Performance Study of Dynamic Provisioning and Restoration Algorithm - (DCOPA) for Optical Networks with promising QoS

R. Deepalakshmi and Dr. S. Rajaram
1Assistant Professor, Department of Computer Science and Engineering, Velammal College of Engineering and Technology, Madurai, Tamil Nadu, India.
2Associate Professor, Department of Electronics and Communication Engineering, Thiagarajar College of Engineering, Madurai, Tamil Nadu, India.

ABSTRACT

Background: Multimedia applications present new challenges to the current networking technology. One of them is Quality of Service (QoS) requirement. Due to the increasing dynamics of traffic introduced by multimedia applications, dynamic and flexible QoS (Quality of Service) control is needed to ensure both QoS satisfaction and resource efficiency. Because integrated service networks are designed to support a wide range of traffic classes, which have different service requirements, developing metrics to evaluate them is complex. Objective: To minimize the total bandwidth reserved on the backup edges coupled with the need to minimize the global use of network resources, imply that the cost of both the primary path and the restoration topology should be a major consideration of the routing process. Results: The performance analysis of DCOPA and BFS are compared based on various QoS parameters and the result showed that the rejection ratios achieved by DCOPA are 0% in all runs however, the rejection ratios by BFS tree routing range from 30% to 55%. According to the simulation results, that DCOPA can reduce the rejection ratio effectively and achieve fairly good bandwidth allocation efficiency. Conclusion: We have proposed DCOPA algorithm where network links are assumed to have infinite capacity, the proposed approximation provisioning algorithms (DCOPA) construct a topology, and this topology protects a portion of the primary QoS path. This approach guarantees to find a topology with optimal cost which satisfies the QoS constraints. In future extending the algorithm for using Alternative Cost Functions as the Performance Metric for Asymmetric case of bandwidth allocation in optical networks and to develop a new algorithm to integrated routing algorithm for this optical architecture, which is adaptive to traffic demands with dynamic load balancing.

INTRODUCTION

Optical networks are promising for serving as the backbone networks of the next-generation Internet due to their potential ability to meet the ever-increasing demands of high bandwidth. Internet applications can differ with respect to many characteristics, such as burst, high data transaction Bandwidth, multi-granularity of traffic flows, and high resilience. For example, data-intensive distributed applications may require moving different amounts of data from Kilobytes to Terabytes or even Petabytes among multiple sites. These applications may require different quality of service (QoS). Optical networking technologies offering enormous transmission bandwidth and advanced capabilities are expected to play an important role in creating an efficient infrastructure for supporting internet applications. Traditionally, optical network connection is established by grouping dedicated lines connecting several geographically dispersed sites (optical cross connects). As the number of endpoints is growing, connecting them with dedicated lines is becoming increasingly expensive Sudipta Sengupta et al., (2001).

As a result, optical network s has emerged as replacements in recent years. Its goal is to provide a service comparable to traditional networks. The two most important issues that must be addressed for optical network are data security and bandwidth guarantees Hongyue Zhu et al., (2005). The former is usually achieved by cryptographic methods, while the latter is achieved by reserving sufficient bandwidths on the links. In current optical network, the capacity of each light path is usually restricted to the granularities such as STM-N, N=1, 4, 16, 64. For high data transport bandwidth requires, single light path couldn't satisfied the demand. So we need to
setup multi-paths and split the traffic to these paths to implement data transport within the limited scope of time. It has been proved that a reasonable and efficient multi-path provisioning could increase the availability and decrease the rejection ratio of the whole network L. A. Cox et al, (2001). It is possible that the overall utilization efficiency of the network resource can be further improved through an appropriate traffic-distributing algorithm in multiple paths and an appropriate protected path provisioning S. Ganguly (2006).

