International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 3 Issue 11, November-2014, pp: (172-185), Impact Factor: 1.252, Available online at: www.erpublications.com

Efficient Service Provisioning with Protection using Path Protection Algorithms for SONET/SDH Networks

Deepak Dhadwal[1], Ashok Arora[2], V. R. Singh[3]
[1]Research Scholar, Manav Rachna International University, Faridabad, Haryana
[2]Professor, Manav Rachna International University, Faridabad, Haryana
[3]Director, PDM College of Engineering, Bahadurgarh, Haryana

---

Abstract: For efficient and very high capacity network for Telecommunication and Data services the SONET/SDH network is used. The Service providers are keenly interested to find automatic ways of commissioning the channels associated with the SONET/SDH. The TDMA technology is used in the SONET/SDH network by which the High speed data communication and telecommunication traffic is handled. Due to very high bandwidth of optical network, this has been proved that manual provisioning is not possible efficiently. Another thing which is hard is to cover the protection of the optical network. In this paper, Automatic Path protection algorithms has been implemented and analyzed. The algorithm is suitable for the VCAT like issues. The Dedicated and shared path protection mechanism has been used for the efficient use of the SONET/SDH network. In this paper the basic parameters like probability error, QoS of network, Accepted request and Rejected request have been analyzed by our algorithm. We also investigated the Bandwidth constraint. Using our algorithm we have improved the bandwidth utilization.

Index Terms: SONET/SDH network, Multipath protection schemes, Dedicated path protection and Shared Path protection Multicontraint paths, VCAT, ATM.

I. INTRODUCTION

Bandwidth Requirement is one of the critical factors in the current environment of high speed communications. Network operators and service providers are too much affected from the efficient bandwidth requirement. Because of this the engineers are finding emerging trends in communication like PDH, WDM, DWDM, ATM, SONET /SDH etc. There is also need of providing different QoS for different users. This directly affects the revenue of Service providers [1].

In this era, a large number of service request have been investigated under the consideration that the network conditions are too dynamic depending on the load of the network dynamic service request and the increment in bandwidth channels. The main point which is important in this case is the efficiency of the network with the dynamic changes in the network parameters. Faster provisioning of services in less time to handle the more customers is important [2].

There is lot of research conducted in the domain of the automatic Service provisioning algorithms. In this kind of algorithm it is important to know the customer requirement. The Service providers check the networking issues put the bandwidth requirement for the user and adjust the quality of the service according to the users demand. For such systems inventory details are needed from the database. EMS i.e. Element Management Systems or Network elements are needed to access so that the network resources can be used easily. Working path information is gathered by the algorithm for the good use of the available paths and to provide the protection to the network. The parameters needed to provide service provisioning are working path, used path, free capacity, cost to the path etc.

In SONET/SDH network, there is need to establish working and protection path for reliable operation [3]. Dedicated and Shared Path protections are the two techniques used for efficient use of Bandwidth in Network [4]. In dedicated-path protection, a protection path to protect a particular working path exclusively is provided, whereas in shared-path protection, a protection path can be shared by many working paths. In both cases, the constraint is that a working and its protection path have to be diversely routed so that at least one path can survive a single failure in the network [34].

E1, E4, DS1, DS3 etc. are the signals which are used by the SDH network which is mentioned in G.703. The protection provided by the SDH is Multiplex Section Protection (MSP), Multiplex Section–Shared Protection Ring (MS-SPRing)

[5], and Subnetwork Connection Protection (SNCP) [5]. In SONET/SDH it is very hard to provide the protection to the failure. The complexity is due to very high traffic used by the service providers. Stopping the traffic at such a high level is too complex and hence need of automatic protection mechanism. In this paper, Section II has the detailed study of Dedicated and Shared Path protection mechanism. Section III has implementation of Dedicated and Shared path protection mechanism and finally the result has been compared with the available researched in section IV and concluded in Section V.

II. Path Protection

1. Using Dedicated Algorithm

Lets' first understand the Dedicated Path Protection algorithm. For dedicated-path protection, a working path and a link and/or node-disjoint protection path that protect only failures in that working path have to be found [34]. A polynomial time optimal algorithm to find a link-disjoint path pair is proposed in [24]. Computing link-disjoint paths for QoS routing under multiple constraints is addressed in [25].

Consider the network shown in Fig. 1. The network is created of following:

1. Physical links (Solid Lines)
2. Logical Links (Trails)
3. Nodes

This portion has been taken from our last research [21]. Problem statement is that to provide service to the service request from a to m. abclm is the shortest path computed from a to m and next generated shortest path is ahjkem. Overlaps for such cases are dk since the trails cl and ke have a traversing through them. So these two paths cannot make the disjoint path pair. The link dk will be interlaced by the computed path pair and we can obtain the abcdem and ahjklm finally after removing it. This condition is associated with [25] and the following points are important to discuss:

If two disjoints paths are Pd1 and Pd2 and P1 is the shortest path belonging to them. Then

1. P1 itself is Pd1 or Pd2 , i.e.P1=Pd1, or P1=Pd2 ;
2. P1 overlaps with both paths Pd1 and Pd2, i.e.,P1∩Pd1= φ , P1≠Pd1 and P1 ∩ Pd2= φ, P1≠Pd1.

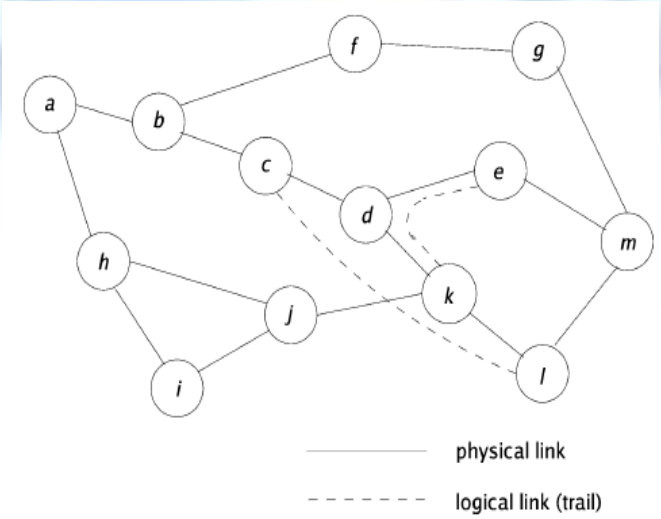

Fig. 1: A sample Network[]

Dedicated Path protection mechanism is used here for SDH networks. The constraints used in this case are the SDH multiplexing hierarchy and the standard protection mechanism used in [34].

Two protection paths will exist for those segments of the working path that have standard protection mechanisms, which results in unnecessary wastage of bandwidth. For traffic from b to m in the network shown in Fig. 2 [21], let us assume the working path is badfghim. This path contains ba, which is part of an MSP, fghi, which is part of an MS-SPRing, and the trail im, which is part of an SNCP [34]. In this case, the protection path can use the inherent protection paths corresponding to those segments and use a disjoint path for the other segments, which in this case is adf instead of finding a disjoint protection path from b to m.

