{2}})
\]  \hspace{1cm} (11)

To calculate the second term \(R_{key}(R_{ASEN}|R_{key})\) of the equation (10), the RBD in Fig. 26 can be considered in Fig. 27, that again one block is highlighted as a key block. According to this figure, we have:

\[
R_{key}(R_{ASEN}|R_{key}) = (r^{\frac{11}{2}})(r^{\frac{9}{2}} - 1 - r^{\frac{3}{2}})^2 + (1 - r^{\frac{9}{2}})(1 - (1 - r^{\frac{3}{2}})^2)
\]  \hspace{1cm} (12)

Therefore, considering equations (10) to (12), the terminal reliability of 8×8 ASEN is given by:

\[
R_t(ASEN) = \left(1 - r^{\frac{11}{2}}\right)\left(r^{\frac{11}{2}} - 1 - r^{\frac{3}{2}}\right)\left(1 - r^{\frac{13}{2}}\right) + \left(r^{\frac{11}{2}}\right)^2\left(1 - r^{\frac{9}{2}}\right)\left(1 - (1 - r^{\frac{3}{2}})^2\right) + (1 - r^{\frac{9}{2}})(1 - (1 - r^{\frac{3}{2}})^2)(1 - r^{\frac{13}{2}})
\]  \hspace{1cm} (13)
In the next figure, the terminal reliability RBD of ASEN is shown for size $16 \times 16$. According to the Fig. 28, as the network size increases, the complexity of the network increases in terms of reliability analysis. Similar to Fig. 25, by calculating the reliability of the series blocks specified in the RBD of Fig. 28, diagram can be plotted in Fig. 29. In Fig. 29, the reliability of each block is written inside it and the key block is highlighted. Therefore, with an approach similar to that used in calculating the reliability of the $8 \times 8$ ASEN, the terminal reliability of $16 \times 16$ ASEN is given by:

$$R_T(ASEN) = r_T^{12} \left( r_T^9 \left( 1 - \left( r_T^{3} \right)^2 \right) + \left( 1 - r_T^{12} \right) \left( r_T^2 \left( 1 - \left( 1 - r_T^{3} \right) \left( 1 - r_T^{6} \right) \right) + \left( 1 - r_T^{2} \right) \left( 1 - \left( 1 - r_T^{3} \right) \left( 1 - r_T^{15} \right) \right) \right) + \left( 1 - r_T^{9} \right) \left( r_T^2 \left( 1 - \left( 1 - r_T^{3} \right) \left( 1 - r_T^{15} \right) \right) \right) + \left( 1 - r_T^{12} \right) \left( 1 - \left( 1 - r_T^{3} \right) \left( 1 - r_T^{15} \right) \right) \right) \right)$$

(14)

Fig. 28. Terminal reliability RBD of $16 \times 16$ ASEN.

Fig. 29. Simplified terminal reliability RBD of $16 \times 16$ ASEN.

As previously mentioned, for larger ASEN networks, a RBD for network size $N \times N$ can be derived from the RBDs in Figs. 25 and 28. The terminal reliability RBD for $N \times N$ ASEN is shown in Fig. 30.

Fig. 30. Terminal reliability RBD of $N \times N$ ASEN.

According to the RBD of $N \times N$ ASEN in Fig. 30, its terminal reliability is given by the equation (15).
According to the ASEN-RL structure (shown in Fig. 14), its terminal reliability RBD for network size N×N is depicted in Fig. 31. According to this figure, we have:

\[
R_t(ASEN) = r_{2}^{7} \left( 1 - \left( 1 - r_{2}^{3} \right)^{2} \right) + \left( 1 - r_{2}^{15} \left( 1 - r_{2}^{3} \left( 1 - r_{2}^{4} \left( \log_{2} N - 2 \right) \left( 1 - r_{2}^{9} \left( \log_{2} N - 2 \right) \right) \right) \right) \right)
\]

(15)

From equation (16), the terminal reliability of N×N ASEN-RL is given by:

\[
R_t(ASEN-RL) = 1 - \left( 1 - r_{2}^{7} \left( 1 - \left( 1 - r_{2}^{9} \left( \log_{2} N - 3 \right)^{3} \right) \right)^{2} \right)
\]

