Publications

Available Online

J. Sci. Res. **1** (1), 43-54 (2009)

JOURNAL OF
SCIENTIFIC RESEARCH

www.banglajol.info/index.php/JSR

On Number of Planes of Rearrangeably Nonblocking Optical Banyan Networks with Link Failures

Basra Sultana and Md. Mamun-ur-R. Khandker[1]

Department of Applied Physics and Electronic Engineering, University of Rajshahi, Rajshahi-6205, Bangladesh

Received 27 August 2008, accepted in final revised form 19 December 2008

Abstract

Vertically stacked optical banyan (VSOB) networks are attractive for serving as optical switching systems due to the desirable properties (such as the small depth and self-routing capability) of banyan network structures. Although banyan-type networks result in severe blocking and crosstalk, both these problems can be minimized by using sufficient number of banyan planes in the VSOB network structure. The number of banyan planes is minimum for rearrangeably nonblocking and maximum for strictly nonblocking structure. Both results are available for VSOB networks when there exist no internal link-failures. Since the issue of link-failure is unavoidable, we intend to find the minimum number of planes required to make a VSOB network nonblocking when some links are broken or failed in the structure. This paper presents the approximate number of planes required to make a VSOB networks rearrangeably nonblocking allowing link-failures. We also show an interesting behavior of the blocking probability of a faulty VSOB networks that the blocking probability may not always increase monotonously with the increase of link-failures; blocking probability decreases for certain range of link-failures, and then increases again. We believe that such fluctuating behavior of blocking probability with the increase of link failure probability deserves special attention in switch design.

Keywords: Banyan networks; Blocking probability; Switching networks; Vertical stacking; Link-failures.

DOI: 10.3329/jsr.vlil.1070

1. Introduction

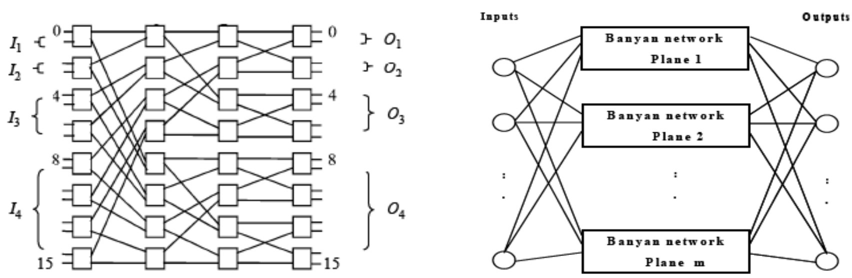
It is expected that users of telecommunication services such as internet, web browsing and distant-education will increase dramatically in future. This will greatly increase the demand for high bandwidth and high capacity communication systems. Optical mesh networks are considered more capacity-efficient and survivable for serving as the backbones for the next generation internet which will be able to handle such huge bandwidth. A key network element of optical mesh networks is the optical switch, which has the capability of switching huge data at ultra-high speed. The main factors those have to be considered

[1] *Corresponding author*: khandker@ru.ac.bd

while designing any optical switching networks are hardware cost, blocking probability, crosstalk, switching speed etc.

A large-scale optical switch is usually composed of numerous basic switching elements (SEs) grouped in multiple stages along with the optical links arranged in a specified interconnection pattern. The basic SEs and the interconnecting optical links in an optical switching device will perform a pre-defined switching function such that the optical flow at an input can be transported to a specific output of the switch. Here we refer to the interconnection pattern of the optical links, the basic SEs and the input/output ports of the switch, as the network of optical switches. The basic 2×2 SE in optical switching systems is usually a directional-coupler (DC) that is made of two waveguides close to each other [1-3]. DC's can switch multiple wavelengths at the same time, and also at high speed (switching time in the order of ns), which is important for the future optical cross-connects (*OXCs*). It is notable that DC suffers from an intrinsic crosstalk problem [1, 4], in which a portion of optical power in one waveguide of a DC will be coupled into the other waveguide unintentionally when two input optical flows pass through the DC at the same time no matter it is in a BAR or a CROSS status. This undesirable coupling effect is called first-order crosstalk, which may propagate downstream stage by stage, leading to a higher order crosstalk in each downstream stage with a decreasing magnitude. A cost-effective solution to the crosstalk problem is to make sure that only one signal passes through a DC at a time such that the first-order crosstalk can be eliminated.

