A Path Computation algorithm in MPLS Traffic Engineering for reducing blocking probability

Shahram Mohrehkesh, Saleh Yousefi, Mahmoud Fathy

Faculty of Computer Engineering, Iran University of Science and Technology (IUST)
Farjam St., Tehran, Iran
mohrhkesh@itrc.ac.ir

Faculty of Computer Engineering, Iran University of Science and Technology (IUST)
Farjam St., Tehran, Iran
syousefi@iust.ac.ir

Faculty of Computer Engineering, Iran University of Science and Technology (IUST)
Farjam St., Tehran, Iran
mahfathy@iust.ac.ir

Abstract: In this paper a new path computation algorithm in Traffic Engineering in MPLS networks for reducing blocking probability of requests is proposed which is an improvement of TE-DB algorithm. The TE-DB is one of best path computation algorithm which is proposed in literatures. The Algorithm deploys multiple path routing and distributes the load more balanced in network. In addition, it reduces the blocking probability of requests. Reducing probability of blocking and distributing load are two important objectives in Traffic Engineering which TEDM improves these two parameters. Four heuristic algorithms are proposed for distributing load among multiple paths. The algorithms are simulated by MATLAB. The results show that TEDBM reduces the blocking probability about 50% in average and distributes the load more balanced than the TE-DB algorithm. Hence, the network utilization is also increased.

Keywords: Traffic Engineering, QoS, Bandwidth, Delay.

1. INTRODUCTION

The exponential growth of the Internet has led to the increasing importance of network management and control functions. It is evident today that adding more bandwidth to networks is not the solution to all congestion problems. At the same time, more and more providers are showing interest in making revenues from offering differentiation of services in their networks. This requirement has increased the importance of gaining control over networks via automated Traffic Engineering (TE). The most common TE objectives are to reduce congestion, improve network utilization, satisfy diversified requirements and thus lead to an increase of revenue. One of the most important applications of MPLS networks will be in TE [1]. MPLS Traffic Engineering (MPLS-TE) enables an MPLS backbone to expand upon TE capabilities by routing flows across the network based on the resources the flow requires and those currently available. Since the essence of TE is mapping traffic flows onto a physical topology, it implies that at the heart of MPLS-TE resides the problem of path computation.
In other way, current applications need quality of service. In recent years QoS based path computation has studied widely [2-5]. Quality of service requirements are expressed in the form of service level agreements (SLA). QoS path computation then translates to finding paths through the network such that these requirements are satisfied. While previous path computation algorithms were mainly hop-by-hop distributed algorithms optimizing static metrics (e.g. link cost, length, etc.), the new generation of path computation algorithms is moving towards source routing schemes that also take into account dynamic metrics. A few examples of these dynamic metrics are available bandwidth on links, link reliability, link load, jitter, packet loss, etc [6-10]. But the most important objectives are bandwidth and delay. The next problem after satisfying QoS objectives is reducing congestion due to distribute and balance load in network. The goal is optimum load balancing and reducing congestion. Traffic Engineering is the key of the problem. TE and providing quality of service guarantees are closely related to each other. In fact both of them lead to minimize network costs and more revenue. There are many algorithms for providing multiple QoS objectives [9-10], and many algorithms are proposed for Traffic Engineering objectives [11-13]. In [13] a path computation algorithm is proposed, TE-DB (Traffic Engineering delay bandwidth) which satisfies two most common QoS objectives (bandwidth and delay). In addition, it considers three Traffic Engineering objects (distributing load, reducing probability of blocking, and reducing cost). Its result shows better performance in comparison to similar path computation algorithm. So, in this paper it is investigated for improvement. On the other hand, it considers five objectives simultaneously.

