Constraint-based Path selection algorithms for Minimizing Blockage in Multi Domain Networks

Fatemeh Banaie, Mohammad Hossein Yaghmaee, Nazbanoo Farzaneh
Department of Engineering
Ferdowsi University of Mashhad
Mashhad, Iran
Email: Banaie.F@stu-mail.um.ac.ir, yaghmaee@ieee.org, farzaneh@stu-mail.um.ac.ir,

***Abstract*— During recent decades new routing paradigms based on policies and quality of service provisioning have been proposed. The aim of constraint-based path selection algorithms is to satisfy a set of quality of service constraints. This can help to reduce costs and balance network load. Path computation algorithms pose new challenges in larger inter domain networks. In these cases the process of path computation is more prone to block due to the long response time of the requests. In view of this, we propose and compare an algorithm to increase the number of successful requests while minimizing the blockage in network. The proposed algorithm also considers several QoS metrics in path computation that can help to have uniform traffic load in network. In this paper we focus on restrictions of the shortest path algorithms and propose a multi constrained path selection algorithm based on PCE.**

Keywords-constraint-based; path computation; inter-domain; PCE; uniform traffic model.

I. Introduction

The evolution of future broadband services enables higher volumes of information and offers better quality of service. So quality of service provisioning, which can be obtained by the Traffic Engineering, is an important requirement to information communication networks. In this context, current transport technologies such as MPLS/GMPLS are the best protocols for the Traffic Engineering. One of the important aspects of the Traffic Engineering is computation and establishment of the reliable communication end to end paths with QoS constraints across multiple routing domains. IETF has proposed path computation element (PCE) [1] to support the inter-domain path computation. In these complex cases, routing decisions in each domain is delivered to the PCEs. The main advantages of PCE-architecture are the simplification of control plane and enabling CPU and memory intensive path computation. Doing this may not be possible at a single network element [2]. In these scenarios, there is a significant time interval between the path computation request and the reception of a reply to this request. This long response time can cause blockage at the actual reservation/deployment time due to the resource fluctuation. It means resources that were available during the path computation may not be available anymore and block the deployment of the TE-LSP[3].

In this paper, we propose a method to increase the network utilization while minimizing blockage in the network. The proposed method considers several metrics in computing the end to end paths. It also consists of three mechanism: a timer setup phase with the path length and network conditions, pruning the traffic to decrease the network load, and multiple metric path cost. Provided results show significant performance gain in the terms of network utilization and reducing network load.

The rest of the paper is organized as follows: Section II, presents related work on the topic. In section III, we describe various functions required in an inter domain Path selection algorithm. In section IV, the proposed method is described in detail. Section V includes the simulation results and discusses the obtained results. Section VI concludes the paper with final remarks. Finally section VII presents the future research directories.

II. Related Work

Aiming to support the inter-domain path computation, IETF has proposed PCE architecture for the context of MPLS and GMPLS traffic engineering. The PCE architecture has been recently introduced to deal with constraint-based path computation challenges in multi domain network[4]. A PCE is an independent entity or can be a part of a router. It has visibility to all network resources through traffic engineering database (TED) [1]. The PCEP protocol [5] relies on TCP and allows clients to requests paths from PCEs and also allow for communication between PCEs. PCE-based routing architecture can be divided into two major group: peer-to-peer and hierarchical [6]. In a peer-to-peer model, neighboring PCEs create peer-to-peer relations and PCEs are explored sequentially to determine the availability of the path [7]. Okumus et al. in [8] propose a hierarchical PCE approach. In this approach a single PCE is used in each domain and a central PCE in global. The central PCE aggregates information from domains and calculates the optimal end-to-end path.

Two general approaches have been proposed for the inter domain path computation with PCE architecture. In the first approach, referred as per domain path computation [9], each domain performs the path computation for the local portion of end-to-end LSP through the pre-determined domain sequence. The second approach is Backward Recursive PCE-based Computation (BRPC) [10] that finds the optimal end-to-end path through the pre-configured domain sequences. The PCE in the destination domain creates a virtual shortest path tree (VSPT) and send it back until it reaches to the PCE in the source domain.

