Enhancement and Implementation of Dedicated Path Protection for SONET/SDH Network

Deepak Dhadwal#1, Ashok Arora*2,V R Singh#3
1Assistant Professor,MM University.*2 Executive Dean,Manav Rachna International University.#3Director,PDM College Of Engineering

Abstract **– Synchronous Digital Hierarchy (SDH) and Synchronous Optical Network (SONET) are time-division multiplexing technologies which are used to provide the Bandwidth Services for the end user. Dynamic Service provisioning requires the use of online algorithms. These kind of algorithms compute the efficient paths and proceed data flow in it. The allocation of the path depends on the service requests and bandwidth demand. SONET/SDH networks multiplexing techniques are used for this purpose to allocate the medium for desired Bandwidth. In this paper we have applied an engine for the use of application and use of dedicated path protection for the SONET/SDH network. In this paper, we have investigated and implemented the dedicated path protection scheme and analyzed the performance matrices like blocking probability, Quality of services etc. The performance is evaluated for different values for the relative weight of dedicated path for the complete and minimum information scenarios.**

Index Terms **- Dedicated protection, SONET/SDH, NP-completeness, Shared Risk Link Group (SRLG), Link, Trail.**

I. Introduction

Bandwidth Requirement is one of the critical issue in the current environment of high speed communications. This also effects for the network operators and service providers in terms of communication business. That is the reason which forces the Engineers to generated different types of services like PDH, SONET/SDH, WDM, DWDM, ATM, DSL etc. There is also need of providing different QoS for different users. This directly affects the revenue of Service providers.

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.

The provisioning is handled manually by the service providers. The Service providers manually log into the network. Arrange the activities and parameters of the network like required QoS, available network resources, and the ability to accommodate future requests before starting the work of provisioning a requested service. For this purpose the service provides has to log in to the Management systems and issue the command for the same.

There is one of the main point which is needed to take into consideration is the availability of the Bandwidth services. This is an important area of research to find the maximum availability of Bandwidth in a dense network. This implies manual path computation and maintenance of huge inventory details. Then, the configuration on the network elements involved has to be done. These processes, being manual, are tedious and more error-prone.

One of the solutions for this problem is to have an automatic provisioning system. For such kind of system it is important to have inventory details from the database. For this Element Management Systems, or Network Elements directly should be accessible and it should be able to issue the configuration commands so that the requested capacity can handled easily. Another important task is to collect the information for the working path, links, nodes, node-disjoint path. This information can provide availability of working path, used path, free capacity etc.

In SONET/SDH network, there is need to establish working and protection path for reliable operation. Dedicated and Shared Path protections are the two techniques used for efficient use of Bandwidth in Network. 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 [34]. 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].

In Synchronous Digital Hierarchy (SDH), protection against the failure is more complex. This is because it is used by many service providers to have main backbone traffic and also provide bandwidth services [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 this paper, dedicated path protection is analyzed, implemented and enhanced for a dense network. A working engine has been created using the Matlab tool and analysis has been done for multiple requests. The engine is able to hold the network for the static analysis and can provide better and enhanced way for the dedicated protection mechanism. The Engine is also able to handle Multiplexing Hierarchy.

The paper is organized such that the previous work has been shown in Section II. Section III explain the Dedicated Shared protection conventional Algorithm and their analysis. Section IV shows the implementation of the above discussed Engine, Results analysis and performance analysis has been shown in Section V and finally Section VI has the conclusion of the whole paper.

II. Related Work

Path Computation problem has been addressed by many researchers in previous works in various networks. One of them is related with the light path provisioning. In this issue, the light path is needed to route to the light path requests and wavelengths are assigned such that some of the performance metrics can be enhanced. This is a very common Routing and Wavelength assignment (RWA) problem [6]–[8].

Numbers of heuristics have been implemented with some assumption [9]. To minimize the number of RWA problem in SONET/SDH network have been implemented in [10]. Shared Backup Path Protection (SBPP) [11] have been implemented for the efficiently configuring the Transport network. A set of diverse protection paths arrangement and keeping spare links for the backup paths unconnected and shared with other setups with connection occurs on failure.