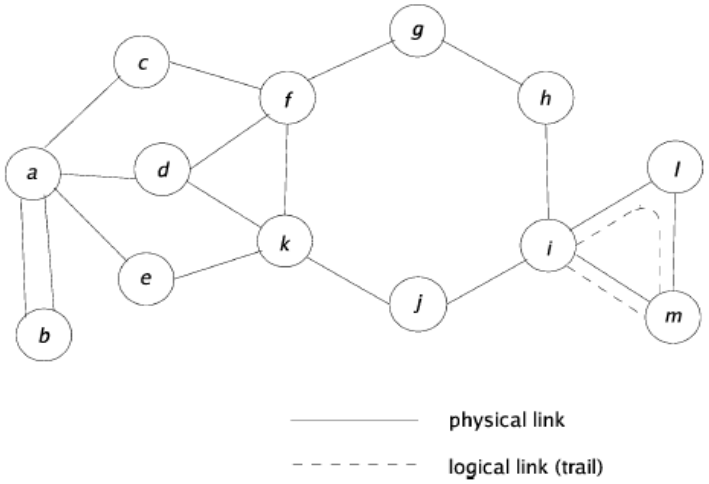

Fig. 2: A sample Network with Standard SDH protection Scheme[21]

2. Using Shared Algorithm

a. Network Model[4]:

In our previous research we have investigated the shared path protection [4]. D(N,L,W), the topology of network, where N is a node sets, L is a bi-directional link sets and W is a set of wavelengths per link. Each link is allocated a capacity of eight wavelength channels. Here, it is assumed that all nodes have wavelength conversion capability. It is also assumed that only one connection request arrives at any time point. A_N, availability matrix, has the availability values of all the links in the network. The network state is compromises of two other such matrices λ_w and λ_p which respectively store the number of working and protection wavelengths being utilized on each link at any time point. λ_p includes information about connections that are sharing protection wavelengths on a particular type of link. The traffic is dynamic and connection requests arrive without knowledge of subsequent arrivals. A connection request is described by r(s, d, a_r), where source node is 's', destination node is 'd' and availability requirement of the request is 'A_{req}'.

b. Traffic Model[4]:

Telecommunication traffic is defined as the average number of connections in progress. A widely accepted approach to dynamic traffic modeling has been adopted in which the arrival of connection requests is a Poisson Process with a constant arrival rate ß. The arrival rate is the rate at which connection arrival requests are received by the network per unit time. This model is popular since it realistically describes the arrival of connection requests, which arc independent of each other. The connection holding time refers to the time duration of a connection from its establishment to its termination. Holding times are random and a negative exponential distribution, with a mean of l/μ, is used as time for holding. Erlangs is the measurement for traffic which is a dimensionless unit. In general, Erlang traffic ($T_{Erlangs}$) is defined by the equation

$$T_{Erlangs} = \beta(1/\mu) \quad (1)$$

where ß refers to the connection arrival rate and 1/μ, the mean holding time. In this study, the time measurements have been normalized by assuming 1/μ = 1 so that the network traffic load can be considered, in units of Erlang as being equivalent to β. Destination and source nodes are randomly chosen which allows for more than one connection to be established between any pair of nodes. The implementation of waiting queue is not done and if the connection is not established by the algorithm, then it is immediately rejected or blocked.

c. Connection Availability Analysis[]:

Here, reliability is measured using availability since availability denotes the time percentage that a connection will be in its normal operating state at any random point in time. Here, availability is defined and calculated for an end to end connection that is established as either a working path or a combination of working and backup paths. Connection requests that meet their fault tolerance requirements (A_{req}) are called dependable connections. Furthermore, availability is an important decision criterion, used in network planning and dimensioning studies as it is often indicated in SLAs between service providers and customers. It is assumed that only one link fails at a time and that the MTBF(Mean Time Between Failure) and the MTTR(Mean Time To Repair) are independent, memory less processes. Due to the greater

effect that link failures have on network performance, other network components' availability such as amplifiers and nodes has been neglected and assumed being 1.

The following notation has been used:

ij: i and j are the link connecting nodes which is represented by two unidirectional fibres.

a_{ij}: the availability of ij.

c_{ij}: the ij cost, determined by ij availability and and the ij wavelength assignment.

a_{path}: arbitrary path availability, consists of series of connected links.

a_{LDP}: the availability of a Link disjoint pair or working and protection paths.
a_{wp}: the availability of a link disjoint working path.
a_{pp}: the availability of a link disjoint protection path.
a_{PLDP}: the availability of a partial Link disjoint pair of working and protection paths.
S_1 : a set of links common to both the working and protection paths of a partial link disjoint path pair.
S_2: a set comprising the links of all link disjoint path segments of a partial link disjoint path pair.

LDP_k: the kth Link disjoint segment of a partial link disjoint path pair.

wp_k: the working path of the L- link disjoint path segment.

pp_k: the protection path of the k link disjoint path segment.

ξ: the link disjoint parameter defined between 0 and 1.

θ: the spare capacity usage factor.

III. IMPLEMENTATION

a. Dedicated Path Protection implementation

Network Engine implemented on the base of the above said algorithms. Following points implemented for the dedicated path protection mechanism:

1. Simulation consist of Two distinctive interactive Units:
2.
    a. By playing of Network
    b. And, By playing the Request

3. **By Playing the Network:**
   Network engine create the network in the back end and accept some important issues to simulate the network

    a. This is a Network Engine which runs in backend
    b. It has 2 Major controls:
        i. Start/Pause/Resume Network
        ii. SONET/SDH State viewer

4. **By Playing the Request**

    a. This is implemented in the frontend.
    b. It has all the controls to generate/process requests
    c. Key controls are:

        i. Posting a User request
        ii. Randomize request parameters
        iii. Randomize + Post request (for ease of analysis)
        iv. Auto – Request Generator

Algorithm 1 Algorithm for Dedicated Protection(source,dest,rate)

1: initialize optimum link disjoint path pair, P_w, $P_p \leftarrow NULL$
2: initialize the cost of optimal link disjoint path pair, $C(P_w, P_p) \leftarrow \infty$
3: $i \leftarrow 0$
4: **while** $i < K$ **do**
5: $i \leftarrow i + 1$
6: let $P_1 \leftarrow NULL$, $P_2 \leftarrow NULL$, $C(P_1, P_2) \leftarrow \infty$
7: compute the $(i)^{th}$ shortest path p_i using Yen's algorithm
8: **if** $p_i = NULL$ **then**
9: break
10: **if** capacity not available in the path p_i **then**
11: continue;
12: **if** $i = 1$ **then**
13: $invertFlag \leftarrow$ **true**
14: **else**
15: $invertFlag \leftarrow$ **false**
16: remove all the trails except those in p_i that traverse the links traversed by p_i
17: remove all the links traversed by p_i but not in p_i
18: $protectionPathFound \leftarrow$ **true**
19: break the path p_i into segments that are either part of or not part of standard SDH protection schemes
20: **for** each segment s_j in p_i **do**
21: **if** s_j is part of some standard SDH protection scheme **then**
22: add s_j to P_1
23: add the inherent protection path corresponding to the protection scheme used to P_2
24: **else**
25: $(p, q) = EPP$ (source of s_j, dest of s_j, s_j, rate, $invertFlag$)
26: **if** $(p, q) \neq NULL$ **then**
27: add p to P_1
28: add q to P_2
29: **else**
30: $protectionPathFound \leftarrow$ **false**
31: break
32: add those links and trails that were removed in step 16 and step 17 back to the graph
33: **if** $protectionPathFound \neq$ **true then**
34: continue;
35: **if** $C(P_1, P_2) < C(P_w, P_p)$ **then**
36: $(P_w, P_p) = (P_1, P_2)$
37: $C(P_w, P_p) = C(P_1, P_2)$
38: **if** $i = 1$ and there is no segment in the graph with any standard SDH protection scheme **then**
39: **return** (P_w, P_p)
40: **if** $C(P_w, P_p) \neq \infty$ and $C(p_i) > C(P_w, P_p)$ **then**
41: **return** (P_w, P_p)
42: **return** (P_w, P_p)