Banyan type (e.g. banyan, baseline, omega, shuffle-exchange etc.) networks [5-8] are a class of attractive switching structures for constructing DC-based optical switches, because they have a smaller and exact same number of SEs along any path between an input-output pair such that absolute loss uniformity and smaller attenuation of optical signals are guaranteed in this class of switching networks. A typical $N \times N$ banyan network consists of $\log_2 N$ stages, each containing $N/2$ 2×2 switches, and the link connections between adjacent stages are implemented by recursively applying the butterfly interconnection pattern as shown in Fig. 1a.



(a) (b)

Fig. 1. Illustration of VSOB networks: (a) a 16×16 banyan network, (b) a VSOB network.

However, the banyan topology has only a unique path from each network input to each network output, and cannot connect all the inputs to all the outputs at any time, for which the network is simply degraded as a blocking one. To deal with this situation, it is an effective approach to make the whole network nonblocking by vertically stacking multiple copies of an optical banyan network [9]. This class of networks is called vertically stacked optical banyan (VSOB) networks (Fig.1b). Numerous studies have been reported for the VSOB networks [10-13] with the focus upon the determination of the minimum number of planes required for achieving the nonblocking characteristic. These studies showed that the adoption of the vertical stacking scheme, although attractive, will significantly increase the hardware cost.

Jian *et al.* [14] in their paper focused on determining the minimum number of stacked copies (planes) required for a nonblocking VSOB networks without link failures if packing strategy is used for routing a request to a plane. It has been shown that required number of planes is minimum when the VSOB network is rearrangeably nonbloking.

Due to the increasing importance and requirement for fault-tolerance in optical switches for large mesh WDM networks, performance analysis on VSOB networks at the presence of probability of link failures becomes critical for the practical adoption of the VSOB networks in the current internet applications. The blocking probability of VSOB networks having link failures has been determined by X. Jiang *et al.* [16]. However, to the best of our knowledge, no results on the issue of number of planes for making a VSOBN rearrangeably nonblocking have been published. When there are no link-failures in the switch network, the lower bound on number of planes required to make the network nonblocking is the same as that required for a rearrangeably nonblocking network. However, when there exist some link-failures in the network, it is a challenging job to find the number of planes required to make the network rearrangebly nonbloking. Proof of rearrangeably nonoblockingness requires a routing algorithm which can make the network nonblocking allowing rearrangement of the existing connections. In this case the existing routing algorithm (Euler's split algorithm) is not applicable. Therefore, we use the packing strategy for routing signals through the network in the simulation. In packing strategy a banyan plane (of a VSOBN) is packed with maximum number of connections so that all N connections can be established using minimum number of banyan planes. Since new connections are accommodated by rearrangement of the existing connections, the results achieved in this algorithm will be a close approximation of the actual number of planes to make VSOB networks rearrangeably nonblocking when some links fail or are broken in a banyan network.

We discuss the published work on VSOB networks briefly in section 2; especially the results of lower bound on number of planes required to make a VSOB networks nonblocking when no failed or broken links exist in the switch network. Section 3 presents our contribution. We conclude this paper in section 4.

2. VSOB Network

Based on the vertical stacking scheme, the conditions for a banyan-type network to be rearrangeably nonblocking (free of first-order crosstalk in SEs; we refer to this as crosstalk-free hereafter) have been determined [11, 13].

$$p \geq \begin{cases} \sqrt{N} & \text{if } \log_2 N \text{ even} \\ \sqrt{2N} & \text{if } \log_2 N \text{ odd} \end{cases} \quad (1)$$