The Protected Working Capacity Envelope (PWCE) [11] provide adaptation of static design models to create not an exact solution for one single demand matrix, but an envelope of protected working channels, well suited for a large family of random demand instances that may be somehow related to a single representative demand pattern.

In [12], the problem of dynamically routing QoS guaranteed with restoration in MPLS network is studied. Successfully setting up the route between a path set up requested nodes and reserves an alternative link for backup paths is known as Restorability. ILP formulation helps in this problem and it shows that partial information for the declaration of route between the nodes is more efficient and effective in spite of the network aggregated information. The problem of finding diversely routed paths with SRLG constraint is proved to be NP-Complete in [13].

To compute SRLG diverse paths for dedicated protection using an iterative heuristic is proposed in [14]. Heuristics for computing SRLG/link-diverse paths under shared protection are proposed in [15], [16]. A failure-dependent SRLG protection scheme where the working path is partitioned into overlapping segments and each individual segment protected using a backup is proposed in [17]. The static provisioning problem incorporating realistic constraints for dedicated, shared, and unprotected SRLG-diverse path protection is considered in [18]. A sequential algorithm and a parallel algorithm for computing disjoint paths for protection and QoS for next-generation SONET networks have been proposed in [19], where the shortest path is computed based on the delay in the links and the bandwidth requested.

For independent costs of working and protection paths an optimal path pair is computed my minimizing the sum of costs. This is known as Independent Cost Structure (ICS). An existing protection bandwidth can be used by a new request at no extra cost to the working paths if they are disjoint. The cost of protection path depends on working path and disjointedness. This concept is known as Dependent Cost Structure (DCS) and heuristics for computing paths as described in [20].

Survivable traffic grooming for SONET/SDH problem is handled in the [21] for mesh networks employing WDM under the shared protection. Wavelength usage, grooming port usage, and available light path capacity issues are proposed in this paper [21].

Design of survivable networks by aggregating traffic at path level is studied in [22], where shared, mixed protection algorithms for guaranteed survival from single link failures are developed. Virtual concatenation [3] is an IM technique that groups an arbitrary number of SONET or SDH containers, which may not be contiguous; to create a bigger container called Virtual Concatenation Group (VCG). Multiple members of the same and/or different VCG can share protection resources if they are SRLG-disjoint. Two approaches for provisioning of survivable Data over SONET/SDH (DoS) connections utilizing the IM capability of virtual concatenation have been proposed in [23].

III. Dedicated Protection

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]. In [19] a simple algorithm has been proposed in which shortest path algorithm has been used for finding the shortest path algorithm store this path and then removes this path temporarily and also removes its associated parameters from network then again run the shortest path for the dedicated path protection. This is Two step Heuristic approach employed in this paper. This approach has the limitation that it may not find a path pair even if it exists or it may not be optimal. 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

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.,$P1 \cap Pd1 = \varphi$, $P1 \neq Pd1$ and $P1 \cap Pd2 = \varphi$, $P1 \neq Pd1$.