Algorithm 2 EPP(src,dst, P_1, rate, *invertFlag*)

1: **for** (i, j) in P_1 **do**
2: remove the directed edge (i, j)
3: **if** *invertFlag* = **true then**
4: $C(j, i) \leftarrow -C(j, i)$
5: **else**
6: $C(j, i) \leftarrow 0$
7: P_2 = ShortestPath (src,dst,rate)
8: add the removed edges back to the graph and revert back to the original weights for those edges for which the weights were changed
9: **if** $P_2 = NULL$ **then**
10: **return** $NULL$
11: Take the union of P_1 and P_2, remove from the union the links and trails that are part of both P_1 and P_2 and then group the remaining links and trails into P and Q
12: **return** (P, Q)

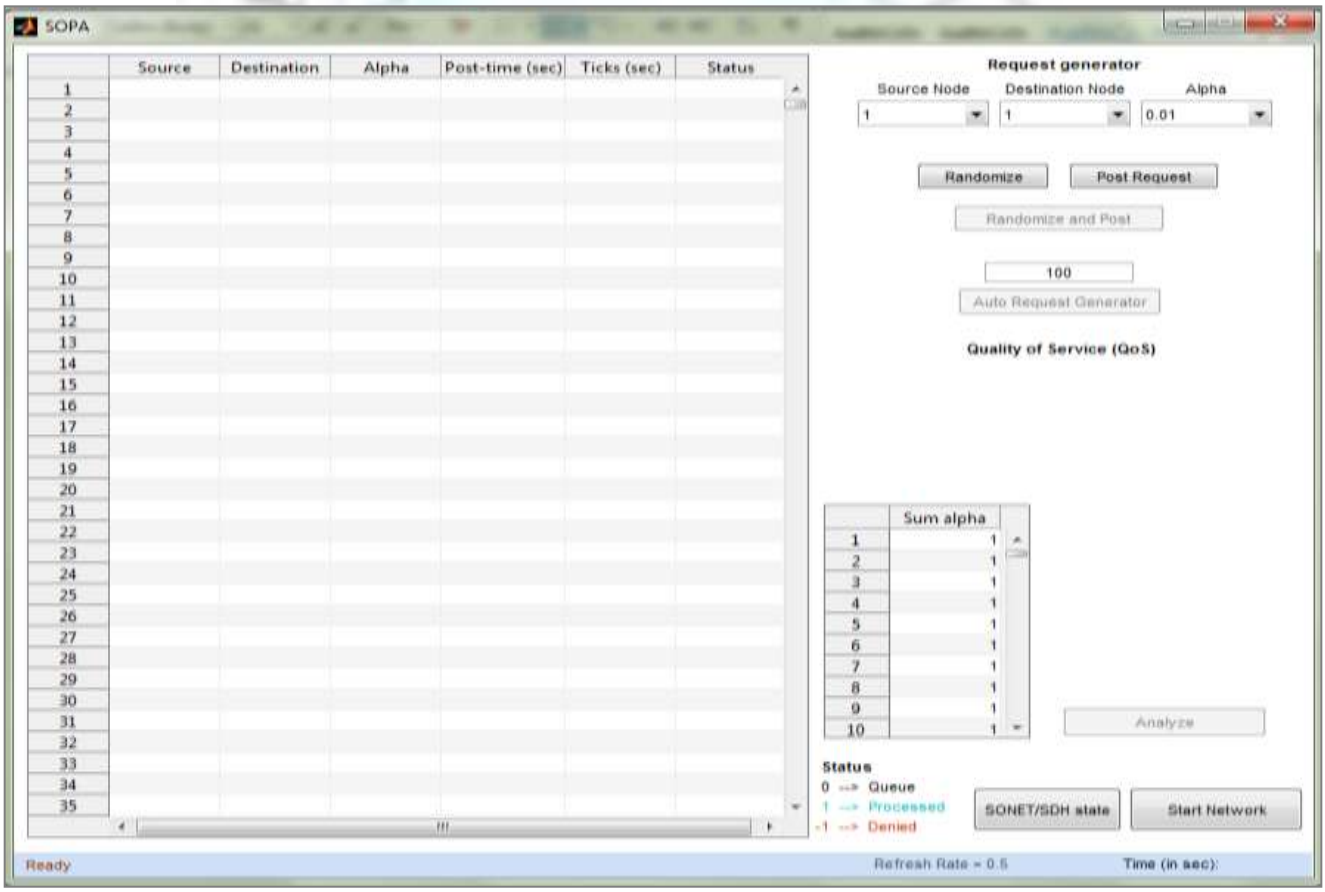

Fig. 3: MATLAB Network Engine for Dedicated path protection

Figure 3 provide the view of working platform. It shows a view of dedicated path protection mechanism. The application is able to take the input parameter like source, destination and value of alpha. Alpha is a dynamic path cost which value vary from 0, 0.1 to 1.0. This value defines the associated path cost from the source node to the destination node. This is one kind of the bandwidth Utilization factor. The alpha also has the effect of the distance from the source to destination. All these things creates cost to the path of the network of SONET/SDH network.

The Engine can be analyzed using the posting the requests and run the requests over the defined network. This point provides flexibility to check the processed requests and generation of requests. The Engine can directly implement on the real time network. The engine is providing good results for the advanced system design for networking of SONET/SDH Networks. The Network engine can be enhanced further using other implications for the Network design issues which affects the Network.

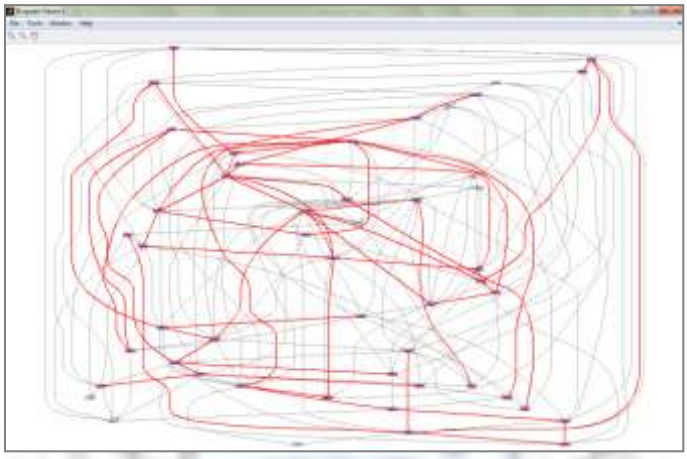

Fig. 4: A simple Network with connected and unconnected nodes using dedicated path protection

In Fig. 4, a network with 120 nodes has been shows with about 70 requests at a time. A Really important Backend parameter **"Refresh Rate"** is also available for view only to the system administrator, to check out the frequency or timeout of each request acceptance/denial.

b. Shared Path Protection Algorithm

SONET is a TDM system with 125μs time slot, and the delay of a SONET path is proportional to the path length [34]. So, shortest path can provide the least delay in the network. There are some of the shortest paths algorithms need to discuss. The example for such kind of shortest path is Bellman Ford Algorithm. The empty slots of the path can be used for commissioning of the other traffic. The Bandwidth can be utilized easily and more versatile in Shared path protection mechanism.