In this paper an improvement in TE-DB algorithm is done based on multiple paths routing technique and distributing traffic load among the multiple paths. This technique will reduce requests probability of blocking and increase network utilization. So, two important Traffic Engineering objectives are improved. In the proposed algorithm, which is called TEDBM, when there is not enough bandwidth in a single path to carry the traffic, multiple paths are used and the traffic is distributed among them. Multiple paths mechanism is simply supported by MPLS network signaling protocols for Traffic Engineering like RSVP-TE [14]. Four heuristic algorithms are proposed and evaluated for distributing load among multiple paths.

In the next section, the TE-DB algorithm is introduced and then the proposed algorithm, TEDBM, is explained. In section 3 the algorithm simulation results are shown and compared with TE-DB. At last in section 4 the conclusion is presented.

2. TEDBM ALGORITHM

In this section, at first the TE-DB algorithm is introduced, then the new algorithm, TEDBM, is explained. The TEDBM improves the TE-DB by deploying multiple paths. In the new algorithm, TEDBM, blocking probability of requests is reduced and load distribution is improved.

2.1. TE-DB

In [13] bandwidth and delay are considered as QoS objectives. In addition, Traffic Engineering objectives are reducing the blocking probability, minimizing network cost, and distributing load. The TE-DB uses the multiple constraint based routing. Multiple constraint based routing algorithms try to find a path based on multiple constraints. For example, limited bandwidth, limited
delay, limited cost, and using special links [6], [8]. Next, mapping of objectives to constraint in TE-DB is explained.

For satisfying bandwidth and delay, the most values of these two parameters are considered as constraint (maximum bandwidth requirement and maximum acceptable delay). For reducing blocking probability of requests, the “maximum flow minimum cut” is used [15]. In [4], it is intuitively shown that this algorithm route the flow of a request in a path which reduces the probability of blocking of next requests. This algorithm finds a path for a flow between a source-destination, so minimizes reduction in maximum flow between others source-destination flows. A max-flow weight reduction related to the bandwidth of request is assigned to every link heuristically. Then “path flow reduction” is computed for every path which is sum of max-flow reduction weight for links in the path and F value as constraint is defined. The F value is derived from the max-flow reduction weight of the least cost path. The path cost is another function. A link cost is assigned to every link. Then, the path cost is calculated as the link cost of all links of the path. The link cost can be defined by administrator. Also the C value as constraint of cost function is defined. The C value can be defined by administrator, or changed dynamically, or defined on every function.

No constraint is defined on distributing load object. However, a weight is assigned to every link related to the bandwidth allocated to all capacity of link. A threshold which called U is defined which can be set up dynamically or statically. If the proportion of the allocated bandwidth to capacity of link is under U threshold, then link weight will be zero; otherwise, the proportion will be assigned.

The TE-DB get a request from source A to destination B with bandwidth BW and D delay requirements. At first step, all links which their residual bandwidth are less than BW, are cut. Then a candidate set of paths should be founded which satisfies D, F, C constraints. Ref. [16] describes the A* prune algorithm for finding k paths subject to multiple constraints. Number of paths, k, is a variable and defined between 4 -6 usually. More value for k will increase the complexity of the algorithm. After finding k paths, for every path the link weight function is computed and the path with minimum weight among paths is selected. In an n-node network with m links and link threshold capacities in range [1..U], the TE-DB algorithm time complexity is $O(\min(n^{2/3}, m^{1/2})m \log(n + m)\log(U + k\log(kn)))$ [13]. In searching k paths step, it may find no path. So, the new proposed algorithm tries to load traffic among multiple paths. Before explaining the new algorithm, TEDBM, the multiple paths concept is defined

2.2. Multiple paths

The On of the capabilities of Traffic Engineering is establishing multiple paths between a source and destination [14]. Multiple paths are used in many areas. Most important applications are included:

1- Backup path: The backup path is used for a primary path. When a failure occurs in the primary path, the traffic is rerouted from the primary path to the backup path. Some applications in this area are explained in [17-18].