III. MULTI-DOMAIN PATH COMPUTATION MODEL

We define G/MPLS path computation along a path using the *RSVP-TE PATH* messages. As LSP is deployed, it compares the required resources with the available resources. A

978-1-4673-2073-3/12/$31.00 ©2012 IEEE

PathErr message is returned when the available resources is not sufficient. In inter-domain cases, path computation time is longer. As a result it increases the probability of resource unavailability and this can block the deployment of TE-LSP. To address this issue [3] has proposed to pre-reserve resources before the actual reservation/ deployment time. In this method, while computing the path, resources are pre-reserved. In other words, they pre-reserve the resources at reply time and set a timer for it. The time of pre-reservation is very important and has a significant impact in network performance. So it should not be very long to waste the resources and also should not be expire before the actual deployment/reservation. Therefore, this technique guarantees the availability of the resources in deployment time and reduces the blocking probability [3].

A. The Reservation Model

Three basic messages are introduced in [3]:

- [Path/Qos request] that is composed of source-destination path and required Qos.
- [Path/Qos reply] that is sent in response to the request and pre-reservation is done in this step.
- [Path/Qos request confirm] is for the actual reservation.

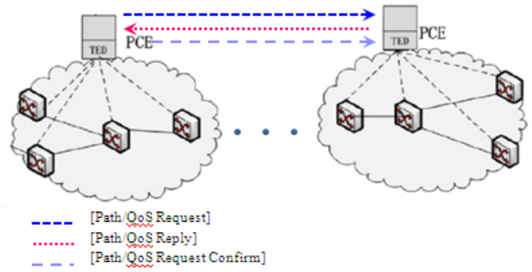

Figure 1. The reservation mechanism

B. The blocking probability reduction model

As mentioned previously, in inter-domain path computation scenarios, we have a higher rate of blocking probability due to the longer response time of the PCEs. The response time of a PCE is defined as the time when a PCE sends a request until it receives the reply. This time can be affected by several factors such as the network complexity and PCE limitations e.g. CPU limitation and workload of PCE [3]. Other factors may also influence the blocking probability such as:

- Long resource pre-reservation that can occur because of the improper timer setting.
- Flooding can cause to reserve a large number of paths. This may lead to the waste of resources in network.

First, source sends a request to the destination. After pruning the non-feasible paths according to the available resources, one or a set of feasible paths will be returned to the source. So, it can select the best one through the available paths and signal the LSP, Fig.1.

IV. MULTIPLE METRICS PATH COMPUTATION MECHANISM

In the proposed model we assume that $N=\{1,2,\ldots,n\}$ PCE nodes are in the network. There are a set of links $L=\{1,\ldots,l\}$ that each one is between PCE i and j.

Resource reservation protocols ensure that the path from a source node to a destination node is reserved before the connection is used. Simple reservation algorithms that determine paths based on the minimum number of hops or delay to a specified destination are no longer sufficient. The need to support diverse traffic types and applications with quality demands imposes new requirements on reservation in the network. In general, constrain based path selection is known to be NP-complete and many heuristic algorithms have been proposed for it.

In this paper, a method for inter-domain path computation is presented. The proposed model Make a reservation based on the request parameters which considers two QoS metrics in path computation, *hop count* and *delay*. When we only use hop count criteria in computing the path, shortest paths will be selected. In fact, in this case we only consider the propagation delay and omit the queuing delay. But queuing delay imposes large time delay that is not negligible. This causes to choose the shortest paths in path computation process and in result this makes some links congested. So some domains in the network that are connected to larger number of domains experience higher congestion. This may have negative impact on not only the domains performance, but also the whole network performance [13].

The proposed model consists of three main parts:

A. Reservation Timer Configuration
B. Resource reservation Reduction mechanism
C. Multiple metrics path cost

Due to the broadcast nature of the resource reservation techniques, network resources can be unnecessarily used in this. The first two parts of the proposed protocol are reducing the reservation messages which are flooded in the network. Thus the blocking probability can be reduced remarkably.