(17)

With regard to equations (6) and (9), terminal reliability analyses of networks of SEN+ and SEN-RL for three different switch reliability, low switch reliability (r=0.9), middle switch reliability (r=0.95), and high switch reliability (r=0.99) are shown in Fig. 32.
As it can be obtained from Fig. 32, in the case $r=0.99$, SEN+ and SEN-RL have a very close performance in terms of reliability. In fact, it is anticipated that most fault-tolerant MINs have an acceptable performance, in high switch reliability, but a MIN would be the best choice compared with other MINs if it can achieve acceptable performance even in conditions of conflict and middle or low switch reliability. In the cases of $r=0.95$ and $r=0.9$, it is obvious that SEN-RL has a significant advantage in comparison with the SEN+ especially in large network sizes. In fact, the main reason for the superiority of the SEN-RL is due to the exploitation of more number of paths between any source-destination pairs compared with SEN+. As the number of communication paths between source-destination pairs increases in a network, it can lead to the creation of more switching components at the network in parallel, from the viewpoint of reliability. As a result, the network can continue to operate even if there are some of defects in the switching components. Overall, it can be concluded that the proposed method of rearranging links is very effective in improving the reliability of MINs.

According to equations (13), (14), (15), and (17), the results of terminal reliability analysis of ASEN and ASEN-RL for three different switch reliability, low switch reliability ($r=0.9$), middle switch reliability ($r=0.95$), and high switch reliability ($r=0.99$) and as a function of network size are shown in Fig. 33.
As Fig. 33 illustrates, the ASEN-RL has a significant advantage compared to the ASEN in terms of terminal reliability, particularly in middle and high switch reliabilities. Also, in all the switch reliabilities, as the network size increases, the ASEN network shows a sharp drop in reliability compared to the ASEN-RL. In fact, these results indicate that the ASEN-RL network has a very suitable reliability yet it can maintain the high performance even in large network sizes. In addition, it can be concluded that the performance of ASEN gradually becomes worse with increased network size. The reason for the weakness of the ASEN in larger network sizes compared with the ASEN-RL is its structure. As previously mentioned and according to Figs. 25 and 28, as the network size increases, network complexity of ASEN also sharply increases. This leads to a drastic reduction in the reliability of the network, especially in large network sizes [21]. On the other hand, the main reason for the increasing network complexity of the ASEN is due to the use of loops between switch pairs in stages 1 to $((\log_2 N) - 2)$. The main purpose of the loops is to provide multiple paths between each source-destination pair. However, the use of the loops has led to an increase in size of the switches in stages 1 to $((\log_2 N) - 2)$ from 2x2 to 3x3. Also, care must be taken that the increase in size of the switches leads to a reduction in their reliability. Therefore, although the approach of using the loops can be fruitful in small network sizes, it can significantly reduce reliability of the network in large network sizes. Because the number of 3x3 switches will increase in larger network sizes, as a result, these switches, whose reliability is less than 2x2 switches, can reduce the reliability of the entire network of ASEN compared with ASEN-RL. Therefore, in practice, ASEN-RL has a very high merit, compared with ASEN for use in large-scale systems.

Moreover, it can be useful to compare all four networks of the SEN+, SEN-RL, ASEN, and ASEN-RL in a figure. Therefore, the results of terminal reliability analysis of four networks as a function of the network size for middle switch reliability ($r=0.95$) is shown in Fig. 34. As this figure shows, ASEN-RL obtains the best results in terms of terminal reliability than the other networks, especially in large network sizes. Also, After the ASEN-RL, the best results from network sizes 64 to 1024 is owned by the SEN-RL.