The blocking probability of a VSOB networks is defined as the probability that a feasible connection request is blocked, where a feasible connection request is a connection request between an idle input port and an idle output port of the network. Due to the topological symmetry, all paths in banyan networks have the same property in terms of blocking. Without loss of generality, we focus on the path between the first input and the first output (which is termed as the *tagged path* hereafter). All the SEs and links on the tagged path are called *tagged SEs* and *tagged links* respectively. The stages of SEs are numbered from left (stage 1) to right (stage $\log_2 N$) and the stages of links are also numbered from left (stage 1) to right (stage $\log_2 N+1$). For the tagged path, an input intersecting set $I_i = \{2^{i-1}, 2^{i-1}+1, \ldots, 2^i-1\}$ at stage i is defined as the set of all inputs that intersect a tagged SE at stage i. Likewise, an output intersecting set $O_i = \{2^{i-1}, 2^{i-1}+1, \ldots, 2^i-1\}$ associated with stage i contains all the outputs that intersect a tagged SE at stage $\log_2 N - i + 1$.

In VSOB networks, blocking happens when two connections intend to use the same link or SE. However, if a connection is not allowed to pass through a SE to avoid cross-talk in the signal, it automatically resolves the link contentions. That means, a crosstalk free network will be free from link blocking.

2.1. *Lower Bound on Number of Planes Without Link-Failure*

To route a feasible connection through a VSOB network, a routing algorithm must be adopted to find a path from an input port to an output port. To get the lower bound on the blocking probability (i.e. the minimum possible blocking probability) of VSOB network, Jiang *et.al*. [14] have considered a packing strategy. Under the packing strategy for a VSOB network, a connection is realized on a path found by trying the least free plane of the network first and most free plane last [15]. This routing strategy only guarantees that each of these requests that block a tagged SE will be in a distinct plane in such a way that the minimum number blocked planes can be achieved. If no plane can satisfy the connection request of the tagged path, this request is blocked. This also gives the minimum number of planes required to make the network nonblocking (in other words rearrangeably nonblocking).

The lower bound on the number of planes required to make VSOB networks nonblocking without link-failures is shown in Fig. 2. It is interesting to see that the minimum number of planes required to make a VSOB networks without link failures nonblocking is the same as that required to make the VSOB networks rearrangeably nonblocking. This has been possible because, in packing algorithm, during the routing of new connections, all existing connections are cleared and re-established with new connections altogether. Therefore, some rearrangements of existing connections are taking place in the packing algorithm.

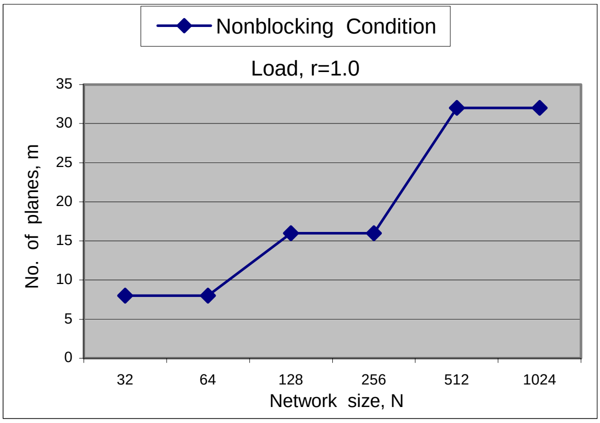


Fig. 2. Minimum number of planes to make VSOB networks nonblocking without link-failure.

3. VSOB Networks with Link Failures

A connection request may also be blocked by a link failure in a faulty VSOB network, which is referred to as the *failure-blocking*. We assume that links in VSOB networks may fail independently and these failures are permanent. Thus, both crosstalk-blocking and failure-blocking should be fully considered in the blocking analysis of a faulty VSOB networks as illustrated in Fig. 3 for a 8×8 network.

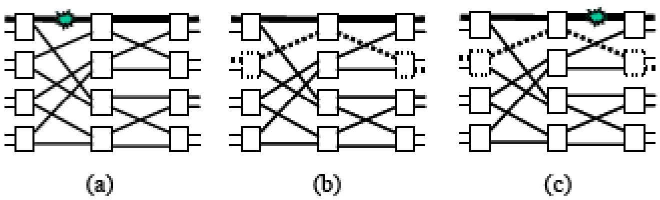


Fig. 3. Blocking in a VSOB network: (a) Failure-blocking, (b) Crosstalk-blocking, (c) Combination of failure-blocking and crosstalk-blocking.