2- Distributing load: Multiple paths are applied in distributing load among them and better utilizing network resources. Some algorithms in this area are proposed in [19-20].
3- Reducing probability of request: After distributing load in the network, the probability of inputting traffic into the network will increase and result in reducing probability of blocking.

As mentioned above, many algorithms are proposed for distributing load in networks. Furthermore, the effect of reducing probability of blocking is explained. In these algorithms multiple paths are established and the traffic is distributed among them based on different algorithms. But, the multiple paths routing algorithm has the disadvantage that more number of resources of networks are used. Thus, it increases the cost of network. The cost of network is an important parameter for ISPs which affects the revenue. So, in the TEDBM algorithm, the goal is reducing probability of blocking and distributing load among multiple paths. But, the multiple paths routing is used only when the traffic request can not be routed through a single path. The result is reducing the number of resources which are used and reduction in cost of network. Indeed, two types of paths computing are combined in TEDBM effectively. In the next subsection the TEDBM algorithm is explained.

2.3. TEDBM

After running of TE-DB, if no path is found then, the TEDBM algorithm is run. In this algorithm no link is deleted and all of the links with any residual capacity are included. Then, k paths are found on the network and traffic will be distributed among them. For distributing traffic among k paths, a function is calculated for every path. This function which is called minBandwidth is equal to minimum available bandwidth of the path. Then, if sum of minBandwidth of all paths were more than requested bandwidth BW, m paths of the k paths will be selected and traffic is distributed among them. For distributing load among the m paths four heuristic methods are proposed and evaluated:

First fits: In this method the paths are selected based on the order which have been found in searching algorithm for the k paths. Then, the traffic is assigned to them until the sum of the minBandwidth of the m paths fills the BW bandwidth requirement.

Largest fits: In this method the paths are sorted in descending order by their minBandwidth values and then selected. Then, the traffic is assigned to them until the sum of the minBandwidth of the m paths fills the BW bandwidth requirement.

Smallest fits: In this method the paths are sorted in ascending order by their minBandwidth values and then selected. Then, the traffic is assigned to them until the sum of the minBandwidth of the m paths fills the BW bandwidth requirement.

Lightest fits: In this method the paths are sorted in ascending order by their weight function values and then selected. Then, the traffic is assigned to them until the sum of the minBandwidth of the m paths fills the BW bandwidth requirement.

Here an Example is stated. Consider five paths (A-E) have founded with minBandwidth 7, 3, 8, 11, and 14 respectively. In addition, the C(minBandwidth 8) and E(minBandwidth 11) paths have the lightest load among the five paths and the bandwidth request is 20. In this situation, the First selects 7, 3, 8, 11; The Largest selects 14 and 18; The Smallest selects 3, 7, 8, 11; The Lightest Selects 8 and 11. Time complexity of the TEDBM is equal to time complexity of TE-DB plus selecting m paths from k paths. In worst case, selecting m paths from k paths has the complexity of sorting k number which is equal to O(klogk). As it is stated before, TE-DB has the O(min(n 2/3 , m ½ )m log(n 2 /m)log U+ knlog(kn)) complexity. So, the TEDBM complexity is O(min(n 2/3 , m ½ )m log(n 2 /m)log U+ knlog(kn))+O(klogk) which is equal to O(min(n 2/3 , m ½ )m log(n 2
/m)log U^+ knlog(kn)). So, the complexity of TEDBM is the same as TE-DB. In the next section four proposed heuristic methods for TEDBM and TE-DB algorithms are simulated and evaluated.

3. SIMULATION

In this section, the TE-B and TEDBM are simulated and their performance is evaluated with each other. The MATLAB 6.5 is used for simulation [21]. Performance comparison is done on the basis of stochastic simulations, where connection requests randomly arrive in the network. Each algorithm attempts to route these connections and the network performance (in terms of metrics discussed in Section 3.1) is measured.

3.1. Simulation model

The network topology for simulation is shown in fig. 1. This network is a real topology of an ISP network which is also used in [13].