In the *Reservation Timer Configuration* part , our goal is to determine the appropriate time duration for pre-reserving the resources. Pre-reservation timer has an important role in the network performance. Because if we consider a long time duration for pre-reservation timer, the network resources remain reserved for a long time and this wastes the resources. As a result the other requests may be rejected due to lack of resources. On the other hand, it should not be expire before the reservation/deployment time. If so, it may block the deployment of the TE-LSP; because resources may be reserved by another request. So, adjusting the pre-reservation timer has a notable impact on network performance.

To find the PCE's sequence, we flood the request in network. Although this imposes a large overhead in terms of network load and resource reservation, but flooding yields a drastically reduced network blocking probability compared to a blind per-domain path computation but introduces significant network control overhead and path computation complexity. When we flood a request, it passes through different routes to reach the destination. So this makes a great number of massages broadcast in the network and the resources are reserved in all paths. While by pruning the non-promising paths we can reduce the traffic load and increase the network utilization.

A. Reservation Timer Configuration

To ensure availability of resources [3] proposes to pre-reserve the resources before the actual LSP deployment. This method requires a timer that has a significant role in network performance. In [3], they set the pre-reservation timers for a fixed time based on experiment. But as we know the network conditions are not fixed and will change constantly. Therefore, considering a constant value for the timer is not appropriate. Hence, our aim is to determine an appropriate time for the timer with considering the network conditions.

After sending the message for each request for the first time, each PCE will schedule a reservation timer based on the network topology. We need this value for initializing the destination timer. This value can be obtained by experiment. So, we change the obtained value according to the network conditions and path length. Since the resources must be reserved from receiving a reply until the time when confirm message is received; So PCEs that are closer to the source, require less time duration than the PCEs that are close to the destination. Therefore, we can adapt the time with the path length. In the proposed method, each node sets its timer according to its distance from destination and the queuing delay of the intermediate nodes. Therefore the value of the timer is not fixed in all nodes. Thus, each node changes the timer value based on two following parameters:

- D_q: PCEs' average queuing delay
- D_l: Communication delay

End-to-end delay consists of queuing delay and propagation time, so they will impact the timer value. Since network conditions are changing, we must consider these parameters in setting the timer value. The amount of this value increases according to the increasing of the end-to-end delay.

Queuing delay is a function of the number of packets in the queue. Packet arrival and packet departure changes queue length and as a result changes queuing delay. Thus a node monitors its queuing delay periodically. Due to pay attention to the past history of the node status, $D_p{}^k$ can be computed by using a Exponential Weighted Moving Average (EWMA). When a data packet arrives, each PCE monitors its current queuing delay (D_{cur}^k) and calculates an average value (D_q^k) using EWMA formula as follows:

$$D_q^k = (1-\omega) \times D_q^k + \omega \times D_{cur}^k \qquad (3)$$

$$0 \le \omega \le 1$$

Where ω is the moving average coefficient with older values. As a result path end to end delay can be calculated as follows:

$$path\ delay = D_e = \sum_{j=0}^{P} (D_q^j + D_l^j) \qquad (4)$$

Where D_e is the total path delay, P is the number of nodes in the path from source to destination, D_q^j is the average queuing delay of *jt*^h PCE and D_l^j is the link delay between j^{th} PCE and its neighbor through a path to the destination.

For adjusting the timer with the network conditions and path length, we subtract link delay and queue delay of the message in each PCE. Let T_d be the obtained time by simulation (which is calculated in pervious subsection) and D_e^k be the end to end delay So we can calculate PCEs pre-reservation timer by the following relation:

$$T_{timer} = T_d - \sum_{k=1}^{i} D_e^k \qquad (5)$$

Where *i* is the number of PCEs from destination to the current PCE.

With this technique resources will be kept less time duration in the PCEs that are closer to the source. As a result, resources will be more available and we have more reply successes. So we can service the requests more. This improves the network utilization and reduces the blocking probability of the requests.