3.1. *Minimum Number of Planes for Rearrangeably Nonblocking Networks*

The rearrangeably nonblocking condtion on number of planes allowing link-failures means the minimum number of planes required to make the network nonblocking when some links are failed in the network. The number of planes required to make the network nonblocking naturally increases if some links in the network are failed or broken. In our simulation we verify that the minimum number of planes required to make a VSOB networks nonblocking without link failures (i.e. when the link failure parameter, *pfr* = 0) is the same as that required to make the VSOB networks rearrangeably nonblocking found in the literature. From this result we assume that our simulated minimum number of planes for the case when link failures are taken into account is also a close approximation to the actual number of planes to make a VSOB networks rearrangeably nonblocking having link-failures. The network simulator we developed consists of six major modules as shown in Fig. 4.

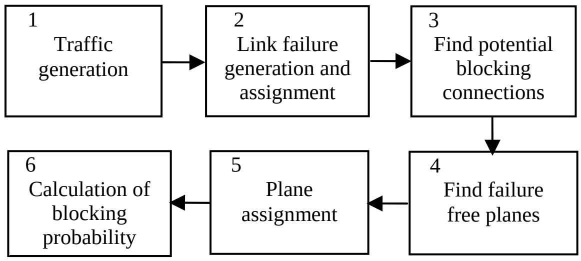


Fig. 4. Block diagram of the network simulator.

We consider here the permutation request as the traffic. A permutation request is such a pattern of requests in which no two inputs request for the same output. Therefore, blocking, if happens, will be due to the internal link-blocking of the switch structure. Due to the symmetric architecture of VSOB(N, T) network, every connection request has the same probability to be blocked. In our simulation, we fix the connection request of input-output pair 0-0 and investigate the blocking probability of this connection request only that may result by other contentious connections. The traffic generation module randomly generates a permutation request for the VSOB(N, T) network based on the workload r (here workload r is defined as the occupancy probability of a port). The link failure generation and assignment module generates link failures based on the given *pfr* (here *pfr* is defined as the probability that a link is failed or broken) and then assigns those failures randomly to different links.

In this section we present some definitions which are used in the discussion of the simulation procedure:

Definition 1. Set of blocking connections, C_{bc}: The connections that are potential to blocking the tagged path.
Definition 2. Set of failure free planes, P_{bc}: The planes that are free from link-failures on the path of each C_{bc}.
Definition 3. Least free plane: It is defined as the plane which is able to establish minimum number of C_{bc}. For example, let plane $|P_{bc}| = 2$, plane 1 be failure free for 5 C_{bc} and plane 2 be failure free for 2 C_{bc}. Then plane 2 is the least free plane.

Module 1 generates random permutations. Each permutation request pass through all the subsequent modules, and the last module checks if the permutation reaches the outputs successfully. Module 2 generates failure pattern with a given probability. These failures are assigned randomly to different links in the switch networks. The next module finds C_{bc} which is determined by the following relation:

$$I_i + O_j < \log_2 N \quad (2)$$

Here, input i is originated from input intersecting group I_i and destined to output j that belongs to output intersecting group O_j (see Fig. 1).

Module 4 checks all the planes if there is a failure on the tagged path, and make a list of planes, say P_{tagged}, in which no links on the tagged path is failed or broken. Then it constructs the list P_{bc} from P_{tagged} and sorts it in the ascending order such that the first plane in the list is least free plane by C_{bc}.