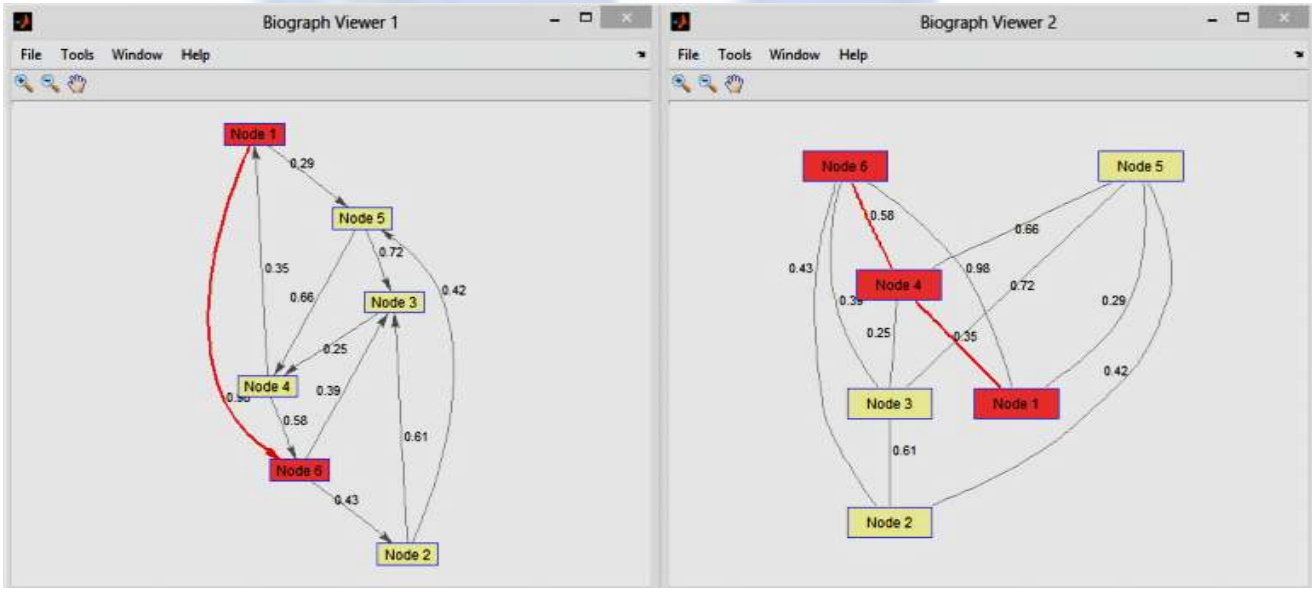

Figure 5: Shortest Path Algorithm

In figure 5, Bellman ford algorithm is used for the shortest path finding in the graph. The graph is weighted corresponding to the Bandwidth, distance etc. The graph shows a multi-node network. The selected shortest path can guarantee the shortest delay. The successional algorithm terminates when it finds the two paths (working path and protection path) and successfully reserves bandwidth requirement along them.

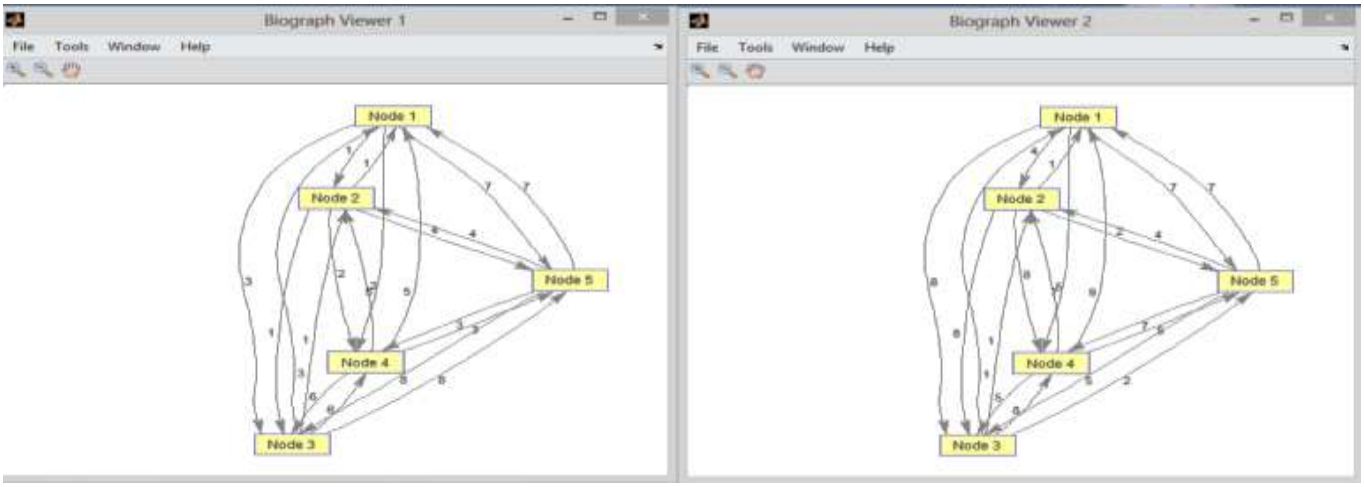

Figure 6: Bellman Ford Algorithm

The cost of the network is also minimized using the above network (Figure 6) about 10% of CSP and RASP