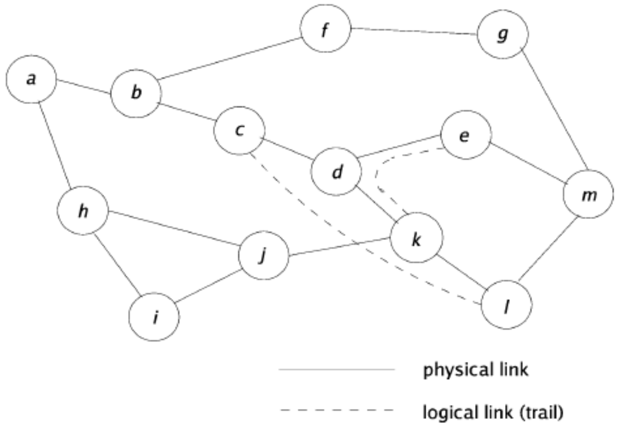

Fig. 1: A sample Network

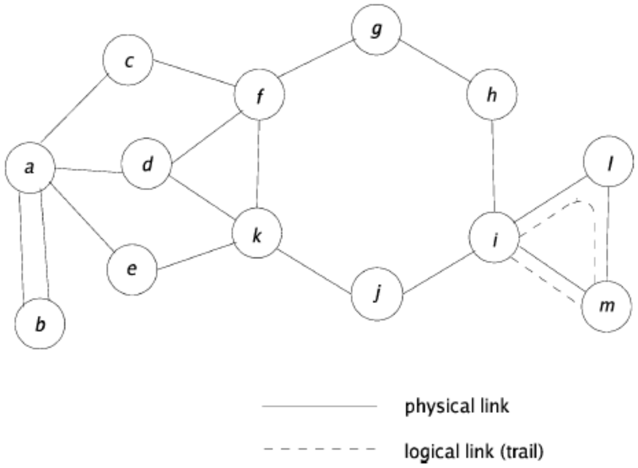

Fig. 2: A sample Network with Standard SDH protection Scheme

Algorithm for Dedicated Protection is now discussed here. Path computation algorithms for SDH networks under dedicated protection have to take into account the constraints imposed by SDH multiplexing hierarchy and the standard protection mechanisms provided by it [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, 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*.

There are some algorithms which are shown in [34]. These algorithms are implemented in this paper and an enhanced engine is developed for the dedicated path protection. Algorithm 1 is used for dedicated path protection. The algorithm is used for the computation of the two paths for the handling of the service request under dedicated protection scheme. The inputs for this algorithm are source, destination, and standard rate. The outputs are dis joint path pair as output.

The algorithm has iterations to provide the best result. The algorithm runs *i*th iterations, for *i*th iterations it has *i*th shortest path. The shortest path is founded by the Yen's K shortest path algorithm [28].

An improved version of Yen's algorithm is provided in the [29] and [1] is used for the shortest path in protection of path and provisioning. The algorithm provides the disjoint path in each iteration. If the shortest path does not exist than break the loop. If capacity is not available in the first ith shortest path then it switch to the next iteration. The iteration goes on till the dedicated shortest path not found. A protected path is also calculated except primary physical link. This protection path is used as backup path if any failure in primary is encountered.

While computing a disjoint path pair, the direction of the seed path is reversed and weights are made negative [24]. This can be done only if the seed path is the shortest path; else it may result in negative cycles. This is achieved by having a flag, which is true for the first shortest path and false for the rest, indicating when the weights have to be inverted. For the paths to be link-disjoint, all the trails except those in the *i*th shortest path that traverse the links traversed by the *i*th shortest path, and all the links traversed by the *i*th shortest path but not in the th shortest path are removed temporarily. Then, the th shortest path is split into segments that are either part of or not part of one of the standard SDH protection mechanisms. This is done by iteratively going through the trails and checking whether the trail itself or the edges it passes through have standard protection.

If it has no protection, then it is added into a list. If it has protection and the protection instance is the same as that of previous edge or trail, then it is added into a second list. If there is a shift from no protection to protection or protection to no protection, the first or second list, respectively, is the segment that is used in the subsequent steps. Then, the lists are reset as empty, and the procedure repeated until the end of the path. For the segments that are part of some standard SDH protection scheme, its inherent protection path is added to . For the other segments, the Edge-disjoint Path Pair (EPP) algorithm outlined in Algorithm 2, which returns an edge-disjoint path pair, is used with that segment as the working path. Finding inherent protection paths for segments that are part of MSP or SNCP is straightforward since the protection path is fixed in those cases.

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, $invertFlag$)

```
for $(i, j)$ in $P_1$ do
   remove the directed edge $(i, j)$
   if $invertFlag =$ true then
      $C(j, i) \leftarrow -C(j, i)$
   else
      $C(j, i) \leftarrow 0$
$P_2 =$ ShortestPath (src,dst,rate)
add the removed edges back to the graph and revert back
to the original weights for those edges for which the
weights were changed
if $P_2 = NULL$ then
   return $NULL$
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 $Q$
return $(P, Q)$
```

C. Algorithm to Find Edge-Disjoint Path Pair (EPP)[34]:

The EPP algorithm is outlined in Algorithm 2. It takes as input the source, destination, working path, standard rate, and gives the edge-disjoint path pair as the output. The term "edge" used in the algorithm refers to both link and trail. This algorithm is very similar to [24]. It uses the working path passed to it to find a disjoint path pair. It removes the edges in the forward direction and either inverts the weight or sets the weight to zero based on the for the edges in the reverse direction. Since this algorithm is called from Algorithm 1, the value that is passed depends on whether it is the first shortest path or not as stated in the earlier subsection.