This raises the research questions of how to decide the numbers of working paths and protection paths, how to design an adaptive algorithm to distribute the traffic onto the available multiple paths, and how to allocate the bandwidth resource to protection path to satisfy the QoS under a dynamic traffic requests. The most important optical network provisioning algorithms are: (1) BFS R. Ramamurthy et al (2004), D. Awduche et al,(2003) (2) Steiner tree Vladica Tintor et al, (2008) (3) tree routing Jun Gu et al (2004). For the approaches of selecting a data transmission path, path u,v, between each endpoint pair, (u,v), in an optical network and the allocated bandwidth on links of the paths in these algorithms, please refer to R. Ramamurthy et al, (2004), Kefei Wang et al, (2005) Jun Gu et al, (2004). The path pinning capacity provided by MPLS(multiprotocol label switching) technology can be used to direct the routing of a data transmission path between each endpoint pair in an optical network Vladica Tintor et al, (2008), Meeyoung Cha et al, (2009). Our approach can be implemented on a MPLS network as well. However, all these algorithms addressed the problem of TCP packets for IP networks, they do not consider important issues involved in optical technologies, and therefore do not meet the requirements and exigencies of optical network. L. A. Cox et al, (2001) investigated the problem of reliable multi-path provisioning for high-capacity backbone optical networks. It proposed and investigated the characteristics of effective multi-path bandwidth as a metric to provision a connection to multiple paths while satisfying its availability requirements. It developed two efficient heuristics for solving the multi-path provisioning problem. The results showed that the heuristics perform significantly better (decreasing blocking probability) than conventional single-path provisioning approaches for high bandwidth connections Ralf H’ulsermann et al, (2009).

Optical network provisioning algorithms can be implemented in two ways: (1) off-line provisioning and (2) on-line provisioning. In off-line provisioning, the NSP has a prior knowledge of all optical networks setup requests. In this setting, the optical network provisioning plan is optimized on some performance metrics (e.g., revenue, network link utilization and the amount of bandwidth reservation) by rejecting selected requests. In the on-line provisioning Tamás Kárász et al, (2005), when optical network setup request is received, it is processed based on the current state of the network.

In this paper, we focus on on-line optical network provisioning. To our knowledge until now, issues about the rejection ratios achieved by optical network provisioning algorithms have not been investigated. In this paper, we consider the problem of minimizing the rejection ratio of provisioning algorithms when (1) the residual bandwidths on links of the network backbone are finite, and (2) multiple cross connects need to be established on-line on the network backbone. Once the data transmission paths between each endpoint pair in an optical network are determined, the provisioning algorithm needs to explicitly allocate sufficient bandwidth on the links of these paths to meet the bandwidth requirement specified by customers Gringeri, S et al.,(2010). As the bandwidth allocation of cross connects is executed on-line, the previous allocation may affect the feasibility of the next optical network provisioning. One of the requisites of a good provisioning algorithm is that it should achieve a low rejection ratio. However, previous provisioning algorithms H. Yoshimi et al, (2003), O. Turkcu et al, (2007), S. Tibuleac et al, (2010), have been unable to meet this requirement. We therefore propose a new provisioning algorithm, the A Dynamic Cost Optimized Provisioning Algorithm (DCOPA) to address this issue. Our experimental simulations show that the DCOPA can reduce the rejection ratio effectively. In addition, it can also rapidly process multiple setup requests. Given a network graph G with n nodes and m edges, DCOPA spends only O(mn) time for a setup request.

The contributions of this paper are summarized as follows: (1) we show by concrete examples that all of the optical network provisioning algorithms mentioned previously is unable to achieve satisfactory rejection ratios. To address this issue, we propose a new optical network provisioning algorithm called DCOPA. (2) The theoretical upper bounds of the rejection ratios achieved by the BFS and DCOPA for the problem we consider are also derived in this paper.

The remainder of this paper is organized as follows. In Section 2, we define the On-line network-model Establishment Problem (ONMEP) where an NSP establishes connections online on a network backbone composed of links with finite residual bandwidths. In ONMEP, the performance metric for comparison with various provisioning algorithms is the rejection ratio. In Section 3, we exemplify the reasons why the provisioning algorithms proposed in [18-20] cannot achieve a satisfactory rejection ratio, and we present DCOPA. In Section 4, we derive the theoretical upper bounds of the rejection ratios for several provisioning algorithms under the ONMEP. In Section 5, we show four experimental simulations to compare the performance of DCOPA with other optical network provisioning algorithms. Finally, in Section 6, we give our conclusions and indicate the direction of our future work.
2. Problem Formulation and Modeling:

In this section, we formulate the problem considered in this paper. The network backbone managed by the NSP is modeled in subsection 2.1. The setup request describing the service requested by customers is modeled in subsection 2.2. Finally, the On-line network-model Establishment Problem (ONMEP) is described in subsection 2.3.