B. Resource reservation Reduction mechanism

A network with high number of nodes which use the flood operation, will quickly be congested and has a higher rate of blocking due to flooding too many messages. For achieving a uniform traffic load in network and higher network performance, we propose a model that uses both hop count and delay metrics in computing the end-to-end path. With this way, users can announce their required delay at first. The proposed algorithm probes paths with minimum hop counts that satisfy required delay. In other words, we prune paths that are congested and cannot satisfy required delay. So, with this way we can distribute the traffic uniformly on network and increase the network performance. So, distributing the traffic on network can improve network performance in delay sensitive applications. Because this model just does not choose the shortest paths .Then it is likely to be less congestion on network routes. As a result we can service to more requests with the required delay in this conditions.

For reducing reservations every request passing from a PCE, record its *id* and its *cost* to that PCE. Therefore, pruning has two main advantages:

- Fewer routes are reserved, so we have more free resources and as a result the rate of successful replies and network utilization is higher in this case.
- It reduces the network traffic load almost by half

When a reservation packet receives a PCE can do three operations as table 1:

TABLE I. PRUNING OPERATION

Operation	Description
New Packet	PCE records the packet Id and its cost in its table, then send it to neighboring PCEs
Better cost	PCE already received a packet with the same Id but its cost is greater than the received one so it record the new values of the Id and cost, then send it to neighbors.
Worse cost	PCE already received a packet with the same Id and its cost is lower than previous one So In this case, the new request has more cost, so we can delete it because we have a path with lower cost.

According to what is discussed previously, we flood the request between PCEs until it reaches to the destination. Flooding in networks that connectivity is high, imposes a large overhead. The message complexity in the worst case is O(N.D), where N is the number of nodes and D is the average degree of links at a node. Let deg_i be the degree of node *i*, [9] prove that the number of messages that are generated by each flooding procedure is *N.(D-1)+1*.

$$deg_1 + \sum_{i=2}^{N} (deg_i - 1) = 1 + \sum_{i=1}^{N} (deg_i - 1) \qquad (6)$$

$$= N.(D-1) + 1 \approx O\,(D.N)$$

This is the lower bound of flooding a request. But one node can receive the same request from all of its links and each time flood it to neighbors. Then we can rewrite the formula of equation (2.1) as follow:

$$D.\left[deg_1 + \sum_{i=2}^{N} (deg_i - 1)\right] = D.\left[1 + \sum_{i=1}^{N} deg_i - N\right] \qquad (7)$$

$$= ND.(D-1) + 1 \approx O(D^2.N)$$

C. Multiple Metrics Path Cost

Since we use multiple metrics to determine the path cost, we use a linear combination of different routing metrics according to following equation:

$$C_{PCE}^{i} = \sum_{j=1}^{K} C_{PCE}^{j} + M_i \tag{8}$$

$$M_i = \sum_{\forall k} \alpha_k * Metric_k, \qquad \sum_{\forall k} \alpha_k = 1 \tag{9}$$

Where C_{PCE}^{j} is the accumulated cost of previous PCEs along the path, M_i is the metrics values and α_i is the weight factor for $Metric_i$. This factor can be varied according to the application requirements, and this can change the importance of cost metrics during path computation.