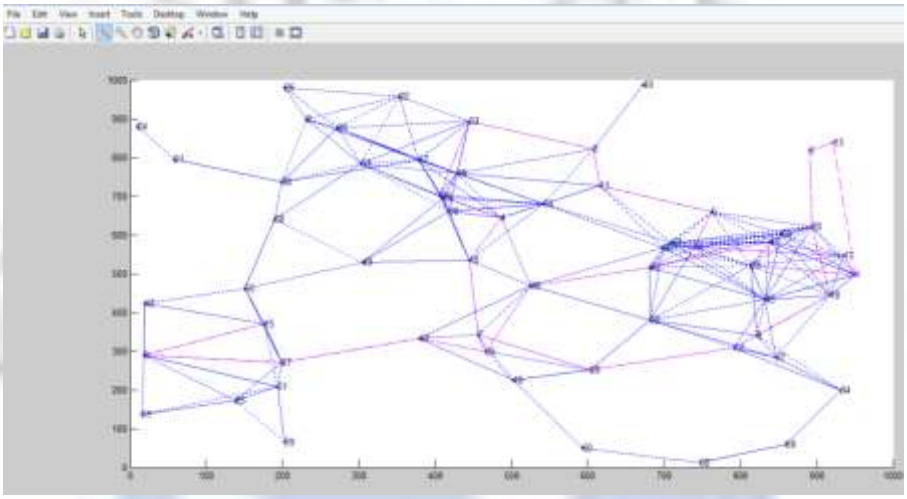

Figure 7: CSP and RASP

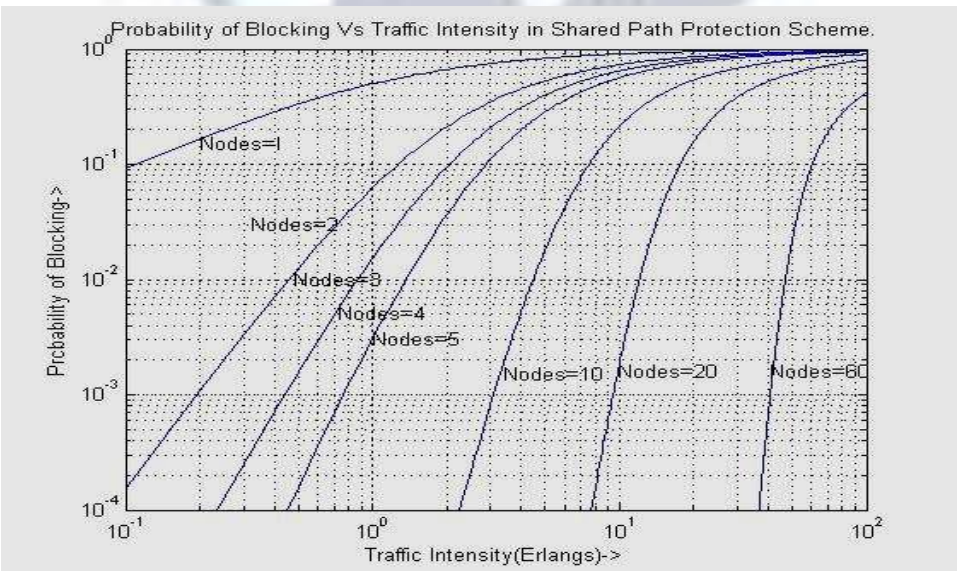

Figure 8: Blocking Probability Vs. Traffic Intensity

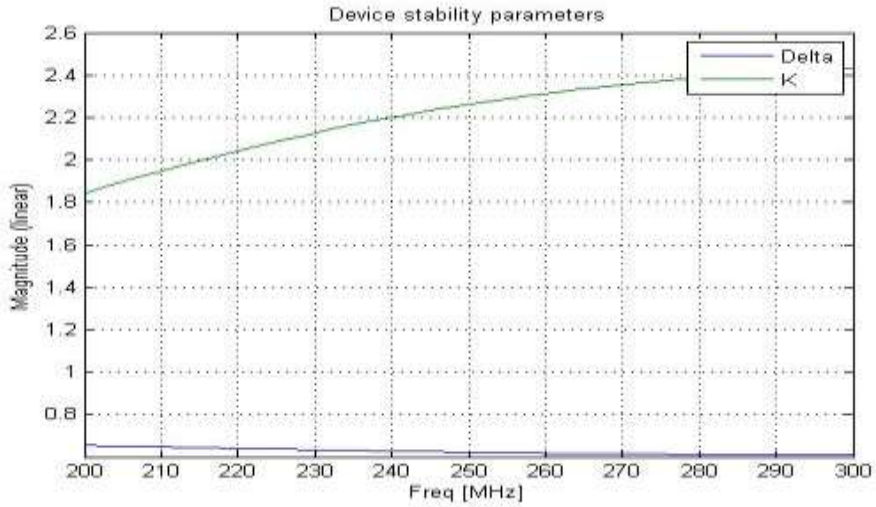

Figure 9: Magnitude vs. Frequency

The systems which we are using are the electronics systems. So the algorithms will be implemented over the electronics systems. Because of the we also have to measure the system performance by calculating the parameters like the Centre frequency (Hz), Transducer gain target (dB), Max noise figure target (dB) Source impedance (Ohm), Reference impedance (Ohm), Load impedance (Ohm), Lower band edge, Upper band edge, Frequency (radians/sec). This also Analyze the unmatched amplifier.

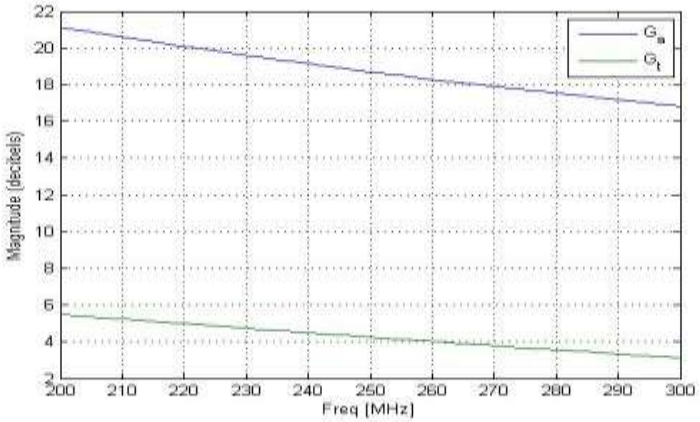

Figure 10: Magnitude vs. Frequency

Two graphs has been plotted regarding the bandwidth utilization and throughput. The alpha factor is some value between 0 and 1, and it refers to the relative weight of a trail.

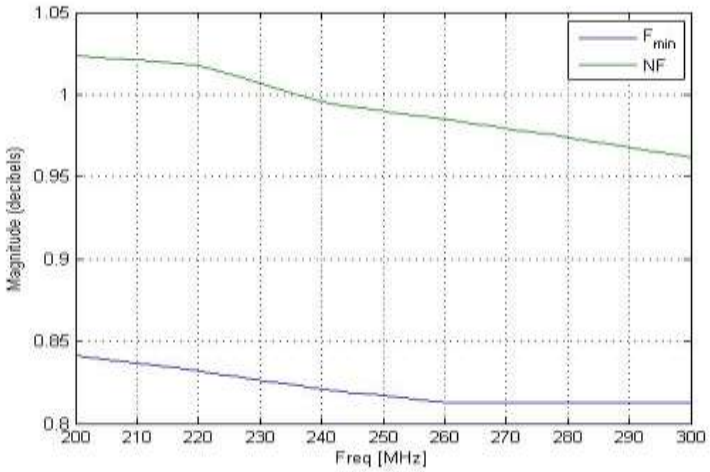

Figure 11: Magnitude vs. Frequency

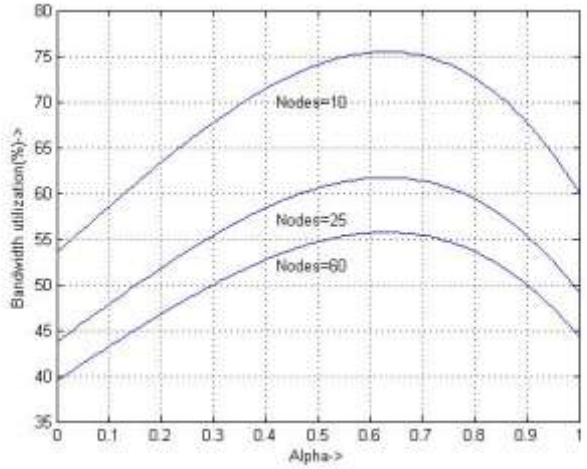

Figure 12: Bandwidth utilization Vs. Alpha

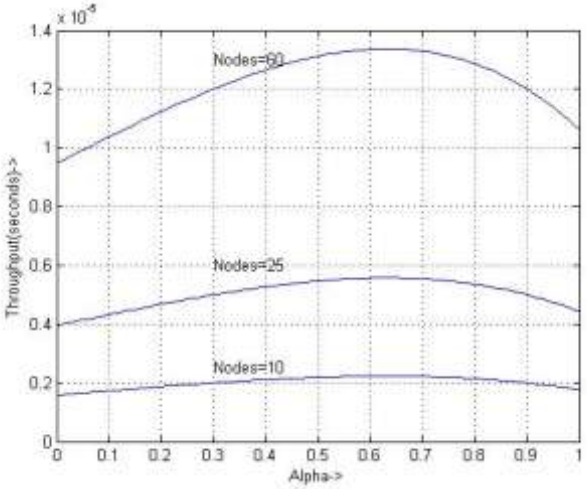

Figure 13: Throughput Vs. Alpha

Here, in each experiment, the following parameters are considered:
a) Weighted number of service requests accepted;
b) Number of service requests rejected;
c) Number of trails created;
d) Percentage of the total bandwidth consumed to satisfy the accepted requests.