The plane assignment module (module 5) attempts to assign connection requests to different planes using the packing strategy. Plane assignment module groups the input and output connections as per Eqs. (3) and (4).

$$G_i = \begin{cases} i/\sqrt{N} & \text{if } \log_2 N \text{ even} \\ i/\sqrt{2N} & \text{if } \log_2 N \text{ odd} \end{cases} \quad (3)$$

$$G_o = \begin{cases} o/\sqrt{N} & \text{if } \log_2 N \text{ even} \\ o/\sqrt{2N} & \text{if } \log_2 N \text{ odd} \end{cases} \quad (4)$$

Here G_i is the input group for input i and G_o is the output group for output o. If $X(i_1,o_1)$ and $Y(i_2,o_2)$ are two connection requests arrive at inputs i_1,i_2 respectively and destine to o_1,o_2 respectively, then their path is completely disjoint if $G_{i1} \neq G_{i2}$ and $G_{o1} \neq G_{o2}$. Therefore, requests X and Y arrive at the output successfully [17].

The plane which is least free plane (that is the first plane in the sorted list of P_{bc}) for C_{bc} is assigned first to a connection chosen from its list. As soon as the plane is assigned it is marked as ‘busy’ for the input and output groups of that connection’s input and output. The other blocking connections belonged to these input and output groups are not assigned to this busy plane. However this plane can be assigned to a connection having different input and outputs groups. Then the second plane from the sorted list is picked and one or more blocking connections are assigned to the plane as mentioned above. This plane assignment algorithm ensures the maximum use of a banyan plane; thereby ensures the use of minimum number of planes for routing a permutation request. At last we try to assign the tagged path to a free plane in list P_{tagged}. If no such free plane is left for the tagged path, the connection request pattern is recorded as a blocked connection pattern. The blocking probability is then estimated by the ratio of number of permutation requests in which the 0-0 request is blocked to the total number of permutation requests generated. It is notable that, in the next permutation request, some inputs’ request may be the same the previous one and some may be different. However, all previous connections are cleared first and then the next permutation is routed.

The simulation results for the minimum number of planes of VSOB networks with link failures to be nonblocking are given below.

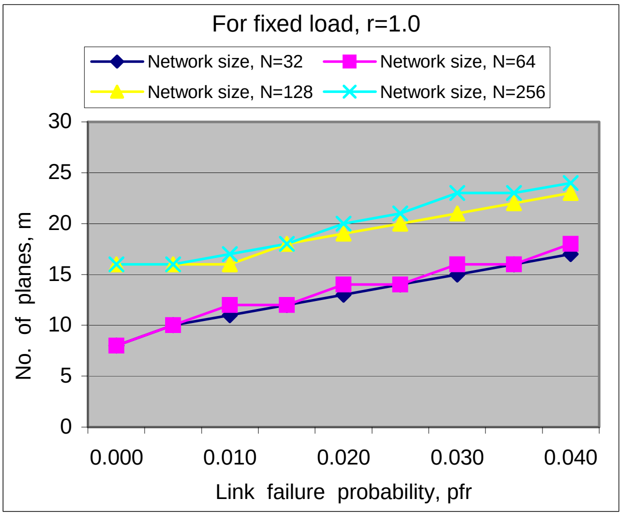


Fig. 5. Minimum number of planes to make VSOB networks nonblocking.

Fig. 5 shows the minimum number of planes required to make a VSOB network nonblocking with a given link failure probability. For *N*=128 and *pfr*=0.01; the minimum number of planes required to make a VSOB networks nonblocking is the same as that required to make the VSOB networks rearrangeably nonblocking. Therefore, we can say that failure-free rearrangeably nonblocking VSOB networks can accommodate small link failures without degrading its performance.