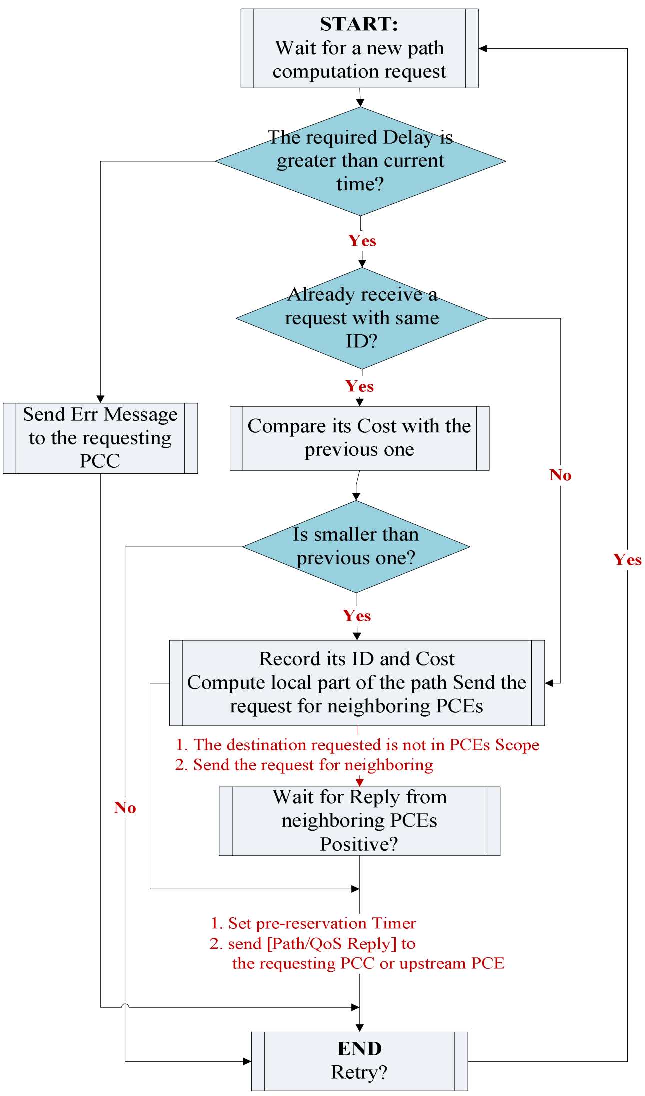

Figure 2. Reservation flowchart

So, when PCE receives a request with the same Id, it compares the cost of the received request with the previous one. Request are sent only when its cost is less, otherwise it is deleted. This cost can be calculated from equation (9).

We can choose the α_k of $Metric_k$ according to its importance. This can be announced by user at the time of sending the request. The idea of further investigating the use of priorities is left for future work as detailed in section VII.

Current reservation protocols are typically optimized with regard to one of metric. For example selects paths that consist of the shortest number of hops. Reservation metrics that are used to make this decision are:

- Delay
- Hop count

The cost of PCE $j(M_j)$ is computed as follows:

$$M_j = \alpha H_j + (1-\alpha) D_j \tag{10}$$

Where H_j is the number of hops to the destination through the PCE j and D_j is the path delay thought PCE j.

The algorithm at first eliminates the paths with delay larger than the requested delay. After that the shortest path is selected among the remaining path. As mentioned above, we can weight to each of these metrics according to its importance. Here we took the same weight in both. With this mechanism we can reduce the number of message flooding from order two to order one. This results in reducing the network traffic and increasing the utilization. The objective of the proposed multi metric QoS reservation protocol is to reserve paths by considering two different QoS parameters such as delay and hop count. Fig.2 illustrates the flowchart of the proposed algorithm. The experimental results show that the proposed multi metric QoS reservation protocol improves the network performance than single metric reservation protocol.

V. SIMULATION RESULTS

In order to evaluate the performance of the proposed protocol, both of the proposed protocol and [3] are implemented in Opnet v.14 simulator. For this purpose, a network topology as shown in Fig.3 is simulated.

The PCE request success is the ratio of successful replies to the maximum number of requests. The network utilization that we define it as the ratio of the successful deployments to the maximum number of requests and network load that is the maximum number of messages that can be exchanged in the network. In order to assess the performance of the proposed algorithms, the network has been examined in two scenarios: At first we evaluate the network parameters with considering both hop count and delay criteria. After that, we consider them by implementing the timer configuration scheme in both cases. At last, we added the flooding reduction technique.

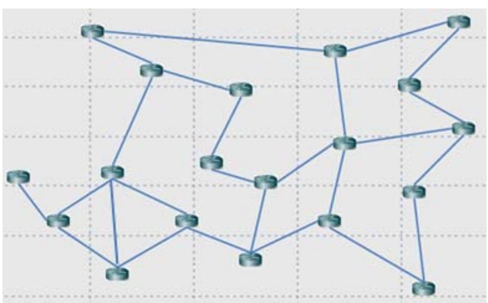

Figure 3. Simulation topology

A. Comparing multi-metric method with base method

At first we implemented the base method by considering two QoS criteria: hop count and delay. Fig.4 shows the amount of resources that are reserved in both case, with considering delay and without it. As we can see, by considering two metric instead of one we have reduction in the amount of reserved resources. As a result we have more available resources. This in turn, improves the rate of successful replies Fig.5 and reduces the request blocking ratio, Fig.6. All these factors also impact on network utilization, so you can see in Fig. 7 that we have an slight improvement in network utilization.