2.1. Network Backbone Modeling:

The network backbone is modeled by an undirected graph $G=(N, L)$, where $N$ and $L$ are the set of routers and the set of links, respectively. Let $n$ and $m$ denote the cardinality of $N$ and $L$, respectively. Let $B$ be the set of residual bandwidths of links on $L$, and the amount of residual bandwidth on link $l \in L$ is denoted by $B(l)$. A subset $AR = \{ar_1, ar_2, ..., ar_p\}$ of $N$ ($AR \subseteq N$) is the set of optical cross connects. Each endpoint $e_i$ of an optical cross connects gains access to service by connecting to a specific endpoint $a_{ri}$ in $AR$.

In other words, for each endpoint of an optical network, there is a corresponding optical cross connects in $AR$. The elliptic region in Figure 1 is an example of the network backbone $G$. The round regions (A to G) inside $G$ are optical cross connects in $N$. The solid lines between any two optical cross connects in $G$ are links in $L$. The number beside each link is the amount of residual bandwidth on it ($B(l)=5$ for all $l \in L$ in this figure). The optical cross connects set $AR = \{A, E, G\}$. The round regions (1, 2 and 3) outside $G$ are endpoints ($e_1$, $e_2$ and $e_3$, respectively, in our notation) of an optical network which gain access to service via optical cross connects in $AR$. The dotted lines labeled as path$i,j$ is the data transmission path for traffic between $e_i$ and $e_j$.

![Fig. 1: An example of Network Backbone G.](image)

2.2. Setup Request Modeling:

The demands for optical network service of customers are described by setup requests. In this paper, we consider that the bandwidth requirement of each endpoint $e_j$ is symmetric and use $b(e_j)$ to denote the bandwidth requirement of $e_j$. Let $Maxr$ denote the maximum bandwidth guarantee provided by the NSP, and $v_{ri}$ be the $i$th setup request from customer for the NSP to establish. Each $v_{ri}$ is represented by a $p$-tuple vector $(r_1, r_2, ..., r_p)$, where $p$ is the cardinality of the optical cross connects set $AR$. The number of nonzero elements in $v_{ri}$ represents the number of endpoints contained in the corresponding cross connects. The value of $j$th element, $r_j$, of $v_{ri}$ represents the bandwidth requirement of endpoint $e_j$.

2.3. On-line network-model Establishment Problem (ONMEP):

The ONMEP defined in this paper is similar to the work in Kefei Wang et al., (2005) Jun Gu et al., (2004) which mainly considers on-line establishment of bandwidth-guaranteed point-to-point tunnels. However, in the context of optical network provisioning, the basic unit of concern is a Optical cross connects consisting of numerous point-to-point tunnels, as opposed to one point-to-point tunnel, that makes the problem more challenging. In ONMEP, the NSP manages an network backbone $G$ (as described in subsection 2.1) on which
OXCs are established. We consider the situation where (a) setup requests arrive one-by-one independently, and (b) the NSP do not have a priori knowledge about future setup requests. This knowledge includes the number of future setup requests, the number of endpoints contained in each setup request, and the bandwidth requirement of each endpoint. In this situation, the NSP must process each path setup request in an on-line manner.

Upon receiving a setup request \( v_i \), the NSP triggers the provisioning algorithm to establish a corresponding connection. The provisioning algorithm performs this task by first choosing a data transmission path, path \( u,v \), between each endpoint pair, \( (u,v) \), and then allocating bandwidth on each link of the path. If there is not enough residual bandwidth on the link when the bandwidth is being allocated, \( v_i \) will be rejected. We use the rejection ratio as the performance metric to compare different provisioning algorithms. Note that the authors of H. Yoshimi et al., (2003), Tamás Kárász et al., (2005) Gringeri, S et al., (2010), Meeyoung Cha et al., (2009) also use the rejection ratio (of tunnel setup requests) as the performance metric to compare different on-line tunnel establishment algorithms. The rejection ratio is defined as:

\[
\text{Rejection ratio} = \frac{\text{Number of requests rejected}}{\text{Total number of request received}}
\]

The optimization goal of provisioning algorithms is to minimize the rejection ratio, which in turn will maximize the number of requests successfully established on the network backbone. In the ONMEP, we assume that the NSP uses a server-based strategy Mohamed E. M et al, (2003) for processing setup requests. In a server-based strategy, the provisioning algorithm is run on a single entity called request server (RS). The RS also keeps the complete link state topology database and is responsible for computing an explicit data transmission path for each endpoint pair of a cross connects. Then the paths can be setup using a signaling protocol such as RSVP or CR-LDP. For computing the explicit paths, the RS needs to know the current network topology and link residual bandwidth. We assume that a link state routing protocol for information acquisition exists.