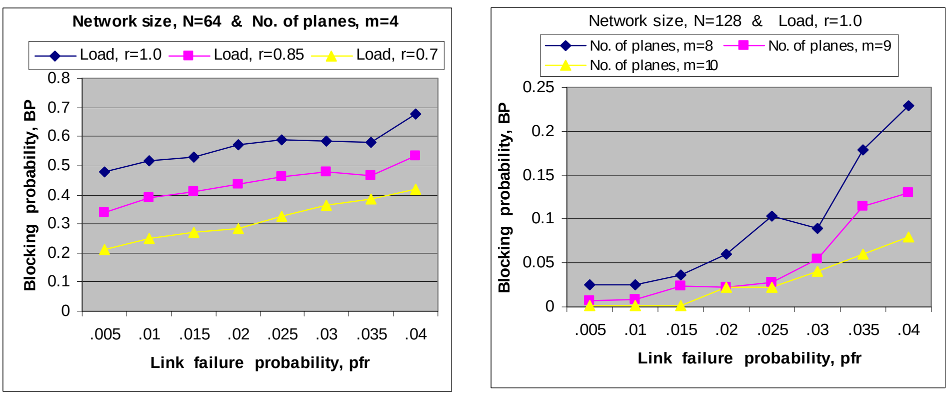


Fig. 6. Blocking probability decreases for certain range of link-failures and then increases again: (a) Effects of number of planes, (b) Effect of work load.

It is also interesting to note from Fig. 6 that the blocking probability not always increases with the increase of link failures; blocking probability decreases for certain range of link failures and then increases again. The reason for decreasing the *BP* may be as follows. When there are some links broken on the path of potential blocking connections, they can not interfere with the tagged path. This phenomenon increases the tagged path's chance of being successful. However, this effect is more prominent for fewer number of planes and lower workloads, and gradually diminishes with increase of these two parameters.

3.2. *Worst Case Scenario*

There are some permutations which require more planes to be routed than others. The worst case scenario is that when all inputs from group G_i are destined to G_o (see Eq. 3 and 4); the minimum number of planes needed to realize this permutation is min{$|G_i|$,$|G_o|$}. In the above simulation, the traffic generation module randomly generates a permutation request. In this set of permutations, the probability of generating such worst-case permutations is very small. Following [14], the probability of worst-case scenario is given by

$$\begin{aligned} P_{worst} &= 2.57\times10^{-10} \quad \text{for } N = 64, r = 0.9 \\ P_{worst} &= 2.45\times10^{-20} \quad \text{for } N = 128, r = 0.9 \end{aligned} \tag{4}$$

The above results indicate that the probability of worst-case scenario is very small for both even and odd number of stages. However, the effect of these permutations on the performance of the switching network must be worst, and need special treatment. Therefore, we have generated a subset of all possible worst-case permutations separately and see its effect on the switching networks. The simulation results for that case are given below.

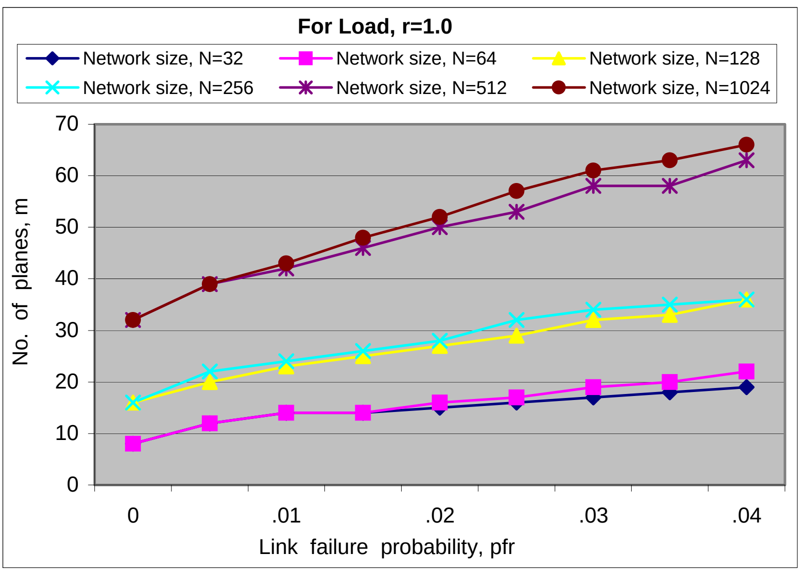
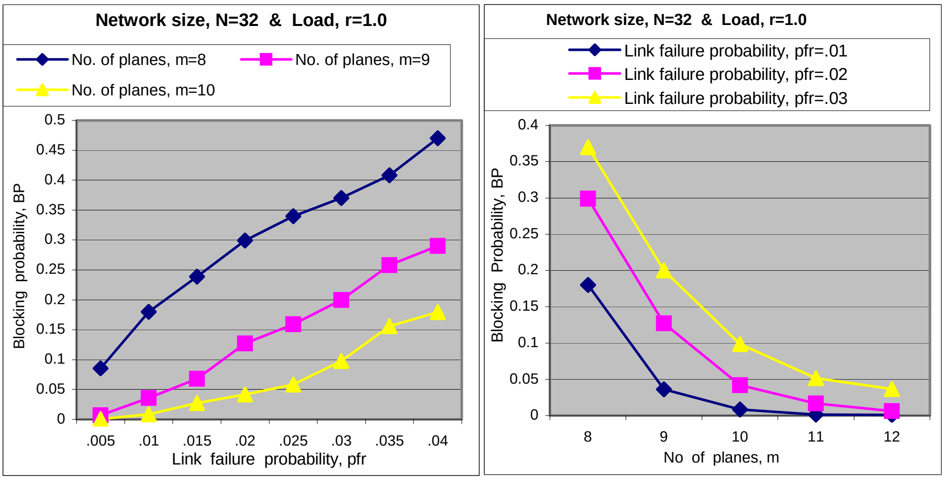