Overall, the results obtained in this section demonstrate that the proposed method of rearranging links can be very effective in improving the critical parameter of reliability in MINs. However, another goal of designing an efficient interconnection network is satisfying cost parameter. Therefore, we will focus in the next section on this parameter. As it will be demonstrated in the next section, the proposed method of rearranging links is also a low-cost approach, in addition to attain a high performance in terms of reliability.
4.2. Cost

Cost is an important metric in the design of MINs, so that if a high-performance network have a high cost, then it would not be a suitable case for implementation. According to previous works [14, 22, 23, 25, 26, 28-30], a common method for estimating the cost of a MIN is calculating the number of crosspoints used in the network. Therefore, the cost is given by the number of crosspoints within a switching element and by the number of switching elements within the network. For example, a switch of size 2×2 has four units of hardware cost and a switch of size 3×3 has nine units of hardware cost. Also, it is assumed that each of m×1 multiplexers or 1×m demultiplexers have m units of cost.

It should be mentioned here that the number of links is also predicted in the assessment of cost. In MINs, the links are used to connect switching elements to each other and terminal nodes. Therefore, increasing the number of links leads to an increase in the size of the switching elements. On the other hand, increasing the size of the switches will directly lead to an increase in the crosspoint cost.

Therefore, the cost of MINs can be calculated by the following equation:

\[(NOM \times \text{the cost of a MUX}) + (NOD \times \text{the cost of a DMUX}) + ((NOS \times \text{the cost of a SE}) \times \text{number of stages})\] (18)

In this equation, NOM, NOD, NOS are the number of MUXs, DMUXs, and switches, respectively. In many cases, networks include switches of different sizes, therefore, value of \((NOS \times \text{the cost of a SE}) \times \text{number of stages}\) must be calculated separately for each switch. For example, consider the ASEN network, we have:

\[(NOM \times \text{the cost of a MUX}) = N \times 2 = 2N\] (19)

\[(NOD \times \text{the cost of a DMUX}) = N \times 2 = 2N\] (20)

\[\left(\left(\frac{N}{2} \times 9\right) \times (\log_2 N) - 2\right) = \frac{9N}{2} (\log_2 N) - 2\] (21)

\[\left(\left(\frac{N}{2} \times 4\right) \times (\log_2 N) - 2\right) = \frac{9N}{2} (\log_2 N) - 2\] (22)

Therefore, according to equation (18), the ASEN network cost is given by:

\[Cost(ASEN) = 6N + \frac{9N}{2} (\log_2 N) - 2\] (23)
Likewise, the cost function can be calculated for the other networks as follows:

**ASEN-RL:**

\[(NOM \times \text{the cost of a MUX}) = N \times 4 = 4N\]  
\[(NOD \times \text{the cost of a DMUX}) = N \times 2 = 2N\]  
\[\left(\frac{N^2}{2} \times \text{the cost of a SE2} \times \text{number of stages}\right) = \left(\frac{N^2}{2} \times 4\right)\left((\log_2 N) - 1\right) = 2N((\log_2 N) - 1)\]  

Therefore, according to equation (18), the ASEN-RL network cost is given by:

\[\text{Cost}(ASEN - RL) = 6N + 2N((\log_2 N) - 1)\]  

**SEN+:**

\[\left(\frac{N^2}{2} \times \text{the cost of a SE2} \times \text{number of stages}\right) = \left(\frac{N^2}{2} \times 4\right)\left((\log_2 N) + 1\right) = 2N((\log_2 N) + 1)\]  

Therefore, according to equation (18), we have:

\[\text{Cost}(SEN +) = 2N((\log_2 N) + 1)\]  

**SEN-R:**

\[(NOM \times \text{the cost of a MUX}) = N \times 2 = 2N\]  
\[\left(\frac{N^2}{2} \times \text{the cost of a SE2} \times \text{number of stages}\right) = \left(\frac{N^2}{2} \times 4\right)\left((\log_2 N) + 1\right) = 2N((\log_2 N) + 1)\]  

Therefore, according to equation (18), the SEN-R network cost is given by:

\[\text{Cost}(SEN - RL) = 2N + 2N((\log_2 N) + 1)\]  

According to the above equations, the cost for each network are given in table 3.