3. Dynamic Cost Optimized Provisioning Algorithm (DCOPA):

To alleviate the drawbacks of (a) inefficiency on bandwidth allocation, and (b) disregarding the amount of residual bandwidth for links selection, we propose a new provisioning algorithm called the Dynamic Cost Optimized Provisioning Algorithm (DCOPA). The BFS and DCOPA are both tree-based (i.e., they establish a connection based on tree topology (optical tree)). While BFS has excellent bandwidth allocation efficiency, it does not consider maximizing the accommodation of on-line requests. On the contrary, DCOPA considers both bandwidth allocation efficiency and accommodation of on-line requests by achieving balance of link residual bandwidths.

The major difference between BFS and DCOPA is that the cost function they defined for optical tree selection. Let \( T \) be optical tree consisting of \( k \) links. The cost functions of BFS and DCOPA are defined as following:

\[
\text{Cost DCOPA}(T) = \sum_{x=1}^{k} RS(l_x) \quad \text{and \ Cost BFS}(T) = \sum_{x=1}^{k} RS(l_x)
\]

where \( RS(l_x) \) and \( B(l_x) \) represent the amount of bandwidth allocation needed and the amount of residual bandwidth on the \( x \)th link, \( l_x \), respectively. The cost function of DCOPA is derived by the cost function defined in the routing algorithms proposed in (12, 16) for route selection. When processing a request, DCOPA tries to find an optical tree that minimizes the cost function defined above. It is clear the additional cost for using a link \( l_x \) in building a optical tree is proportional to the value of \( RS(l_x) \) and is reciprocal to the value of \( B(l_x) \). Therefore, DCOPA tries to find optical tree with links of abundant residual bandwidths and low overall bandwidth allocation. As a result, DCOPA can satisfy both bandwidth allocation efficiency and balance of residual bandwidths. The pseudo code of DCOPA is described in Table.

Given a network graph \( G \) consisting of \( n \) nodes, to process a optical setup request \( v \), DCOPA iterates totally \( n \) times, once for each \( v \in N \). In each iteration, DCOPA first finds a candidate VPN tree \( PTV \) rooted at \( v \) for \( v \), and then computes the amount of bandwidth needed to be allocated to each link \( l_x \) of \( PTV \). Finally the cost value associated with \( PTV \) can be computed. After finding all \( PTV \) (\( v \in N \)), if there is not any \( PTV \) (\( v \in N \)) on which all links have enough residual bandwidth for allocation, DCOPA will reject \( v \). In the case of accepting \( v \), DCOPA will return the optical tree with the minimum cost value among all \( PTV \) (\( v \in N \)) for \( v \), which is denoted by \( OTMC \). In addition, DCOPA then allocates bandwidth to each link \( l_x \) of \( OTMC \) by performing \( B(l_x) := B(l_x) - RS(l_x) \).
Table 1: Pseudo code for DCOPA.

<table>
<thead>
<tr>
<th>Function: Dynamic Cost Optimized Provisioning Algorithm (DCOPA)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> A Network graph G=(N,L), Optical Cross Connects (OXCs) AR=(ar1,ar2,…,arp)(\subseteq)N, residual bandwidth constraints B on L, and a optical connection setup request vri =(r1,r2,…,rp).</td>
</tr>
<tr>
<td><strong>Output:</strong> A minimum cost optical tree OTMC for vri, on which all leaf nodes are Optical Cross Connects arj with rj&gt;0.</td>
</tr>
</tbody>
</table>

1. OTMC :=\(\varnothing\);
2. For each v\(\in\)N {
3.   Tv:= BFS_Tree(G,v);
4.   PTv:=Prune_Tree(Tv, vri);
5.   Compute_RS(PTv, vri);
6.   if(Cost(PTv)<Cost(OTMC) ) OTMC:= PTv;
8. }
9. if (Cost(OTMC) = \(\infty\)) { Reject(vri); Return \(\varnothing\); }
11. else { For each link lx\(\in\)OTMC {B(lx):= B(lx)-RS(lx);}
12.        Accept(vri); Return(OTMC);}
14. }

To find a candidate optical tree PTv rooted at v, DCOPA first finds a BFS tree (breadth first search tree (18)) Tv rooted at v (by calling Function BFS_Tree). Tv contains all nodes in G and in addition, Tv may contain nodes that are not VPN access routers used in vri as leaf nodes. Therefore, DCOPA prunes Tv and obtains a candidate optical tree PTv, on which all leave nodes are optical cross connects used in vri (by calling Function Prune_Tree).