Fig. 7. Minimum number of planes to make VSOB networks nonblocking considering worst-case permutation.

In Fig. 5 we find that for *N*=128, load=1.0 and *pfr*=0.01, 16 planes are required to make VSOB networks nonblocking. But if we consider a subset of all possible worst-case permutation then for the same configuration it requires 23 planes (Fig. 7) to make VSOB networks nonblocking. It is notable that for a particular link failure probability the number of planes required to make VSOB networks nonblocking is significantly higher than the previous result.



(a) (b)

Fig. 8. (a) Blocking probability increases with the increase of link-failures, and (b) Blocking probability decreases dramatically with plane.

Fig. 8a shows the effect of link-failure on blocking probability for a fixed size network and work load. Blocking probability gradually increases with link failure probability but the nature is different for different number of planes. For *pfr*=0.04-0.02; *BP* decrease 76% for *m*=10, *BP* decrease 56% for *m*=9 and *BP* decrease 36% for *m*=8. This will be clearer from Fig. 8b. For *N*=32, *pfr*=0.01, *r*=1.0 if we increase number of planes from 8 to 9 then blocking probability decreases 80%. This picture reveals the fact that if we allow small amount of blocking probability then the number of planes required to make the VSOB network nonblocking decreases dramatically.

The effect on blocking probability with various traffic loads for a fixed network size and link failure probability is shown in Fig. 9a. The effect on blocking probability depends on number of planes; for low value of *m*, blocking probability increases linearly with the increase of traffic load but for large value of *m* the nature turns into nearly exponential. For *N*=64, *pfr*=0.02, *m*=8; if we decrease work load 1.0 to 0.8 then the blocking probability decreases by 49% (Fig. 9b). It also shows that the effect of link-failure on blocking probability does not depend much on work load; only for higher value of *r*, blocking probability is affected nonlinearly at different link-failure probabilities.

