Dynamic Lightpath Protection in WDM optical Networks Using Ant-based Mobile Agents

Son Hong Ngo
Graduate School of Information Science
JAIST, Japan
Email: sonhong@jaist.ac.jp

Xiaohong Jiang and Susumu Horiguchi
School of Information Science
Tohoku University, Sendai, Japan
Email: {jiang, susumu}@ecei.tohoku.ac.jp

Abstract

In this paper, we consider the problem of dynamic lightpath protection in survivable WDM networks with single link failure model. Our work concerns in how to dynamically determine a protection cycle to establish a dependable lightpath with fault-tolerance requirement. This problem is identified as NP-complete, thus a heuristic for finding near optimal solution with reasonable computation time is usually preferred. Inspired from the principle of ant colony optimization, we develop in this paper an ant-based mobile agents algorithm for this problem with improved blocking performance. To enable the new ant-based algorithm, we propose to use on each network node both a routing table that contains a set of feasible cycles between source destination nodes and also a pheromone table for mobile agents. By keeping a suitable number of mobile agents in a network to continually and proactively update the cycles in a routing table based on the current network congestion state, the routing solution of a connection request can be obtained based on simply a local searching with a reasonable computation time. Extensive simulation results upon the ns-2 network simulator show that our new algorithm can achieve a significantly lower blocking probability than the promising algorithm for dynamic lightpath protection proposed in [11] with a comparable computation complexity.

1. Introduction

Wavelength division multiplexing (WDM) in optical networks has been considering as a promising technology for future backbone networks that can provide huge bandwidth capacity. These networks are viable solution to meet the bandwidth demands from various emerging multimedia applications such that web applications, video on demand, multimedia conference, image access and distribution, home broadband services etc. [1]

A WDM optical network consists of optical cross-connects (OXCs) interconnected by fiber links, in which an OXC can switch an optical signal from an input fiber to an output fiber without performing optoelectronic conversion. In wavelength-routed optical networks, end-users communicate with each other via all-optical channels, which are referred to as *lightpaths* as shown in Fig 1. A lightpath is an optical channel on a wavelength that spans multiple fiber links to provide a connection between two network nodes. By using *wavelength converters*, an OXC is capable of changing the wavelength of incoming signal. However, wavelength conversion cost is very expensive, thus it is preferred to use the OXCs without wavelength converters. In our works, we consider the later case where the same wavelength must be used along a lightpath, which is referred to WDM optical networks with the *wavelength continuity constraint*.

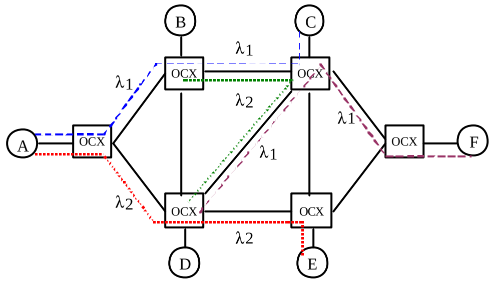


Fig. 1. Architecture of a wavelength-routed network

1.1. Routing and wavelength assignment

Given a set of connection requests, the problem of setting up lightpaths by routing and assigning a wavelength to each connection is called routing and wavelength assignment (RWA) problem [2]. If we cannot setup a lightpath for a connection request, then it is blocked. Generally, a RWA algorithm tries to minimize the total *blocking probability* overall the network. A well-designed RWA algorithm is critically important to improve the performance of WDM networks

RWA problem can be classified into static RWA and dynamic RWA problems. In the static RWA problem, the connection requests are given in advance. The static RWA is always performed offline, the objective is minimized the total blocking probability or to have the maximum number of setting up connections. This problem can be formulated as mixed-integer linear program, which is NP-complete [3]. In contrast, the dynamic RWA considers the case where connection requests arrive dynamically. The dynamic RWA is performed online, it is much more challenging; therefore, heuristic algorithms are usually employed in resolving this problem [2]

Our work mainly focuses on dynamic RWA problem. In literature, there are static routing approaches available for the dynamic RWA problem, such as shortest-path routing or alternate shortest-path routing [4, 5]. These approaches use a set of pre-computed shortest paths for lightpath establishment. Advantage these approaches is its simplicity, e.g. small setup time and low control overhead. Adaptive routing approaches [6] are more efficient than static routing methods in terms of blocking probability, because the route is chosen adaptively depending on the network state. The main problems of adaptive routing methods are their longer setup delay and higher control overhead, including the requirement of global network's state on each node.