DCOPA computes the amount of bandwidth needed for each link lx of a optical tree T according to the bandwidth requirement information in vri (by calling Function Compute_RS in Table 2). To compute the value of RS(lx) (Ix\(\in\)T), we first remove lx from T which partitions the optical tree into two subtrees Tx a and Tx b. Let BR_ Tx a and BR_ Tx b denote the accumulated bandwidth requirement of the VPN access routers (endpoints) on Tx a and Tx b, respectively. Then RS(lx) is determined by the minimum value of BR_ Tx a and BR_ Tx b. For more details about computing the RS(lx) value for each lx on a optical tree, please refer to Assi, C *et al.*, (2001).

Given a optical tree T, in a normal case, the function Cost of DCOPA returns the cost value computed by the cost function defined previously. However, where T is null (\(\varnothing\)), or there are links on T that do not have enough bandwidth for allocation, the function Cost will return \(\infty\). The time complexity of each iteration in DCOPA is O(m), which is determined by the function BFS_Tree. To process a request, a total of n iterations are required. So, It is clear that the time complexity of DCOPA for processing a request is O(mn).

Table 2: Pseudo code for Compute_RS.

<table>
<thead>
<tr>
<th>Function: Compute_RS(T, vri)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let lx be the xth link on T.</td>
</tr>
<tr>
<td>Let RS(lx) be the amount of bandwidth allocation needed on lx with respect to the bandwidth requirement specified in vri.</td>
</tr>
</tbody>
</table>
| For (each lx in T) {
2.      Initialize two variable BR_ Tx a, BR_ Tx b to value 0;
4.      For (each element rj\(\neq\)0 (1\(\leq\)j\(\leq\)p) of vri) {
5.          if (arj \(\in\) Tx ) then add rj to BR_ Tx a
7.          else add rj to BR_ Tx b
8.      }
9.      RS(lx):=min(BR_ Tx a, BR_ Tx b );
10.  }

We now consider Scenario disregarding the amount of residual bandwidths on links in BSF routing algorithm results in a higher rejection ratio. Initially, B(l)=5 for all l in L. The sketch of G, after accepting vr1, is shown in Figure 2. The number beside each link in G is its residual bandwidth after accepting vr1. The dotted lines form the minimum cost optical tree OTMC that DCOPA will output for vr1. After accepting vr1, DCOPA then processes vr2. Each candidate optical tree PTv (v\(\in\)EN) for vr2 considered by DCOPA is shown in Figure 3. We can find four different types of candidate optical tree for vr2. Note that PTA, PTB, PTC and PTD are identical. The number beside each link of PTv (v\(\in\)EN) is the amount of bandwidth that needs to be allocated to it.
The cost value associated with each PTv (v∈N) is:

\[
\text{Cost}(PTA) = \text{Cost}(PTB) = \text{Cost}(PTC) = \text{Cost}(PTD) = \text{Cost}(PTE) = \text{Cost}(PTF) = \text{Cost}(PTG) = \infty
\]

\[
\frac{RS(i_{A,C})}{B(i_{A,C})} + \frac{RS(i_{B,C})}{B(i_{B,C})} + \frac{RS(i_{C,D})}{B(i_{C,D})} + \frac{RS(i_{B,E})}{B(i_{B,E})} + \frac{RS(i_{D,G})}{B(i_{D,G})} = \frac{3}{3} + \frac{3}{5} + \frac{3}{3} + \frac{3}{5} + 0 = 3.4
\]

It is clear that DCOPA will return PTA for vr2 (vr2 is accepted by DCOPA). Hence the rejection ratio achieved by DCOPA in Scenario 0%.

4. Comparision and Analysis:

We conducted a study to measure the performance of our DCOPA and BFS algorithms, and compared them with the approach to connect OXCs. The major findings of our study can be summarized as follows.

- The BFS generates optical trees with the smallest cost for a wide range of incoming and outgoing bandwidth ratios. It outperforms the DCOPA algorithms for medium-to-large bandwidth ratios.
- For low incoming and outgoing ratios, the BFS algorithms slightly outperform the DCOPA algorithm. In many cases, they construct optical trees that reserve half the bandwidth reserved by BFS.