IV. IMPLEMENTATION

Following are the motive of the implementation of dedicated path protection algorithm:

1) To improve bandwidth consumption as per requests.

2) Allows multiple requests to be generated to decrease wastage as in transport networks which have fixed leased lines and hence wastage of available resources.

There are some of the prerequisites which are as follows to develop the dedicated path protection.

A. Real time Network

Request or Calls originated and completed follows a random fashion, which cannot or hard to predict before it actually occurs. Thus in the given situation, there is a compulsory need for a Real Time Network Engine, which could process the calls/request as per the present time and conditions. These *conditions* are discussed in upcoming sections.

Here we are studying the behavior of SONET/SDH in a Standard Network which uses **50 nodes (users) & 175 links (logical)** connections. A **Network Operating Engine** is there by implemented to process all call requests on real time basis. It can *allow* or *reject* or could keep in *queue* the coming requests at an instance.

B. Request generating system

Request Generating System, created by us, is a key to initiate any request. This system helps in INPUTTING the call requests into the Engine. For simplification purpose of request generation, we kept certain parameters constant and predefined them accordingly. However, on alpha basis, it allows one to generate a request from any one node to any other node within the network limits. You can select the request Source, Destination and the Relative weight (Alpha) which can only be varied within permissible ranges.

C. Analytic System

Analytics is one of the multi-dimensional discipline. The insights from data are used to recommend action or to guide decision making. Thus, analytics is not so much concerned with individual analyses or analysis steps, but with the entire methodology. There is a pronounced tendency to use the term analytics in business settings e.g. text analytics vs. the more generic text mining to emphasize this broader perspective. There is an increasing use of the term advanced analytics, typically used to describe the technical aspects of analytics, especially predictive modeling, machine learning techniques, and neural networks.

Analytic system helps us in judging current situation of the network. It works on many factors like: QOS, accepted & rejected no. Of requests, NOR generated vs alpha, nor rejected vs alpha, QoS vs time, NOR generated vs time, nor rejected vs time, blocking probability vs time (blocking probability Is defined in terms of offered load and call arrival time), (NOR is number of requests).

D. Alpha (α)

Alpha is the relative weight assigned for a Virtual channel (VC) allotted for a particular voice or data communication. This value gives the amount of bandwidth to be used for creating a mutually independent channel within the SONET fiber.

E. Blocking Probability

Blocking probability [32] that describes the probability of call losses for a group of identical parallel resources (telephone lines, circuits, traffic channels, or equivalent), sometimes referred to as an M/M/c/c queue. It is, for example, used to dimension a network's links. It describes a probability in a queuing system (albeit a special case with a number of servers but no queuing space for incoming calls to wait for a free server). Hence, the formula is also used in certain inventory systems with lost sales.

The formula applies under the condition that an unsuccessful call, because the line is busy, is not queued or retried, but instead really vanishes forever. It is assumed that call attempts arrive following a Poisson process, so call arrival instants are independent. Further, it is assumed that the message lengths (holding times) are exponentially distributed (Markovian system), although the formula turns out to apply under general holding time distributions.

The Erlang B formula assumes an infinite population of sources (such as telephone subscribers), which jointly offer traffic to N servers (such as telephone lines). The rate expressing the frequency at that new calls arrive, λ, (birth rate, traffic intensity, etc.) is constant, and does *not* depend on the number of active sources. The total number of sources is assumed to be infinite. The Erlang B formula calculates the blocking probability of a buffer-less loss system, where a request that is not served immediately is aborted, causing that no requests become queued. Blocking occurs when a new request arrives at a time where all available servers are currently busy. The formula also assumes that blocked traffic is cleared and does not return.