1.2. Lighpath protection in WDM networks

The failure in optical communication networks such as accidental link disruption or switching device disorder will affect a huge amount of bandwidth in transmission, thus *survivability* is one of the most important issues in the design of WDM optical networks [7]. Two major techniques to prevent failures are protection and *restoration* [8]. In *protection* schemes, backup resources are pre-computed and reserved for each connection before a failure occurs. In *restoration* schemes, the backup route is dynamically computed after the failure occurs. In compare with restoration schemes, protection schemes have a faster recovery time and can guarantee 100% of recovery ability, but require more network resources.

Protection schemes are divided into *path protection* and *link protection*. In the former, a working path and a disjoint protection path are pre-computed for each connection. In the later, each link of the working path is protected by separate backup resources. Path protection schemes usually require lower backup resources and lower recovery delay than link protection [9]. The pair of working and protection paths forms a *protection cycle* between two network nodes. A connection that is setup from a protection cycle is called a *dependable connection*. Protection schemes can be further classified into *dedicated protection* and *shared protection*. In the former, the backup resources such as links or nodes are used for at most one connection. In the later, the backup resources can be used for multiple connections, because these connections rarely fail simultaneously. Dedicated protection consumes more resource but is simpler to implement. In contrast, share protection is more efficient but more complexfor management [1].

There are two kinds of failure in WDM networks: link failure and node failure. It is observed that most modern switching devices are equipped built-in redundancy to improve their reliability. Therefore, link failure is more concern than node failure. Many studies in the literature justify that single link failure happens much more frequent than multiple link failures, thus the single link failure model attract more attentions in the optical survivability research.

1.3. Motivation and contribution

In this paper, we consider the problem of dynamic routing and wavelength assignment with lightpath protection in WDM mesh networks. We adopt the path protection scheme with shared backup resource. The single link failure model is concerned in our work.

In [10], Mohan et al. proposed an efficient protection scheme called primary independent backup wavelength assignment (PIBWA). This method uses the shared protection scheme by adopting the backup multiplexing technique. In PIBWA algorithm, a set of k link-disjoint paths is pre-computed for every source-destination node pair. Whenever a connection request arrives, a working (primary) path and a backup path with total minimized cost are selected from these K paths. If such working-backup path-pair has no wavelength available then the connection request is blocked. The PIBWA with backup multiplexing technique is simple but can still provide a protection mechanism with efficient network performance in terms of blocking probability.

The main limit of PIBWA method is that it uses a set of fixed alternate link-disjoint routes, so it exists a big space to improve the network performance. Many researchers proposed optimal approaches by formulating this problem as an Integer Linear Program (ILP), thus it is NP-complete [7, 11, 12]. However, it is not practical to solve such ILP problem by optimal approach because the dynamic connection setup requires a low computation time. To achieve that goal, these authors also proposed several heuristics to solve this problem. However, these heuristics have to compute online a set of shortest paths between a node pair based on current link state, thus the computational complexity is still high, which may increase the setup delay of a connection request.

From the above observations, we propose a novel heuristic for this problem by using the principle of ant colony optimization [13]. We inherit the ant-based mobile agents proposed in our previous works [14] to explore the current network congested state. We propose to use both forward ants and backward ants to discover a set of link-disjoint cycles between every node pair in the network. These cycles are always available and updated before the connection request arrives, thus we can easily select the best cycle among them for the connection setup. This method can guarantee a small setup delay because these cycles already exist by the time of connection setup. By following the ant-colony optimization principle, the mobile agents can report to each network node the cycles with low cost to increase the network performance. Extensive simulation results show that with a suitable number of cycles, our proposed heuristic can significantly outperform the PIBWA method in terms of blocking probability.

1.4. Paper's organization

The rest of this paper is organized as follows. Section 2 presents the principle of ant-based network routing and some related works about routing in WDM optical networks using ant colony optimization. In Section 3, the proposed ant-based algorithm for routing in WDM networks with lightpath protection is presented. The results of simulation experiments are described in Section 4. Finally, we conclude by some discussions in Section 5.