4.1. Experimental Results:

(Designed and developed DCOPA Vs BFS Tree Routing)
We compared the provisioning cost (that is, the total bandwidth reserved on links of the optical tree) and the running times of the algorithms for the symmetric as well as the asymmetric bandwidth models. In the study, we examined the effect of varying the following two parameters on provisioning cost: 1) network size 2) number of OXCs. Most of the plots in the following subsections were generated by running each experiment five times (with different random networks) and using the average of the cost/execution times for the five repetitions as the final result.

Fig. 4: Effect of Number of OXCs in performance of Algorithm (Symmetric)

4.2. Simulation 1: (Network Size):

Figure 4 depicts the provisioning cost of the BFS and DCOPA algorithms as the number of network nodes is increased from 10 to 100. OXC is assigned equal incoming and outgoing bandwidths and the number of OXCs set to 10% of the network size. The DCOPA algorithm is provably optimal for the symmetric case. Further, unlike the BFS algorithm which is oblivious to the bandwidths of endpoints, the BFS algorithm does take into account the bandwidth requirements for OXCs. As a result, it outperforms by almost a factor of 2 for a wide range of node values.

Table 3: Parameter configuration of Simulation 1.

<table>
<thead>
<tr>
<th>Number of OXCs</th>
<th>DCOPA-cost</th>
<th>BFS-cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>2050</td>
<td>3000</td>
</tr>
<tr>
<td>30</td>
<td>2500</td>
<td>4400</td>
</tr>
<tr>
<td>50</td>
<td>4500</td>
<td>7000</td>
</tr>
<tr>
<td>75</td>
<td>5500</td>
<td>9000</td>
</tr>
<tr>
<td>100</td>
<td>6500</td>
<td>10000</td>
</tr>
</tbody>
</table>

4.3. Simulation 2: (Rejection Ratio):

We conduct 15 runs with various number of topology, in each of which, 100 requests are randomly generated. The rejection ratios achieved by the two provisioning algorithms are (see Figure 5). The x-axis represents the number of OXCs, and the y-axis represents the rejection ratio and average link utilization achieved by each provisioning algorithm in each run.

Fig. 5: Effect of Rejection Ratio (symmetric).

Table 4: Parameter configuration of Simulation 2.

<table>
<thead>
<tr>
<th>Number of OXCs</th>
<th>DCOPA-Rejection Ratio</th>
<th>BFS-Rejection Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>50</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>75</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

The table 2 shows that the rejection ratio achieved by DCOPA is much less than that achieved by BFS routing.
4.3 Simulation 3: Performance Comparison:

The parameter configuration of Simulation 3 is shown in Table 5. Due to extensive adaptation of the topology as the network backbone in the literature about traffic engineering, we also adopt it as G. The topology is composed of 15 optical cross connects and 28 links.

The OXCs labeled as ar1~ar7 are optical cross connects, the amount of residual bandwidth on the light links is 1500 units, and the amount of residual bandwidth on the dark links is 6000 units.

Table 5: Parameter configuration of Simulation 3.

<table>
<thead>
<tr>
<th>G</th>
<th>B(li)</th>
<th>p</th>
<th>Maxr</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topology</td>
<td>Light links=1500 units</td>
<td>7</td>
<td>75</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Dark links=6000 units</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We conducted 15 runs in Simulation 3, in which each run randomly generated 100 requests. The simulation results are shown in Figure 6. The x-axis represents the run number and the y-axis represents the rejection ratio achieved by each provisioning algorithm in each run. The result shows that the rejection ratio achieved by DCOPA is much less than that achieved by BFS tree routing. The rejection ratios achieved by DCOPA are 0% in all runs except in run 8 and run 10 (where they are only 2% and 1%, respectively). However, the rejection ratios by BFS tree routing range from 30% to 55%. According to the simulation results, we believe that DCOPA can reduce the rejection ratio effectively.

Fig. 6: Performance comparison.

4.4 Simulation 4: The Effect of Maxr:

The parameter configuration of Simulation 4 is shown in Table 6. In order to evaluate the performance of DCOPA on general G, we used Brite to randomly generate a connected graph G with 20 nodes and 40 links in each run. The value of Maxr varies from 40 to 120 with a step of 20. We conducted 8 runs for each value of Maxr, and took the average rejection ratio achieved in these 8 runs.

Table 6: Parameter configuration of Simulation 4.