The formula provides the GoS (grade of service) which is the probability P_b that a new call arriving to the resources group is rejected because all resources (servers, lines, circuits) are busy: $B(E, m)$ where E is the total offered traffic in Erlang, offered to m identical parallel resources (servers, communication channels, traffic lanes).

$$P_b = B(E,m) = \frac{\frac{E^m}{m!}}{\sum_{i=0}^{m} \frac{E^i}{i!}} \qquad \text{...1}$$

where:

- P_b is the probability of blocking
- m is the number of identical parallel resources such as servers, telephone lines, etc.
- $E = \lambda h$ is the normalised ingress load (offered traffic stated in Erlang).

F. Quality of Service

Quality of service (**QoS**) [33] is the overall performance of a network, particularly the performance seen by the users of the network.

To quantitatively measure quality of service, several related aspects of the network service are often considered, such as error rates, bandwidth, throughput, transmission delay, availability, jitter, etc.

Quality of service is particularly important for the transport of traffic with special requirements. In particular, much technology has been developed to allow computer networks to become as useful as telephone networks for audio conversations, as well as supporting new applications with even stricter service demands.

Quality of service comprises requirements on all the aspects of a connection, such as service response time, loss, signal-to-noise ratio, crosstalk, echo, interrupts, frequency response, loudness levels, and so on. A subset of telephony QoS is grade of service (GoS) requirements, which comprises aspects of a connection relating to capacity and coverage of a network, for example guaranteed maximum blocking probability and outage probability.

In the field of computer networking and other packet-switched telecommunication networks, the traffic engineering term refers to resource reservation control mechanisms rather than the achieved service quality. Quality of service is the ability to provide different priority to different applications, users, or data flows, or to guarantee a certain level of performance to a data flow. For example, a required bit rate, delay, jitter, packet dropping probability and/or bit error rate may be guaranteed. Quality of service guarantees are important if the network capacity is insufficient, especially for real-time streaming multimedia applications such as voice over IP, online games and IP-TV, since these often require fixed bit rate and are delay sensitive, and in networks where the capacity is a limited resource, for example in cellular data communication.

A network or protocol that supports QoS may agree on a traffic contract with the application software and reserve capacity in the network nodes, for example during a session establishment phase. During the session it may monitor the achieved level of performance, for example the data rate and delay, and dynamically control scheduling priorities in the network nodes. It may release the reserved capacity during a tear down phase.

A best-effort network or service does not support quality of service. An alternative to complex QoS control mechanisms is to provide high quality communication over a best-effort network by over-provisioning the capacity so that it is sufficient for the expected peak traffic load. The resulting absence of network congestion eliminates the need for QoS mechanisms.

QoS= (No. of Dropped calls)/ (Total No. of Calls); in a period of time

G. Applied platform:

The MATLAB is used for the application of the network and request generation. The output is analyzed with the help of MATLAB tool. The engine is made for the above said issues and a real time network virtually establish to check the real time issues like blocking probability and QoS. The engine is able to generate number of requests and able to generate the shortest path following the rules of the Dedicated path protection. The Dedicated Path protection algorithm are implemented and measured on different strength of network. The networks are built by the engine and then different kind of the analysis on the base of the dynamic and static network conditions.

There are the following issues which are covered by the engine and these are as follows:

1. Simulation consist of Two distinctive interactive Units:
   a. Network Play
   b. Request Play

2. Network Play

   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

3. Request Play

   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

The Simulations are based on the above points. Network interface provides the interface with the Network which should be specially SONET/SDH network. This network engine runs at back end. It provides the virtual view of the network and we can change the service requests and their bandwidth requirement at our own choice. This help to better analysis on the network. In this engine we can assign number of the nodes and their links with weights automatically or manually. We can stop the running network for the analysis purpose and again resume the network. The states of the network can be check at any time. That how many number of requests handled or rejected by the network at any particular time. This can be check using the state viewer of the Network Engine.