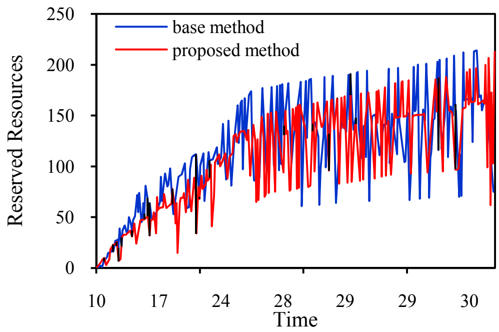

Figure 4. Reserved Resources

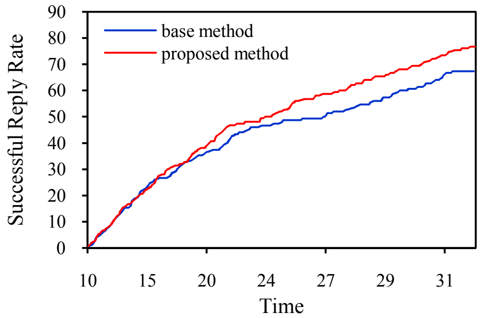

Figure 5. Successful Reply Ratio

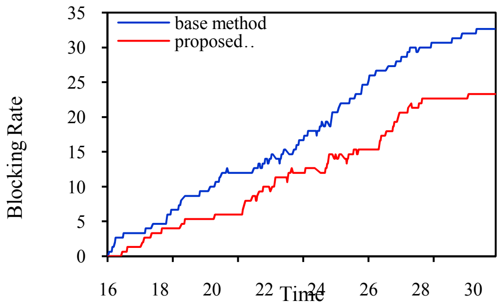

Figure 6. Request blocking rate

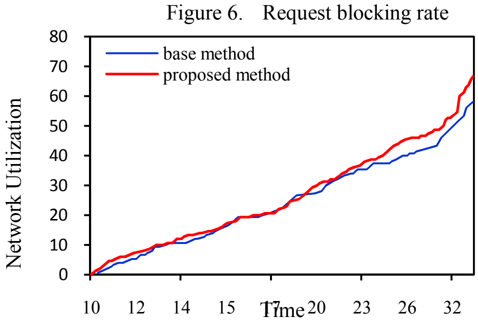

Figure 7. Network utilization

B. Evaluating reservation timer configuration method

In this section we implement reservation timer configuration method in both cases and comparing the results with base method. As seen in Fig.8 we have a significant performance gain in the terms of successful relies and network utilization. Adjusting the timer with network conditions help us to have more available resources. That's because the resources that are close to the source, are released earlier. So they are more available and this increase the successful replies and decrease the blocking ratio of the requests (Fig.9). As mentioned previously, this factors cause an improvement in overall network utilization (Fig.10). The results also show further improvement in these parameters when we consider both hop count and delay criteria.