IV. Analysis of Results

1. Dedicated Path Protection Results:
   The algorithms implemented using MATLAB and analyzed and evaluated. The comparisons were performed on the base of the following:

- Probability of blocking Vs Traffic Intensity
- Bandwidth Utilization Vs Alpha
- Throughput Vs Alpha

- Computational Complexity:

The computational complexity of Conventional Algorithm is, O ($KV(V^2+(E+V)\log V)$) where V is the number of network nodes, E is the number of edges, and K is the number of distinct paths. The complexity of the proposed algorithm is is $O(KV(E+V)\log V)$.

- Comparison between Probability of blocking Vs Traffic Intensity:

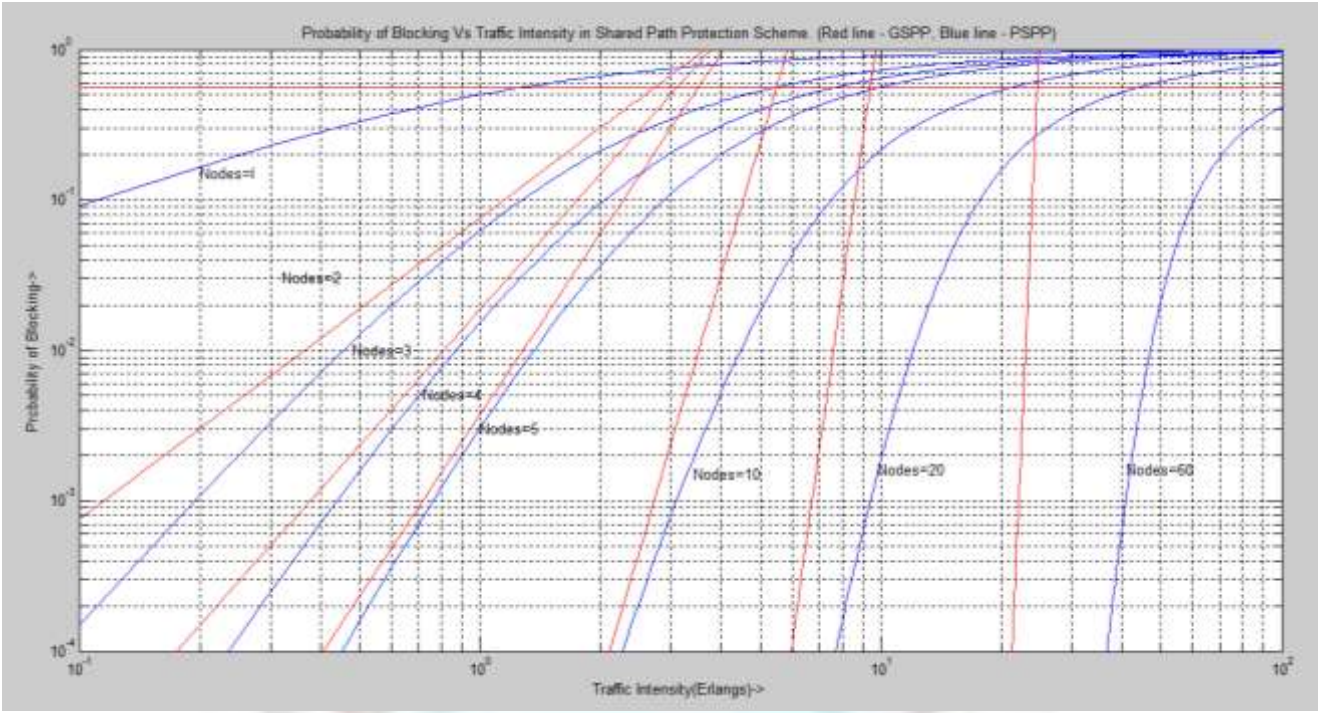

Figure 14: Blocking Probability vs. Traffic Intensity

In figure 14 there is a comparative study between the existing algorithms Vs. proposed algorithm. At different nodes a blocking probability Vs Traffic intensity.

- Comparison between Bandwidth Utilization Vs Alpha:

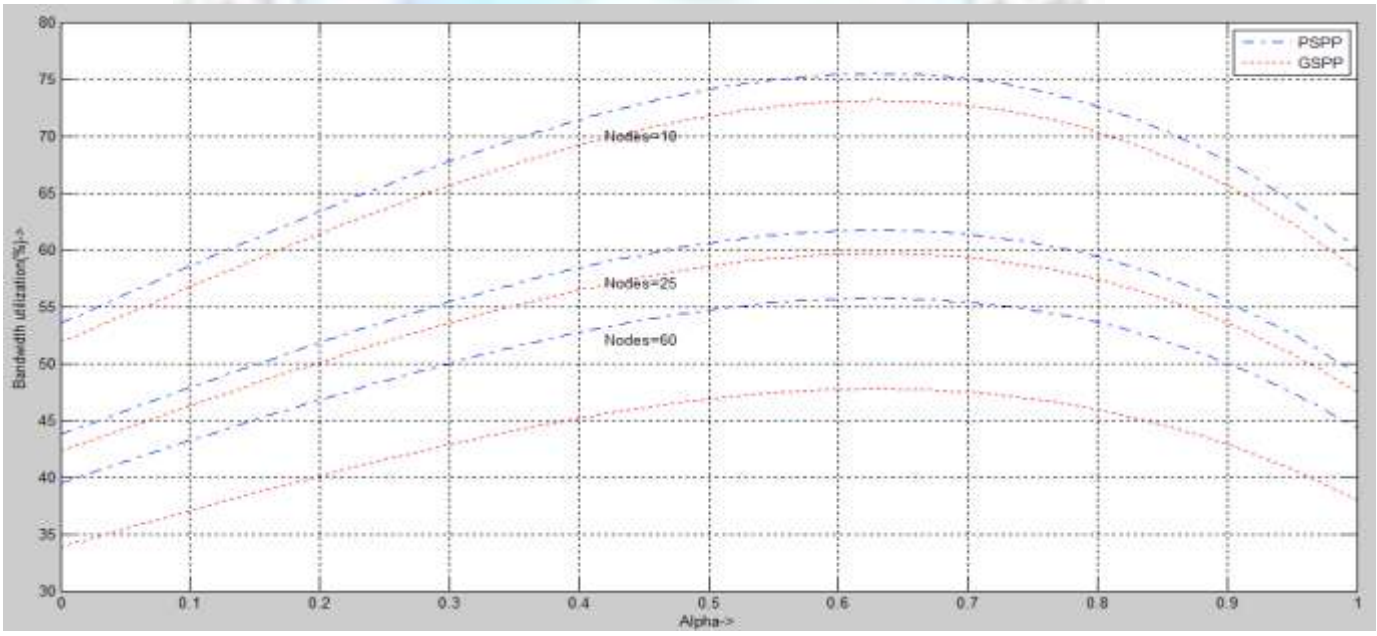

Figure 15: Throughput Vs. Alpha

Making comparisons between GSPP Algorithm and PSPP Algorithm performance, using some standard values, in blocking probability, throughput and bandwidth utilization; it is clearly observed that PSPP Algorithm is better than GSPP Algorithm in all performance criteria.