2. Background

2.1. Ant-based network routing

Inspired from the behaviors of natural ant system, a new class of ant-based algorithms for network routing is currently being developed. These algorithms use artificial ants concurrently and independently, and communicate in an indirect way to build the routing solution. Previous work has shown the potential success in routing for both packet and circuit switching networks [15-17]. The idea from ABC algorithm [16] could be used in WDM routing because this is circuit-switched routing. In ABC algorithm, the mobile agents modify the routing policy at every node by deposing pheromone trail on routing table, and the agents' goal is to build and to adapt the routing tables to the load changes at run time so that the rate of accepted coming calls is maximized. However, ABC algorithm cannot be applied directly into RWA problem because they do not consider the wavelength continuity constraint in WDM optical networks.

2.2. Related works

Recently, there have been some ant-based approaches for dynamic routing and wavelength assignment problem such as Garlick's algorithm [18] or Ngo's algorithm [14]. These approaches use ant-based agents to explore the network and build the routing solution based on current network congested state. However, to our knowledge, there has been no algorithm using ant-based approach to resolve the RWA problem with lightpath protection, which requires the mobile agents to find concurrently the cycles in WDM networks.

In this work, we extend our ant-based routing (ABR) algorithm in [14] to solve the dynamic lightpath protection problem. We summarize the ABR algorithm as follows. In ABR, a network node is equipped with a probabilistic pheromone table that contains the selection probability of neighbor node when an ant moves toward its destination node. Ants are launched from each node with a given probability to a randomly selected destination every time unit. This algorithm ensures that the information about network congestion is well reflected in the pheromone tables. The route for a connection request is selected directly based on the highest selection probability or the second highest probability, thus the setup time can be reduced. The results show that the ABR algorithm significantly outperforms other alternate methods in terms of blocking probability. Since the ant-based mobile agents in ABR algorithm can efficiently explore the network state, we inherit this property in this paper and use mobile agents to explore cycles in order to establish dependable connections in WDM networks.

3. Fault-tolerate ant-based routing algorithm for dynamic lightpath protection.

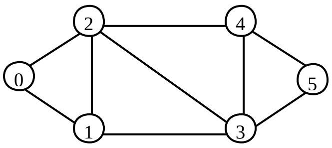
In this section, we will describe our proposed dynamic routing and wavelength assignment algorithm with lightpath protection called FT-ABR (Fault-tolerate ant-based routing). The main idea is to use mobile agents with ant-colony behavior to explore link disjoint cycles between each network node pair. On each node, these cycles are stored in a routing table that is continuously updated by mobile agents. By the time a dependable connection request arrives, the best one among these cycles will be used for connection setup.

We suppose that ant-based mobile agents run in a separated control plane that is carried out in a packet switching network with the same topology as optical network, or in optical domain where control data is transported on a dedicated wavelength [1]. For simplicity, we also suppose that all connection requests are dependable connections. The following subsections will present the FT-ABR algorithm in detail.

3.1. Pheromone table and routing table

In addition to the routing table that contains a list of K feasible cycles to every possible destination for dependable connection setup, each network node now also has a pheromone table for mobile agents as other ant-based algorithm [14].

For a network node i with k_i neighbors, its probabilistic pheromone table $R_i = [\, r^i_{n,d} \,]_{k_i,\ N\text{-}1}$ has N-1 rows (N is number of network nodes) and k_i columns. Each row corresponds to a destination node and each column corresponds to a neighbor node; the value $r^i_{n,d}$ is used as the selection probability of neighbor node n when an ant is moving toward its destination node d. An example of pheromone table is illustrated in Fig 2.



Destination	1	4	5
0	$r^3_{1,0} = 0.6$	$r^3_{4,0} = 0.3$	$r^3_{5,0} = 0.1$
1	$r^3_{1,1} = 0.8$	$r^3_{4,1} = 0.2$	$r^3_{5,1} = 0.0$
…	…	…	…
5	$r^3_{1,5} = 0.0$	$r^3_{4,5} = 0.2$	$r^3_{5,5} = 0.8$

Fig. 2. A simple network and its pheromone table of node 3.

The cycle routing table on each node contains N-1 rows according to every destination node. Each row corresponds to a destination node and contains a list of K cycles to it. Each cycle has a primary path and a backup path that are link-disjoint. This routing table is continually updated by mobile agents based on the current network congestion state, so it always contains the K best cycles candidates for a dependable connection request.