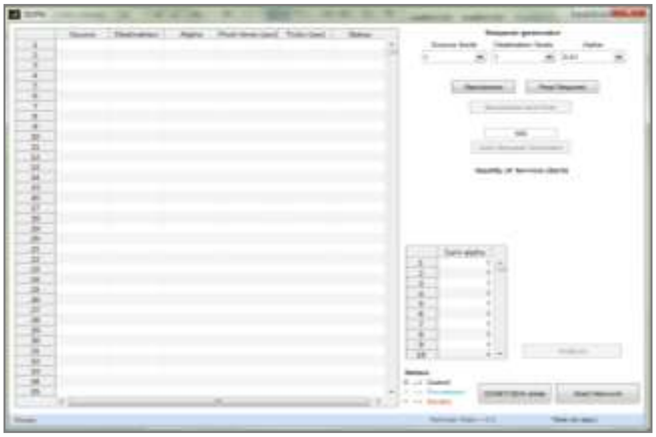

Fig. 3: MATLAB Network Engine for Dedicated path protection

Figure 3, shows a view for the working platform to apply the Network issues for the Dedicated Path protection and provisioning. At backend the network is created with automatic weight assignment which is based on the bandwidth and distances from the node and applied through dedicated path protection. The distance between the nodes also plays important role for the weight assignment and are implemented in the algorithm for dedicated path protection.

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 effects the Network.

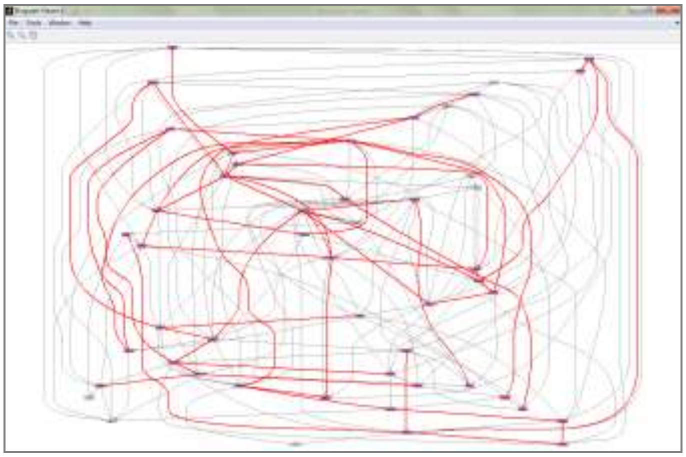

Fig. 4: A simple Network with connected and unconnected nodes using dedicated path protection

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.

IV. Results and Performance analysis

This requires section-wise data gathering and computation as well as relevance to the overall goal perspective. Our Analytic System computes the data and provide us in a raw form, which needs to be understand and then implement to for higher gains.

Our goal is to minimize the no. of rejected/dropped call in a particular network setup. This requires the enforcement of our Bandwidth utilization strategy and engine for predictive calculations and harness of any unstable situations.