A. Data Computation

1. Number of requests **Accepted** and **Rejected** vs **Alpha**

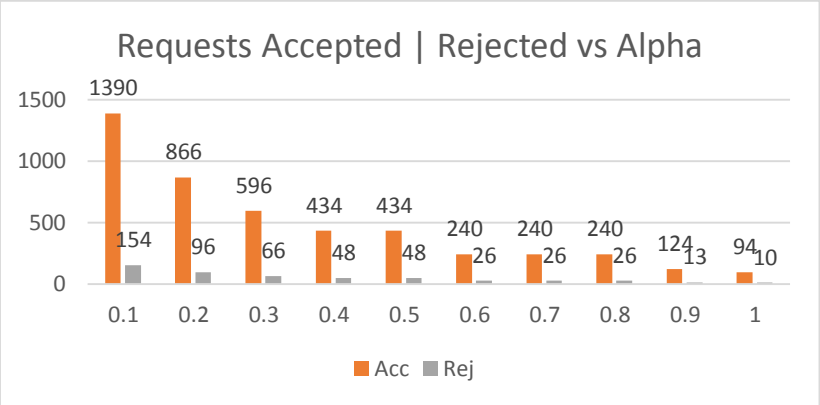

Fig. 16: Accepted and Rejected Requests Vs Alpha

2. **Blocking Probability** vs **Time**

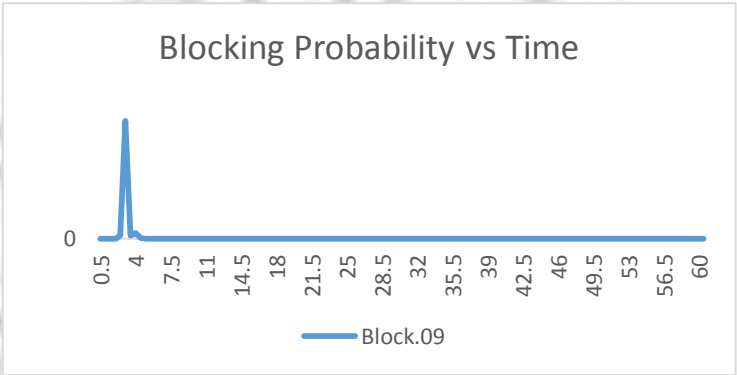

Fig. 17: Blocking Prob. Vs Time

3. **QoS** vs **Time**

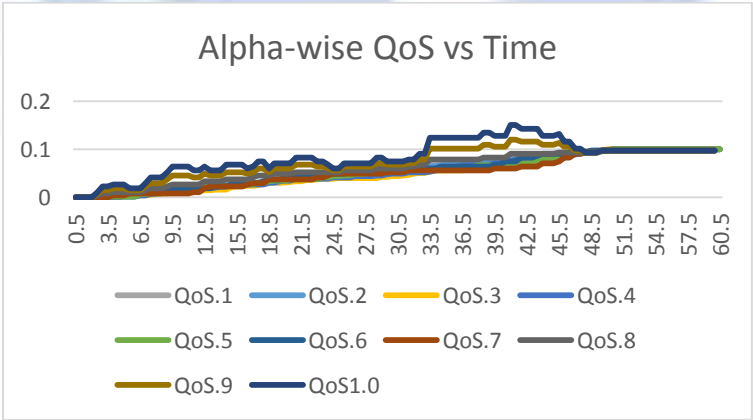

Fig. 18: QoS Vs Time

Now, since we gathered these parameters and their reactions for the chosen network conditions and load, we can analyse in-depth. The following inferences we can make now:

1. **Bandwidth utilization depends on Alpha values:**
   Alpha (α) values range between 0.01 and 1.00 with a precision up to 2 decimal points. We can check from the algorithm that on the value of the dynamic weight and demand of the bandwidth the paths are assigned.

2. **QoS decides the Ideal Alpha:** The alpha conditions provide the QoS. No. of accepted request defines the QoS. QoS is will be increased as alpha is increasing till the value of 0.8 and then saturates.

B. Analysis of whole scenario of Path Protection

- As we all know, if there is an increase in the call drop rate, the QoS will also increase, and so the network performance degrades. [QoS= (No. of Dropped calls)/ (Total No. of Calls), in a period of time]
- Initially QoS will increase with time and then converges to a constant value for the rest of the period.
- QoS is more and also rapidly increase for higher value of Alpha, so choosing more Alpha will leads to poor performance of the network.
- So, if we take the Alpha value to be least, it would also not solves the problem as its QoS increases with time in a linear way.
- From the graph, if we select Alpha value to be 0.3 or 0.7, then we got the least effective QoS for the network. On these values system performs at its best.
- In addition to this, if we implant a method to mix-match alpha values for the whole system that to be changed as per the call flow rate-changes, we could achieve an IDEAL SONET system. We can implement this by using an Automated Alpha scheduler. Let's say, for the experimented SONET network, we can set Alpha=0.7 for first 10 min., then Alpha=0.3 for next 20 min., and then finally we can have Alpha=0.7 again in the last 30 min. This will give ideal results for a particular call flow scenario.