3.2. Mobile agents behavior

The main behavior of ant-based mobile agents (ants) is summarized as follows (Fig.3):

- Ants are launched from each node with a given probability ? to a randomly selected destination every t time units; here ? and t are design parameters. Ant status now is forward ant.
- Each ant is considered to be a mobile agent: it collects information on its trip, performs routing tables updating on visited nodes, then move forward
- Whenever an forward ant visits a node, it updates the pheromone table with the information gathered during its trip
- An ant stochastically determines its next hop according to its selecting probabilities in the pheromone table.
- When an ant reaches the destination, it changes its status to backward ant. Then ant moves back to the source node by the same rule as forward ant. In addition, this backward ant checks not to move on the visited links so that all the visited nodes will formulate a cycle. If an ant cannot select a node to move, it will be killed.
- When an ant gets to the source node, it reports its cycle to the routing table. After that, this ant is killed.

Whenever a forward ant visits a node, it updates the pherom one table element with the information gathered during its trip, include the path's length and the number of idle wavelengths along the path. Suppose a forward ant moves from source s to destination d following the path (s,… i-1, i,…d). When this ant arrives at node i, it updates the entry corresponding to the node s by a reinforcement value (pheromone) as follows: the probability of neighbor i-1 is increased while the probabilities of other neighbors are decreased. The amount of pheromone decreases with the path length and increases with the number of available wavelengths. If an ant visits node i at time t, so the next values for routing entry in next time t+1 are determined as follows:

$$r^i_{i-1,s}(t+1) = \frac{r^i_{i-1,s}(t) + dr}{1+dr} \quad \text{and} \quad r^i_{n,s}(t+1) = \frac{r^i_{n,s}(t)}{1+dr},\ n \neq i-1 \tag{1}$$

Here, dr is the reinforcement parameter or the amount of trailing pheromone, and dr decreases with the paths length and increases with the number of available wavelengths:

$$dr = \frac{a}{dl} + (1-a)*dw \tag{2}$$

Where dl corresponds to the length of the path, dw corresponds to the percent of free wavelengths of this path, a is a scalar parameter that can be used in adjusting the emphasis of path length versus free wavelength percentage. These two factors are computed as follows:

$$dl = b*(e^{\frac{-1.0}{l}} - e^{-1})\,,\ dw = e^{g*w} - e^{0} \tag{3}$$

Where w is the percentage of free wavelengths of the path and w is deducted from the corresponding wavelength mask in ant stack. Here a, b and g are designed parameters and they can be adjusted to get a good system performance.

Whenever a backward ant reaches to its source node, it reports its cycle to the routing table on the source node. If this is a new cycle and there is not enough K cycles, ant will insert the new cycle into the table; otherwise, it will replace a random cycle that has been in the routing table.

The above rules for agent's moving and pheromone tables updating make ant-based mobile agents tend to follow paths with shorter length and larger number of free wavelengths. Thus, the cycles reported by ants will have high possibility to be selected for connection setup.

3.3. Connection setup

We adopt the dependable connection setup scheme using share protection as PIBWA method [10], because the authors have shown that PIBWA method achieves the

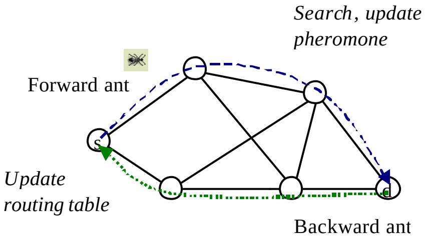


Fig. 3. Ant's moving and updating task

best performance among the protection methods using alternate routes. The major difference is that our FT-ABR algorithm uses a set of cycles dynamically updated by ants based on the network congestion states.

The purpose of this task is to select a minimum cost cycle. The key idea here is to choose the cycle that requires minimum number of extra free wavelength channel. To achieve that goal, we define the cost of primary path $C(L_p)$ and cost of backup path $C(L_b)$ for the wavelength w as follows: The value of $C(L_p)$ is the length of L_p if the wavelength w is free on L_p; otherwise, $C(L_p)$ is infinity. $C(L_b)$ is defined as the sum of cost of all the links on this path. The cost of one link is 1 if w is free on this link. The cost of one link is 0 if the wavelength w on this link is shared with other backup path whose the primary path is link-disjoint with L_p; otherwise, the cost is infinity.