A. Data Computation

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

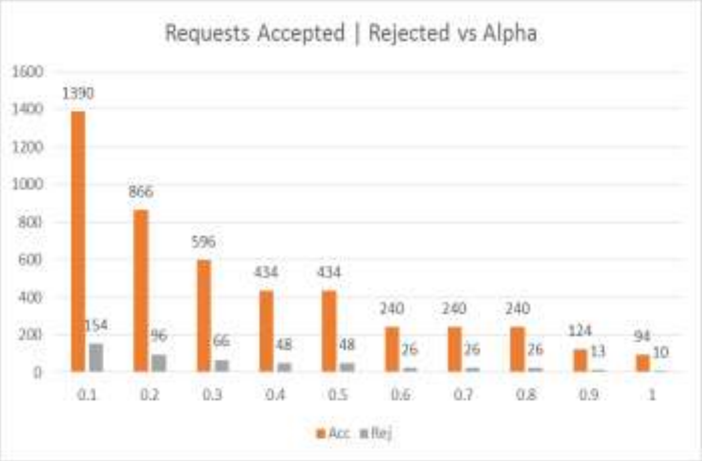

Fig. 5: Accepted and Rejected Requests Vs Alpha

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

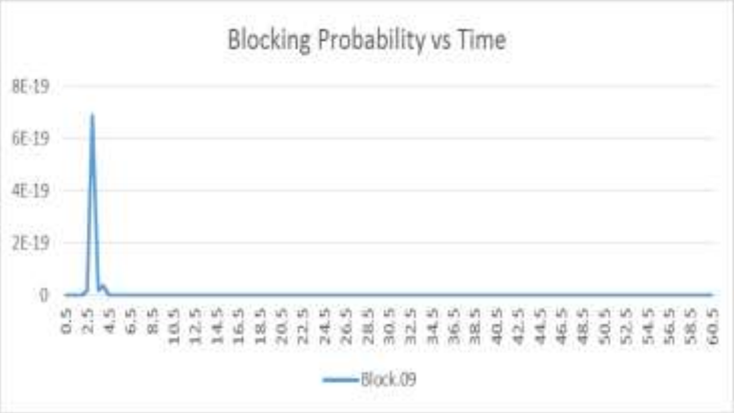

Fig. 6: Blocking Prob. Vs Time

3. *QoS vs Time*

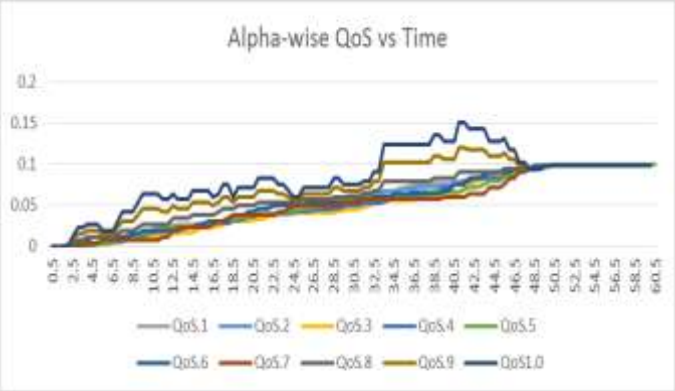

Fig. 7: QoS Vs Time

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 has a clear dependency on Alpha values:** Alpha (α) values range between 0.01 and 1.00 with a precision up to 2 decimal points. Due to this high accuracy rate in setting up the exactly demanded Alpha value, we can adjust the system as per the increasing day-by-day call-flow rates.

2. **QoS decides the Ideal Alpha:** As we look closely to the ***QoS vs Time*** graph we finds some really interesting, yet ultimate system manipulation criteria's. This graph shows the Functioning of the NETWORK in the TEST Condition set-up for about 1 Hour with a *predefined* Call-flow rate. This test condition has been repeated for 10 different equally-spaced Alpha levels.

B. Analysing/Inferences

- 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.

V. Conclusion

If the SONET/SDH network is implemented using the above inferences as the criteria for its operation, then the functional capacity of the network increases by 50% (as compared with the ***non-bandwidth-optimized*** networks; in the testing environment). This implies that, it will give 50% more acceptance rate for all the incoming/outgoing voice and data channels. This percentage might be affected by the hardware quality, routers in remote area locations, demographics, topography, power management units, and even by other environmental effects.

References

[1] R. Madanagopal, N. U. Rani, and T. A. Gonsalves, "Path computation algorithms for dynamic service provisioning in SDH networks," in *Proc. 10th IFIP/IEEE IM*, May 2007, pp. 206–215.

[2] W. Fawaz, B. Daheb, O. Audouin, B. Berde, M. Vigoureux, M. Du-Pond, and G. Pujolle, "Service level agreement and provisioning in optical networks," *IEEE Commun. Mag.*, vol. 42, no. 1, pp. 36–43, Jan. 2004.

[3] "Network node interface for the synchronous digital hierarchy (SDH)," ITU-T, Recommendation G.707/Y.1322, 2000.

[4] "Physical/electrical characteristics of hierarchical digital interfaces," ITU-T, Recommendation G.703, 2001.

[5] "Types and characteristics of SDH network protection architectures," ITU-T, Recommendation G.841, 1998.

[6] 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.

[7] 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.

[8] 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.

[9] M. Alanyali and E. Ayanoglu, "Provisioning algorithms for WDM optical networks," *IEEE/ACM Trans. Netw.*, vol. 7, no. 5, pp. 767–778, Oct. 1999.

[10] 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.

[11] 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.

[12] M. Kodialam and T. V. Lakshman, “Dynamic routing of bandwidth guaranteed tunnels with restoration,” in *Proc. IEEE INFOCOM*, Mar. 2000, pp. 902–911.

[13] J. Q. Hu, “Diverse routing in optical mesh networks,” *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 489–494, Mar. 2003.