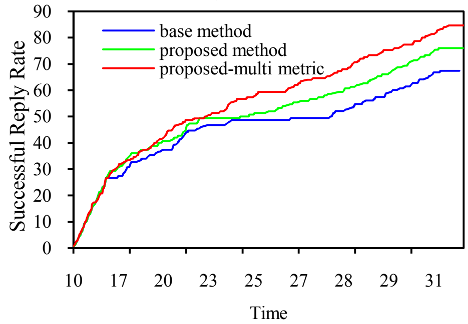

Figure 8. Successful reply ratio

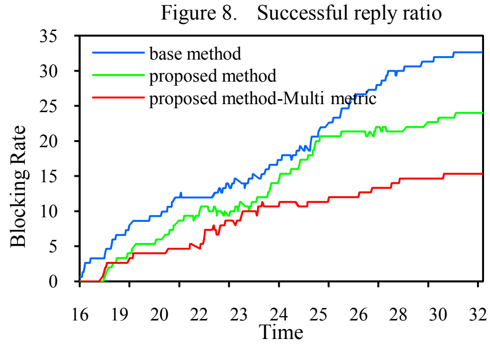

Figure 9. Request blocking rate

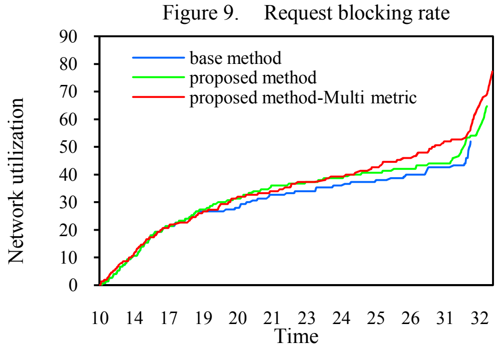

Figure 10. Network utilization

C. Implementing both the timer configuration and flooding reduction scheme

In order to evaluate the performance of the proposed protocol, both of the timer setup and flooding reduction mechanisms are implemented. As shown in Fig.11, in this case we have improvement over base method in all of the mentioned factors. But because we prune paths according to two metrics, in fact we choose the paths that satisfy both delay and hop count criteria. This causes to eliminate some longer paths that satisfy requested delay and make the remained paths more congested. This in turn increases the queuing delay. So, they cannot satisfy requested delay and we have more blocking here.

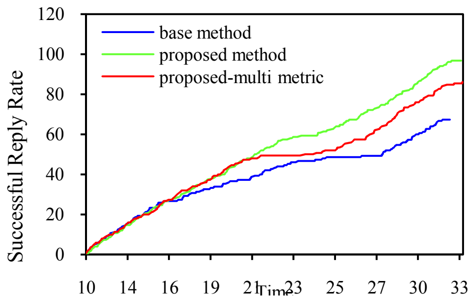

Figure 11. Successful Reply ratio

Having more reduction cause to resend some blocked requests. This increases the traffic load and has negative impact on network utilization, Fig 12-14.

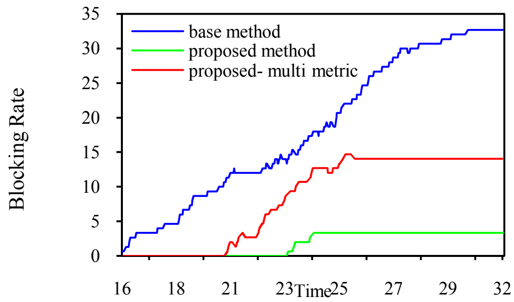

Figure 12. Request blocking rate

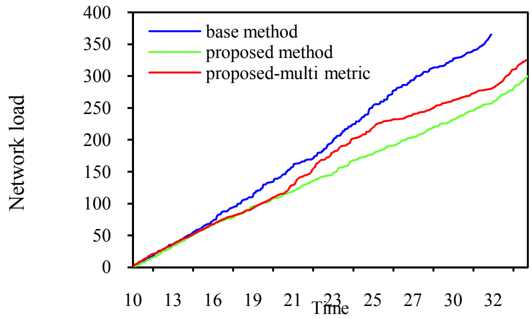

Figure 13. Network Traffic load

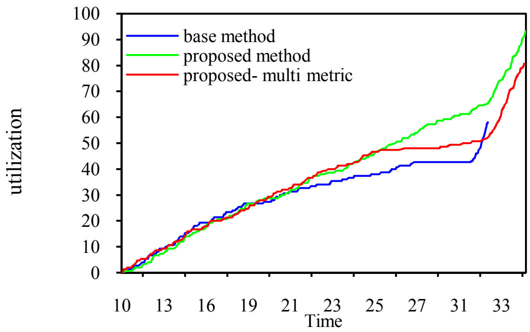

Fig. 14 network utilization

However we have significant performance gain over base method and other benefits in considering multi metric instead of shortest path algorithms.

VI. CONCLUSION

In this paper, we discuss the general issues of the constraint based path selection algorithms. We also propose an scheme that avoids blockage at the time of TE-LSP deployment in multi domain networks. It provides three mechanisms for increasing the network overall utilization. First, it increases the rate of successful replies by setting up the pre-reservation timer according to the path length and network conditions. Then, it reduces the overhead of the network by means of a pruning mechanism that prunes non promising paths and only pre-reserve the promising ones. Finally it defines a linear function for calculating the costs. The simulation results give conclusive insight to the advantages of the proposed solution. The results show that the proposed method increases the chance of the successful deployment of TE-LSP and decreases blocking probability without deteriorating the network utilization.