It is noticed that we have to scan all the wavelength w belonging to the set of W wavelengths to compute the cost $C(L_p)$ and $C(L_b)$ for each cycle. The task to select a minimum cost cycle is is summarized as follows:

1) Consider every candidate cycles. For each of W wavelengths, compute $C(L_p)$ and $C(L_b)$ Among the cycles that has at least one wavelength on it $C(L_b)$ is not infinity, select a set of cycles with minimum $C(L_p)$.
2) Among the cycles of this set, select the cycle with minimum $C(L_b)$
3) After above three steps, we get the cycle with minimum cost and the corresponding wavelengths of its primary path and backup path that minimize the cost this cycle. The minimum cost cycle is used to setup a dependable connection for the request.

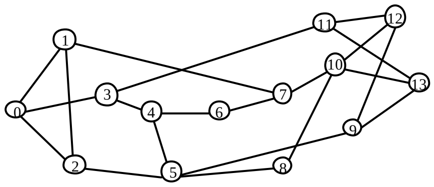
3.4. Complexity analysis

We now compute the computational complexity of FT-ABR for setting up a connection. We do not take into account the complexity of mobile agent's task because they work proactively before a connection request arrives, so they do not affect the setup delay. To compute the cost of backup path for a given primary path, the complexity is $O(H^2)$ where H is the path length, because H could be N-1 in maximum thus the complexity of step 1 is $O(KN^2W)$. The complexity of step 2 and step 3 is only $O(KW)$. Thus the worst-case complexity of FT-ABR become $O(KN^2W)$,which is comparable to the complexity $O(k^2N^2W)$ of the PIBWA method, where k is the number of alternate routes. We will show in section 4 that our new algorithm is more flexible in the sense that we can increase the performance of FT-ABR by increasing K, but we cannot increase k as the average number of alternate routes is limited by average network degree (this value is usually around 3).

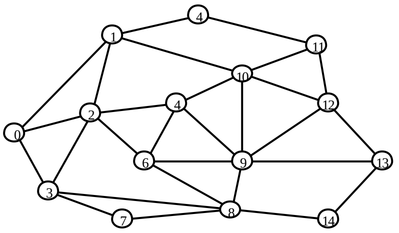
4. Simulation results

To verify the performance of our proposed ant-based algorithm for dynamic lightpath protection, a simulator module was implemented on the open source package Network Simulator ns-2 [19]. A circuit module is implemented to simulate dynamic lightpath routing in WDM networks. The simulation of mobile agents is performed directly by network packets in ns-2. Simulations are carried out upon two typical network topologies: the NSFNet with 14 nodes, 21 links and the TestNet with 15 nodes, 28 links as shown in the Fig. 2.

We adopt a general traffic model widely used in performance analysis of data communication networks. We suppose that every connection is dependable to guaranty 100% of recovering capaccity under single link failure model. The arriving sessions is distributed randomly over the network. The connection requests arrive at each node according to a Poisson process with an arrival rate l (call/s). The session holding time is exponentially distributed with mean m. There are totally T sessions, the total network load is measured by $T*l*m$ (Erlang).



(a) NSFNet



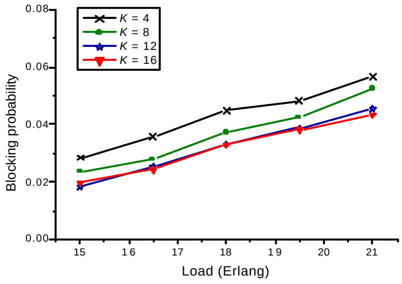
(b) TestNet

Fig. 4. Topologies used in experiments

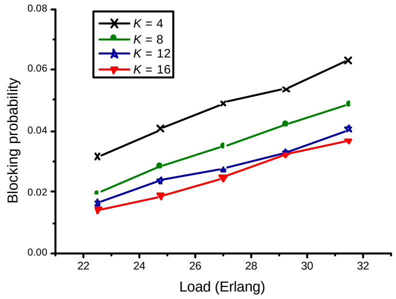
The session holding time m = 1s. For each case, the number of session, the arrival rate are selected to have a reasonable range of traffic load such that the total blocking probability is about 5% - a practical value for WDM networks. The time step for ant's generation is t = 1ms. For simplicity, the ant's launching probability is set as *?* = 1. The others pheromone parameters are set as same as our previous algorithm [14]. To get a stable result, each experiment is conducted with about 10000 requests, it is repeated five times to get the average value of blocking probability.