[14] A. Todimala and B. Ramamurthy, “IMSH: An iterative heuristic for SRLG diverse routing inWDMmesh networks,” in *Proc. 13th ICCCN*, Oct. 2004, pp. 199–204.

[15] 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.

[16] P.-H. Ho, J. Tapolcai, and H. T. Mouftah, “Diverse routing for shared protection in survivable optical networks,” in *Proc. IEEE GLOBECOM*, Dec. 2003, pp. 2519–2523.

[17] D. Xu, Y. Xiong, and C. Qiao, “A new PROMISE algorithm in networks with shared risk link groups,” in *Proc. IEEE GLOBECOM*, Dec. 2003, pp. 2536–2540.

[18] L. Shen, X. Yang, and B. Ramamurthy, “Shared risk link group (SRLG)-diverse path provisioning under hybrid service level agreements in wavelength-routed optical mesh networks,” *IEEE/ACM Trans. Netw.*, vol. 13, no. 4, pp. 918–931, Aug. 2005.

[19] N. Ansari, G. Cheng, S. Israel, Y. Luo, J. Ma, and L. Zhu, “QoS provision with path protection for next generation SONET,” in *Proc. IEEE ICC*, Apr. 2002, pp. 2152–2156.

[20] A. Todimala, B. Ramamurthy, and N. V. Vinodchandran, “On computing disjoint paths with dependent cost structure in optical networks,” in *Proc. 2nd BROADNETS*, Oct. 2005, pp. 155–164.

[21] C. S. Ou, K. Zhu, H. Zang, L. H. Sahasrabuddhe, and B. Mukherjee, “Traffic grooming for survivableWDMnetworks—shared protection,” *IEEE J. Sel. Areas Commun.*, vol. 21, no. 9, pp. 1367–1383, Nov. 2003.

[22] S. Balasubramanian and A. K. Somani, “Dynamic survivable network design for path level traffic grooming in WDM optical networks,” in *Proc. IEEE GLOBECOM*, Nov. 2007, pp. 2359–2363.

[23] C. S. Ou, L. H. Sahasrabuddhe, K. Zhu, C. U. Martel, and B. Mukherjee, “Survivable virtual concatenation for data over SONET/SDH in optical transport networks,” *IEEE/ACM Trans. Netw.*, vol. 14, no. 1, pp. 218–231, Feb. 2006.

[24] J. W. Suurballe and R. E. Tarjan, “A quick method for finding shortest pairs of disjoint paths,” *Networks*, vol. 14, no. 2, pp. 325–336, 1984.

[25] Y. Guo, F. Kuipers, and P. V. Mieghem, “Link-disjoint paths for reliable QoS routing,” *Int. J. Commun. Syst.*, vol. 16, no. 9, pp. 779–798, Nov. 2003.

[26] R. Bhandari, “Optimal diverse routing in telecommunication fiber networks,” in *Proc. IEEE INFOCOM*, Jun. 1994, pp. 1498–1508.

[27] S. Z. Shaikh, “Span-disjoint paths for physical diversity in networks,” in *Proc. IEEE Symp. Comput. Commun.*, Jun. 1995, pp. 127–133.

[28] J. Y. Yen, “Finding the K shortest loopless paths in a network,” *Manag Sci.*, vol. 17, no. 11, pp. 712–716, Jul. 1971.

[29] E. Q. V. Martins and M. M. B. Pascoal, “A new implementation of Yen’s ranking loopless paths algorithm,” *4OR, Quarter. J. Oper. Res.*, vol. 1, no. 2, pp. 121–133, Jun. 2003.

[30] “70 node network diagram (Network 1),” [Online]. Available: http://www.lantana.tenet.res.in/website_files/projects/NMS/nwdiagram1. Jpg

[31] “VSNL network diagram (Network 3),” [Online]. Available: http://www.lantana.tenet.res.in/website_files/projects/NMS/nwdiagram3. Jpg

[32] Available: http://en.wikipedia.org/wiki/Erlang_%28unit%29

[33] Available: http://en.wikipedia.org/wiki/Quality_of_service

[34] 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, pp: 1492–1054.