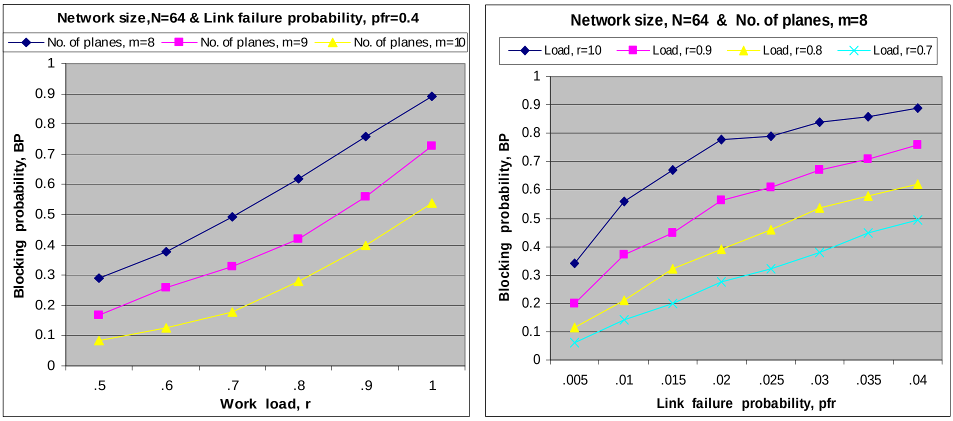


Fig. 9. (a) Effect of load on blocking probability, (b) Effect of link-failures on blocking probability considering worst-case permutation.

4. Conclusion

We have presented a close approximation of minimum number of planes required to make the VSOB networks nonblocking having link-failures. We have shown that VSOB networks can accommodate some link failures without degrading the performance. That means a rearrangeably nonblocking VSOB network can still be nonblocking even when there are some broken or failed links in the structure. The blocking probability does not monotonously increase with the increase of the link-failure probability. The results presented in this paper will help optical switch designer make a trade-off among different performance metrics for given input parameters (e.g. crosstalk, link-failure, blocking probability etc.). Since packing algorithm routes requests sequentially one after another rather instead of searching for the best-fit plane, it produces a close approximation to the actual minimum value. A routing algorithm for making the VSOB networks having link-failures rearrangeably nonblocking is under research. A more detail and mathematical analysis is required in this regard.

Reference

1. H. S. Hinton, An introduction to Photonic Switching Fabrics (Plenum, New York: 1993).
2. H. J. R. Dutton, Understanding Optical Communications (Prentice-Hall, New Jersey, 1998).
3. R. Ramaswami and K. N. Sivarajan, Optical Networks: A Practical Perspective (Morgan Kaufmann, San Mateo, California, 2002).
4. V. R. Chinni, T. C. Huang, P. -K. A. Wai, C. R. Menyuk, and G. J. Simonis, IEEE/OSA Journal of Lightwave Technology, **13 (7)**, 1530 (1995). doi:10.1109/50.400714
5. G. R. Goke and G. J. Lipovski, *Proc.1st Annual. Symposium Computer Architecture* (1973) pp. 21-28.
6. C. P. Kruskal and M. Snir, IEEE Transaction on Communications **COM-32**, 1091 (Dec. 1983).
7. F. T. Leighton, Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes (Morgan Kaufmann, California, 1992).
8. J. H. Patel, IEEE Transaction on Computers **C-30**, 771 (Oct. 1981). doi:10.1109/TC.1981.1675695

9. C. -T. Lea, IEEE Transaction on Communications **38** (10), 1740 (Oct. 1990). doi:10.1109/26.61445
10. G. Maier and A. Pattavina, IEEE Transaction on Communications **49** (7), 1268 (July 2001).
11. M. M. Vaez and C. -T. Lea, IEEE Transaction on Communications **48** (2), 316 (Feb. 2000).
12. M. M. Vaez and C. -T. Lea, IEEE J. Selected Areas on Communications **16**, 1327 (Sept. 1998).
13. Xiaohong Jiang, Md. Mamun-ur-Rashid Khandker, Hong Shen, and Susumu Horiguchi, IEICE Transaction on Communications **E86-B** (12) (Dec. 2003).
14. X. Jiang, H. Shen, Md. Mamun-ur-R. Khandker, and S. Horiguchi, IEEE/ACM Transactions on Networking **11** (6) (Dec. 2003).
15. X. Jiang, Md. Mamuan-ur-R. Khandker, and S. Horiguchi, *Proc. 2001 IEEE Workshop on High Performance Switching and Routing* (May 2001) pp. 307-311.
16. X. Jiang, P. H. Ho, H. Shen, and S. Horiguchi, *Proc. the 2004 IEEE Workshop on High Performance Switching and Routing*.
17. M. M. R. Khandker, X. Jiang, P. -H. Ho, and S. Horiguchi, Optics Communications **259** (2), 517 (2006).