To find a suitable number of cycles using for the size of routing tables, extensive simulations are performed with different values of K (Fig. 5). Initially the number of cycles is K = 4. When K increases, the blocking probability decreases. However, the blocking probability does not significantly decrease when K is large. We can observe in both network topologies that with K = 12 or 16, the FT-ABR provide the similar performance. With the network topologies using in our experiments (NSFNet and TestNet), the number of cycle K =16 could provide a good performance in terms of blocking probability.

We use the algorithm PIBWA with k = 3 alternate routes for performance comparison. Fig. 6 show the comparison between the FT-ABR algorithm and PIBWA algorithm when the number of cycles is set as K = 16 and the number of wavelengths is set as W = 4. We found that FT-ABR algorithm always significantly outperform the PIBWA algorithm for all the values traffic load we studied. For the NSFNet network with a workload of 15, the blocking probability of PIBWA algorithm is about 0.05 while the blocking probability of our algorithm is only 0.02. For TestNet network, the blocking probability of PIBWA algorithm is about 0.03 while the blocking probability of our algorithm is only 0.01 when workload is 22.5. So our new algorithm achieves a blocking probability that is at least two or three times lower than that of the promising PIBWA algorithm for both the network topologies. These results clearly demonstrate that by using the ant-based mobile agents to explores the networks states, we can reduced dramatically the total blocking probability than using only the pre-computed static alternate routes as in PIBWA algorithm.

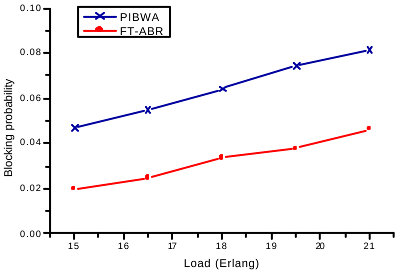


a) NSFNet

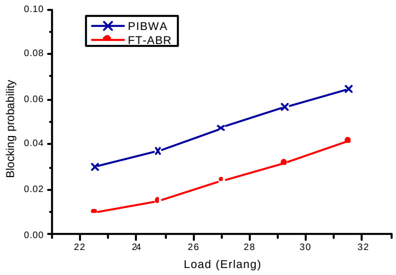


b) TestNet

Fig. 5. Performance of FT-ABR with different numbers of cycles



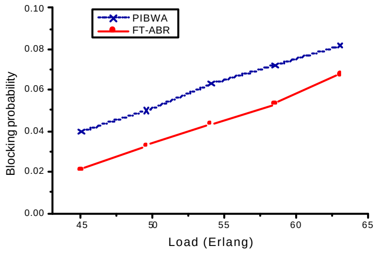
(a) NSFNet



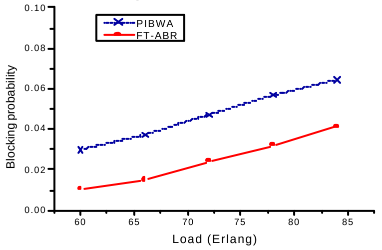
b) TestNet

Fig. 6. Performance of FT-ABR compared with PIBWA for 4 wavelengths

A similar result is also observed in Fig. 7 when the number of wavelengths 8 (W = 8). For the NSFNet, the blocking probability of PIBWA algorithm is about 0.04 while the blocking probability of our algorithm is only 0.02 when workload is 45. For TestNet network, the blocking probability of PIBWA algorithm is about 0.03 while the blocking probability of our algorithm is only 0.01 when workload is 60. For both the cases of W = 8 and W = 4, we observe that the decrease of blocking probability of our algorithm is more significant for TestNet topology. This due to reason that the TestNet topology has a higher network degree (3.73) than the NSFNet (3.0), so the ant-based agents of our algorithm have more chances to find a good routing solution in the TestNet network. In contrast, the PIBWA use only the pre-computed static routes, thus it is less flexible. The above results indicate clearly that our FT-ABR is much more adaptive than PIBWA algorithm.



(a) NSFNet



b) TestNet

Fig. 7. Performance of FT-ABR compared with PIBWA for 8 wavelengths

5. Conclusion

Inspired from the behavior of ant colony optimization, we proposed a new algorithm FT-ABR for dynamic lightpath protection in WDM optical networks. By using the proactive ant-based mobile agents to continuously explore the networks state and update the cycles in routing tables, our algorithm has a smaller computation time for dependable connections in comparison with the optimal method or others heuristics that compute the cycles after a connection request arrives. Moreover, the exploring mobile agents in our algorithm can discover feasible cycles between every node pairs based on the current network congested state, so our algorithm is more adaptive than other algorithms using static alternate routes, as confirmed by the simulation results upon two network topologies and the promising PIBWA method.