VII. FUTURE RESEARCH DIRECTIRIES

To continue the work presented in this paper, a mechanism to investigate the integration of request priorities into the proposed Scheme should be defined to obtain overall utilizations and blocking rates of requests in different priority categories. Future mechanism can also dynamically change the weight factor α_i according to the network conditions. It is also interesting to relate the request priorities to the solution for the correct setting of pre-reservation timers.

REFERENCES

[1] A.Farrel, J.P. Vasseur, J.Ash, "A path computation element (PCE)-based architecture", , IETF , RFC 4655, August 2006.

[2] Q. Zhang, M. Masud Hasan, Xi. Wang, P. Palacharla, M. Sekiya, "Efficient PCE-Based Survivable Path Computation in Multi-Domain Networks", Workshop at IEEE infocom, 2011.

[3] M. Shirazipour , S. Pierre, " End-to-end path computation schemes for traffic engineering in next generation multi-domain networks", Ph.D thesis, Montreal University of Canada, Juin 2010.

[4] B. Kim, S. Yoon, Y. Jeong, H. Song , "Implementation of Inter-Domain Path Computation Element", Information Networking (ICOIN), 2012 International Conference, pp. 501 - 505, 1-3 Feb. 2012.

[5] J.LEROUX, "Path Computation Element Communication Protocol (PCECP) Specific Requirements for Inter-Area MPLS and GMPLS Traffic Engineering", , IETF, RFC4927, 2007.

[6] M.Chamania , A. Jukan , "A Survey of Inter-Domain Peering and Provisioning Solutions for the Next Generation Optical Networks", IEEE communications surveys & tutorials, vol. 11, no. 1, first quarter 2009.

[7] L. Gommans, F. Dijkstra, C. de Laat, A. Taal, A. Wan, B. van Oudenaarde, T. Lavian, I. Monga, and F. Travostino, "*Applications Drive Secure Light path Creation across Heterogeneous Domains*", IEEE Commune. Mag., vol. 44, no. 3, pp. 100-106, Mar. 2006.

[8] T. Okumus, H. Junseok, H. A. Mantar, S. J. Chaplin, "*Inter-domain LSP setup using Bandwidth Management Points*," IEEE GLOBECOM 2001, pp. 7 - 11

[9] J.P. Vasseur, A. Ayyangar, R. Zhang, "A per-domain path computation method for establishing inter-domain traffic engineering (TE) label switched paths (LSPs)" , IETF, RFC 5152, February 2008.

[10] J.P. Vasseur, R. Zhan, N. Bitar, J. Le Roux, "A backward recursive PCE-based computation (BRPC) procedure to compute shortest constrained inter-domain traffic engineering label switched paths", IETF, RFC 5441, April 2009.

[11] H.A. Mantar, J. Hwang, I.T. Okumus, and S.J. Chapin, "A Scalable Model for Inter bandwidth Broker Resource Reservation and Provisioning", IEEE Journal on selected areas in communications, vol. 22, no. 10, pp. 2019-2034, December 2004.

[12] S. Sanchez-Lopez, X. Masip-Bruin, J. Sole-Pareta, and J. Domingo-Pascual, "Fast setup of end-to-end paths for bandwidth constrained applications in an IP/MPLSATM integrated environment", Computer Networks, vol. 51, no. 3, pp. 835-852, February 2007.

[13] **G. Hernández-Sola , J. Perelló , F. Agraz, L. Velasco Salvatore Spadaro , G. Junyent ,"** Enhanced domain disjoint backward recursive TE path computation for PCE-based multi-domain networks", Springer Science+Business Media, LLC 2010.

[14] S. Salsano, A. Botta, P. Iovanna, M. Intermite, A. Polidoro, "Traffic engineering with OSPF-TE and RSVP-TE: Flooding reduction techniques and evaluation of processing cost", Computer Communications 29 , pp. 2034–2045, January 2006.