![Figure 1. ISP network topology](image)

The dark shaded nodes represent the gateway nodes, which serve as entry and exit points for the network traffic. The remaining nodes are the backbone nodes, which carry transit traffic only. Link capacities are either moderate (OC3 links) or high (OC12 links). The link delays assigned randomly and are of the order of a few milliseconds. In reality, link delays and weights vary largely from one provider’s network to another. However, there exists a range of acceptable values for these parameters that are used in most studies (e.g., link delays for OC3 links are usually in the range of 10–100 ms). In the absence of real values, we simulate the link parameters by generating them randomly from the acceptable range of values.

The parameter k determines the size of the candidate set of paths. The more candidate paths we have, better quality solutions we can attain. However, the complexity of the algorithm also increases with k. Ref. [9] suggests 4–6 as a possible range of k values.
Requests are generated randomly. It is noticeable that for putting the network on high load condition and getting better results, the requests have infinite life time. It means that after inputting a request to the network and allocating the resource, the resource is not released until the end of simulation. The bandwidth requirement of requests varies uniformly between 1 and 200 kilo bits per sec. Requests bandwidth requirement are usually Normal Distribution Function [22], so the experiment are done for various Normal Distribution Function with different mean and variances. The delay requirement varies uniformly between 50 and 100 ms. The evaluation of algorithms is done from a TE perspective with the purpose of analyzing how these algorithms compared with each other in terms of blocking probability of requests and distributing network load. We define the following parameters:

**Blocking probability of requests**: Number of requests which a path did not find for them to all request in the system, is the blocking probability of requests.

**Average utilization**: For every request, link with maximum utilization is defined as the request utilization of links. Then, the average utilization is equal to the average of link utilization of all requests.

### 3.1. Results

Comparison of results is done between TE-DB and TEDBM with the four proposed methods (first, smallest, largest, lightest fits). In Fig. 2 the blocking probability of requests is shown. As it is shown, the performance of all methods of TEDBM algorithm is better than TE-DB algorithm in term of **Blocking probability of requests**. The TEDBM algorithm reduces the probability of blocking by about 50% in average. For example, when there are 18000 request in the system, the probability of blocking of TE-DB & TEDBM(first fits) are 0.5 and 0.25 respectively.

In addition, all of TEDBM methods distribute traffic in network more balanced than TE-DB algorithm. Among the four proposed methods for TEDBM algorithm, the smallest performance is better than the others. It seems that the reason is better usage of bandwidth which first uses the small available bandwidth and saves larger available bandwidth for next requests. Average utilization is shown in fig. 3. Inputting more requests in the network, **Average utilization** increases too. In all of four methods of TEDBM algorithms, the **Average utilization** is better than TE-DB
algorithm. It is noticeable that because many requests are blocked so they have zero utilization which decreases the Average utilization.

![Figure 3. Average utilization](image)

4. CONCLUSION

In this paper a path computation algorithm, called TEDBM, is proposed which improves an algorithm called TE-DB [13]. The TE-DB is selected because it has the best performance than other path computation algorithms. The TEDBM algorithm reduces the blocking probability of requests as a goal of Traffic Engineering in an MPLS network. In TEDBM when the traffic can not be routed through a single path, it is rerouted and distributed among multiple paths by using multiple parallel paths routing. Four heuristic methods are proposed for distributing traffic among paths in which one of them called the smallest fits gives the best performance. Deploying multiple paths, the traffic is distributed more balanced than single path routing. The advantage of the algorithm is that it uses the multiple paths only when the traffic can not be routed by a single path. When network resources has cost and minimizing cost is an objective, this method has more benefit. In addition, the utilization of network resources in TEDBM algorithm is more than TE-DB algorithm. However, TEDBM algorithm can be used for path computation for a requested bandwidth, maximum delay, reducing blocking probability of request, minimizing cost and distributing load.

REFERENCES