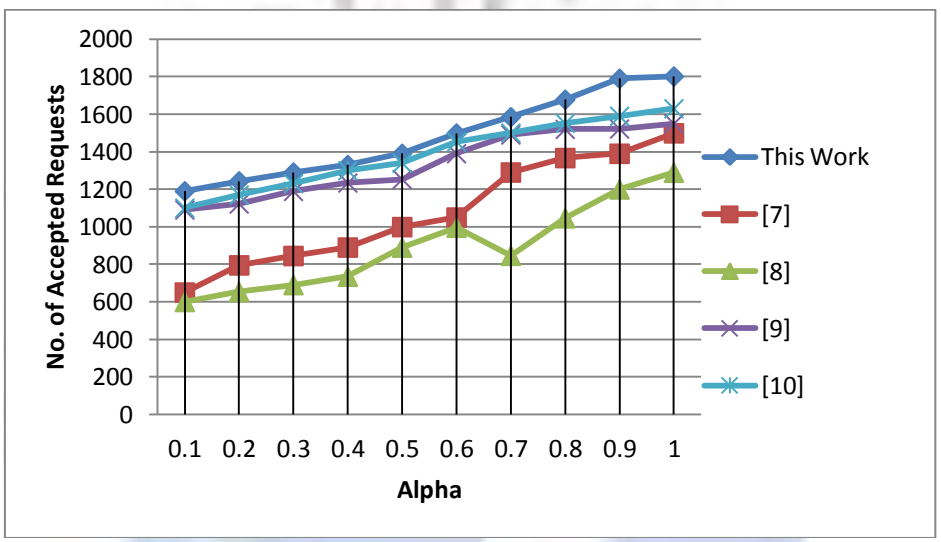

Figure 19: No. of Accepted Request Vs. Alpha

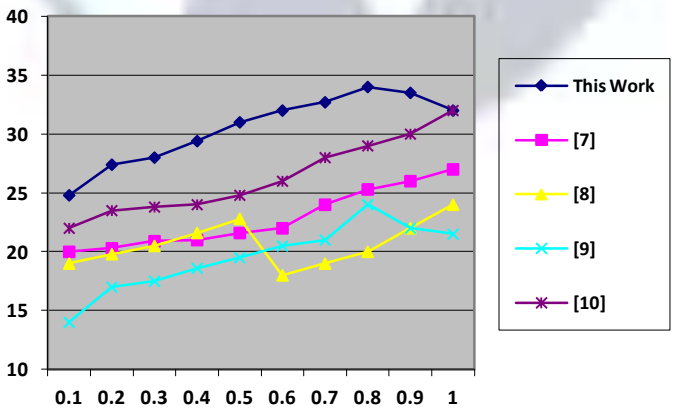

Figure 20: Bandwidth Utilization Vs. Alpha

Figure 19 shows the no. of accepted request Vs. Alpha for different studies. In our study and analysis the number of accepted requests are more than that of available study work. Figure 20 shows the bandwidth utilization percentage Vs. alpha. We have improved the algorithm and at 0.8 the work is showing maximum bandwidth utilization percentage.

V. Conclusion

We have improved the no. of accepted requests through our dedicated and shared algorithm. We have analyzed the dynamic weight and the dependence of the network over alpha. Alpha is dynamic weight which depends on number of parameters of the network like bandwidth, distance etc. We analyzed the at 0.1 of alpha the accepted number of requests are 1102 and bandwidth utilization us 25%. At alpha 0.5 the number of request accepted increased by 368 and becomes 1470 out of 1500. In this case the bandwidth utilization factor is 32%. Blocking probability also enhanced by 12% in our research work using our algorithm for Dedicated Path Protection and Shared Path Protection.

References

[1]. Satish M B, Savitha C, Dr. M Z Kurian, "Implementation of ATM Packets Over MPLS Network on FPGA" International Journal of Computer & Organization Trends ,Vol. 3 Issue 5, June, 2013.
[2]. Rohith Ramkumar, H.A.Chan, "VCAT Differential Delay Minimization for Delay Sensitive Multiservice Networks", 2006.
[3]. Joji Philip, Sailesh Kumar, Sunil Shukla, Raja Venkatesh,, "Architecture for Flow Control and Input Buffering on High Speed Interfaces" Paxonet Communications, CA, USA, 2007
[4]. Jin Y. Yen, "Finding the K Shortest Loopless Paths in a Network", Management Science, Vol. 17, No. 11, Theory Series (Jul., 1971), pp. 712-716.
[5]. Yuchun Guo1, Fernando Kuipers and Piet Van Mieghem, Link-disjoint paths for reliable QoS routing , INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS, Int. J. Commun. Syst. 2003; 16:779–798 (DOI: 10.1002/dac.612)
[6]. Bill Lin, Gjalt de Jong, Carl Verdonck, Sven Wuytack, Francky Catthoor, "Background Memory Management for Dynamic Data Structure Intensive Processing Systems" IMEC, 1996.
[7]. Keyan Zhu Jing Zhang and Biswanath Mukherjee, "Inverse Multiplexing in Optical Transport Networks with the Support of SONET/SDH Virtual Concatenation", IEEE, 2004.
[8]. Kuan Chou Loh, "SIMULATION AND PERFORMANCE ANALYSIS OF ROUTING IN SONET/SDH", DATA COMMUNICATIONS NETWORK (DCN), IEEE. Dec. 2006
[9]. Kuan Chou Loh, "Understanding Virtual Concatenation and Link Capacity Adjustment Scheme in SONET/SDH" ,Thesis NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA ISSN, 2012.
[10]. Madanagopal Ramachandran, N. Usha Rani, and Timothy A. Gonsalves, "Path Computation Algorithms for Dynamic Service Provisioning With Protection and Inverse Multiplexing in SDH/SONET Networks", IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 18, NO. 5, OCTOBER 2010
[11]. "Types and characteristics of SDH network protection architectures," ITU-T, Recommendation G.841, 1998.
[12]. A. E. Ozdaglar and D. P. Bertsekas, "Routing and wavelength assignment in optical networks," IEEE/ACM Trans. Netw., vol. 11, no. 2, pp. 259–272, Apr. 2003.
[13]. H. Zang, C. S. Ou, and B. Mukherjee, "Path-protection routing and wavelength assignment (RWA) in WDM mesh networks under ductlayer constraints," IEEE/ACMTrans. Netw., vol. 11, no. 2, pp. 248–258, Apr. 2003.
[14]. X. Chu, B. Li, and Z. Zhang, "A dynamic RWA algorithm in a wavelength- routed all-optical network withwavelength converters," in Proc.IEEE INFOCOM, Apr. 2003, pp. 1795–1804.
[15]. M. Alanyali and E. Ayanoglu, "Provisioning algorithms for WDM optical networks," IEEE/ACM Trans. Netw., vol. 7, no. 5, pp. 767–778, Oct. 1999.
[16]. S. Janardhanan, A. Mahanti, D. Saha, and S. K. Sadhukhan, "A routing and wavelength assignment (RWA) technique to minimize the number of SONET ADMs inWDM rings," in Proc. 39th HICSS, Jan. 2006, pp. 1–10.
[17]. G. Shen and W. D. Grover, "Performance of protected working capacity envelopes based on p-cycles: Fast, simple, and scalable dynamic service provisioning of survivable services," Proc. SPIE, vol. 5626, pp. 519–533, Feb. 2005.
[18]. M. Kodialam and T. V. Lakshman, "Dynamic routing of bandwidth guaranteed tunnels with restoration," in Proc. IEEE INFOCOM, Mar. 2000, pp. 902–911.
[19]. J. Q. Hu, "Diverse routing in optical mesh networks," IEEE Trans.Commun., vol. 51, no. 3, pp. 489–494, Mar. 2003.
[20]. A. Todimala and B. Ramamurthy, "IMSH: An iterative heuristic for SRLG diverse routing inWDMmesh networks," in Proc. 13th ICCCN, Oct. 2004, pp. 199–204.
[21]. A. Todimala and B. Ramamurthy, "A heuristic with bounded guaranteed to compute diverse paths under shared protection in WDM mesh networks," in Proc. IEEE GLOBECOM, Nov. 2005, pp. 1915–1919.
[22]. Deepak Dhadwal, Ashok Arora, and V R Singh, "Enhancement and implementation of Dedicated Path Protection for SONET/SDH Network", International Journal of Engineering Trends and Technology (IJETT) – Volume 14 Number 3 – Aug 2014.
[23]. Deepak Dhadwal, Ashok Arora, and V R Singh,"SONET/SDH: Review of Technology and Developments", International Journal of Innovative Science and Modern Engineering (IJISME) ISSN: 2319-6386, Volume-2, Issue-7, June 2014.
[24]. Deepak Dhadwal, Ashok Arora ,VR Singh," Optimization of Shared Path Protection for SONET/SDH Network", International Journal of Scientific & Engineering Research, Volume 5, Issue 7, July-2014.