<table>
<thead>
<tr>
<th>G</th>
<th>B(li)</th>
<th>p</th>
<th>Maxr</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Randomly generated by Brite with 20 nodes and 40 links</td>
<td>1500 units</td>
<td>6</td>
<td>40~120 step 20</td>
<td>100</td>
</tr>
</tbody>
</table>

Fig. 7: The Effect of Maxr.

The simulation results are shown in Figure 7. The x-axis represents the value of Maxr, and the y-axis represents the average rejection ratio achieved by the provisioning algorithms. As expected, the average rejection ratio increases as the value of Maxr increases in all two algorithms. The average rejection ratio achieved by DCOPA is much less than the other algorithms in almost all the Maxr values considered in this simulation (except for the light load case, when Maxr=40, the average rejection ratios is 0% in all the two
algorithms). The experimental results show that DCOPA can indeed achieve a lower rejection ratio on general $G$ compared to the other algorithms.

4.5 Simulation 5: The Bandwidth Allocation Efficiency:
The parameter configuration of Simulation 5 is shown in Table 7. This experiment investigates the bandwidth allocation efficiency achieved by DCOPA. Because BFS tree routing is certain to find a bandwidth-optimization optical tree for each request, we compare the average amount of bandwidth allocated for processing 100 requests in DCOPA with BFS tree routing. As the cost functions defined, we expect the behavior of DCOPA will be more similar to BFS tree routing as the residual bandwidth amount on links ($B(lx)$) is more abundant. We conducted 8 experiments for each amount of residual bandwidth on the links, and took the average on the allocated bandwidth in these 8 runs. For the comparison to be fair, only simulation runs that had no rejected requests were considered.

Table 7: Parameter configuration of Simulation 5.

<table>
<thead>
<tr>
<th>G</th>
<th>B(li)</th>
<th>p</th>
<th>Maxr</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Randomly generated by Brite with 20 nodes and 40 links</td>
<td>5000–10000</td>
<td>6</td>
<td>40–120 step 20</td>
<td>100</td>
</tr>
</tbody>
</table>

We define RSDCOPA and RSBFSTree routing as the average amount of bandwidth allocated for processing 100 requests in DCOPA and tree routing, respectively. We also define $\text{PercentExtra_BW} = \frac{\text{RSDCOPA} - \text{RSBFSTree routing}}{\text{RSBFSTree routing}}$.

Table 8: The average amount of allocated bandwidth for 100 requests.

<table>
<thead>
<tr>
<th>B(lx)</th>
<th>RSDCOPA</th>
<th>RSBFSTree routing</th>
<th>PercentExtra BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>5000</td>
<td>21704</td>
<td>21138.38</td>
<td>2.6758%</td>
</tr>
<tr>
<td>6000</td>
<td>21884.63</td>
<td>21858.38</td>
<td>1.4926%</td>
</tr>
<tr>
<td>7000</td>
<td>21957.25</td>
<td>21721.13</td>
<td>1.0871%</td>
</tr>
<tr>
<td>7500</td>
<td>20377.63</td>
<td>20294.5</td>
<td>0.4096%</td>
</tr>
<tr>
<td>10000</td>
<td>21927.63</td>
<td>21872.5</td>
<td>0.252%</td>
</tr>
</tbody>
</table>

The simulation results are shown in Table 8. As expected, DCOPA achieves better bandwidth efficiency when $B(lx)$ is more abundant. For all the $B(lx)$ values we consider in this experiment, the values of PercentExtra_BW are all below 3%. Therefore, DCOPA can achieve fairly good bandwidth allocation efficiency.

Conclusions:
In this research work, the designed novel algorithms for provisioning optical network DCOPA connected OXC using a tree structure and attempted to optimize the total bandwidth reserved on edges of the optical tree. The algorithm showed that even for the simple scenario in which network links are assumed to have infinite capacity, the general problem of computing the optimal tree is NP-hard. However, for the special case when the incoming and outgoing bandwidths for each OXC are equal, DCOPA proposed a breadth-first search algorithm for computing the optimal tree. According the simulation results DCOPA can indeed reduce the rejection ratio effectively.

In future extending the algorithm for using Alternative Cost Functions as the Performance Metric for Asymmetric case of bandwidth allocation in optical networks and to develop a new algorithm to integrated routing algorithm for this optical architecture, which is adaptive to traffic demands with dynamic load balancing.

REFERENCES