<table>
<thead>
<tr>
<th></th>
<th>ASEN</th>
<th>ASEN-RL</th>
<th>SEN+</th>
<th>SEN-R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>[6N + \frac{9N}{2}((\log_2 N) - 2)]</td>
<td>[6N + 2N((\log_2 N) - 1)]</td>
<td>[2N((\log_2 N) + 1)]</td>
<td>[2N + 2N((\log_2 N) + 1)]</td>
</tr>
</tbody>
</table>

With regard to the table 3, the results of the cost analysis for the ASEN, ASEN-RL, SEN+, and SEN-R as a function of the network size is shown in Fig. 35. From this figure, all four networks have a similar cost in the small sizes of 8 to 32. However, as the network size increases, the differences become more apparent. As Fig. 35 shows, the least costly and most costly networks are SEN+ and ASEN, respectively. Also, ASEN-RL and SEN-R are quite similar in terms of cost. As can be seen, although SEN-R is slightly more expensive than the SEN+ network, their cost is very close together and there is very little difference between the two in terms of cost. Therefore, according to high performance of SEN-R in terms of reliability compared to SEN+, this is a perfectly acceptable low cost. In comparing the ASEN and ASEN-RL, it is clear that ASEN have a higher cost than ASEN-RL. Therefore, in this case, the proposed method of rearranging links has prompted to reduce cost, in addition to improving performance in terms of reliability.
Altogether, according to the arguments made in sections 4.1 and 4.2, we can conclude that the proposed method of rearranging links can improve the MINs in terms of reliability, especially for use in large-scale systems. In addition, it has two effects on networks in terms of cost: (1) It increases the cost only a small amount, which is affordable and reasonable. (2) It sometimes reduces the cost, which is very extraordinary.

5. Conclusion

Multiprocessor systems like many other systems need reliability as a vital parameter. On the other hand, the performance of interconnection network used in these systems is very important in the performance of the entire system. Therefore, multiprocessor systems need to reliable interconnection networks. MINs have a great potential in terms of cost and performance. In addition, so far, great efforts to improve the reliability of these networks have been made. However, cost is a engineering constraint during reliability improvement. The cost parameter is very important such that not all the proposed schemes for reliability improvement are reasonable in practice. Therefore, in this paper, our aim was to provide a creative approach, named rearranging links, for improving the reliability of MINs in such a way it satisfies the cost requirement. Almost all the previous works were based on creating redundancy in the number of switching components for improved reliability, which led to skyrocketing cost. However, in this paper, the basic idea of the proposed method is based on reorganizing the pattern of links between stages. To prove the efficiency of the proposed method in terms of reliability, first, the proposed method was implemented on two known MINs namely SEN+ and ASEN. Then, a deeply analysis of terminal reliability as one of the important dimensions of MINs reliability was performed. Reliability analysis results confirmed that the proposed method has a very positive impact on improving reliability of MINs, particularly for use in large-scale systems. For instance, the results demonstrate that the proposed method improves the reliability of the 1024×1024 SEN+ for low switch reliability (r=0.9), middle switch reliability (r=0.95), and high switch reliability (r=0.99) by 26%, 9%, and 0.24%, respectively. Also, the proposed method improves the reliability of the 1024×1024 ASEN for low switch reliability (r=0.9), middle switch reliability (r=0.95), and high switch reliability (r=0.99) by 98%, 52%, and 2%, respectively. Furthermore, the cost analysis also demonstrates that the results derived from rearranging links are truly extraordinary in terms of cost-effectiveness; the results showed that the proposed method has two possible effects on the networks in terms of cost: (1) It increases the cost only to a small extent, which is a reasonable cost. For instance, in the case of the 1024×1024 SEN+ network, a 9% increase occurs in the cost of the network. (2) It can sometimes reduce the cost, which is very extraordinary. For instance, in the case of the 1024×1024 ASEN network, a 75% reduction occurs in the cost of the network. Overall, conducted analyzes are indicative of the fact that the proposed method has a good performance in terms of both vital parameters of reliability and cost. In
addition, given that the proposed method was implemented on two different topologies, the proposed method has the potential to implement on the various MINs as a general approach. Therefore, future works could be focused on the use of this method in other networks to achieve higher levels of performance in terms of reliability and cost.

References