In the future, we are interested in some approaches for setting the parameters to improve the performance of this survivable routing algorithm. The comparison in terms of setup delay, control overhead and blocking probability with other heuristics will be investigated.

References

[1] R. Ramaswami et al., *Optical networks: a practical perspective,* Morgan Kaufmann Publishers Inc., San Francisco, CA, 2002

[2] H. Zang et al., “A Review of Routing and Wavelength Assignment Approaches for Wavelength-Routed Optical WDM Networks,” in *Optical Networks Magazine,* Vol. 1 pp. 47-63, Jan. 2000.

[3] R. Garey Michael and David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP Completeness,* W. H. Freeman and Company, San Francisco, CA, 1979.

[4] H. Harai et al., “Performance of Alternate Routing Methods in All--Optical Switching Networks,” in *Proc. IEEE INFOCOM,* p.516, April, 1997

[5] R. Ramamurthy and B. Mukherjee,, “Fixed Alternate Routing and Wavelength Conversion in Wavelength-Routed Optical Networks, ”, *IEEE/ACM Trans. on Networking,* pp. 351-367, June 2002

[6] H.T. Mokhtar and M. Azizoglu, “Adaptive wavelength routing in all-optical networks,” *IEEE/ACM Trans. Networking,* vol. 6, pp. 197-206, April 1998.

[7] Pin-Han Ho, “State-of-the-Art Progresses in Developing Survivable Routing Strategies in the Optical Internet”, *IEEE Communications Surveys and Tutorials,* Vol. 6, No. 4, 2004.

[8] S. Ramamurthy and B. Mukherjee, “Survivable WDM mesh networks, part I – protection,” in *Proc. IEEE INFOCOM,* vol. 2, pp. 744-751, March 1999.

[9] S. Ramamurthy and B. Mukherjee, “Survivable WDM mesh networks, part II – restoration,” in *Proc. IEEE ICC,* vol. 3, pp. 2023-2030, June 1999.

[10] G. Mohan et al., “Efficient Algorithms for Routing Dependable Connections in WDM Optical Networks,” *IEEE/ACM Trans. Networking,* vol. 9, no. 5, pp. 533-566, 2001.

[11] S. Yuan and J. P. Jue, “Dynamic Path Protection in WDM Mesh Networks under Wavelength Continuity Constraint,” in *Proc. IEEE Globecom,* vol. 3, pp. 2019-2023, Dallas, TX, November 2004.

[12] Pin-Han Ho et al., "On Optimal Diverse Routing for Shared Protection in Mesh WDM Networks", *IEEE Trans. Reliability,* Vol. 53, No. 6, pp. 216 - 225, June 2004.

[13] M. Dorigo and V. Maniezzo. “Ant system: Optimisation by a colony of cooperating agents,” *IEEE Trans. Systems, Man, and Cybernetics-Part B,* Vol. 26, No. 1, pp. 29-41, 1996.

[14] S.H. Ngo et al., “Adaptive routing and wavelength assignment using ant-based algorithm,” in *Proc. 12th IEEE ICON,* vol. 2, pp 482-486, Singapore, Nov. 2004.

[15] M. Di Caro and M. Dorigo, “AntNet: Distributed stigmergetic control for communications networks,” *J. Artificial Intelligence Research.* pp. 317-365, Dec. 1998.

[16] R. Schoonderwoerd et al., “Ant-like agents for load balancing in telecommunications networks,” in *Proc. 1st Inter. Conf. Autonomous Agents,* pp. 209-216. ACM Press, 1997.

[17] N. Minar et al., “Cooperating Mobile Agents for Dynamic Network Routing,” in *Software Agents for Future Communication Systems,* pp.287-304, Springer, 1999

[18] R.M. Garlick and R.S. Barr, “Dynamic wavelength routing in WDM networks via Ant Colony Optimization,” in *Ant Algorithms,* Springer, pp. 250-255, Sept. 2002

[19] The VINT project, *The Network Simulator ns-2,* http://www.isi.edu/nsnam/ns/index.html, 2000