Modularity and Economy-of-Scale Effects in the Optimal Design of Mesh-Restorable Networks

John Doucette$^{1,2}$, Wayne D. Grover$^{1,2}$, Randy Martens$^1$
$^1$TRLabs, #800 10611 - 98 Avenue, Edmonton, Alberta, Canada T5K 2P7
$^2$Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta
Contact: {jdoucett, grover, rmartens}@edm.trlabs.ca

Abstract – Most, if not all, theoretical work on the capacity design of mesh-restorable networks has been done in an integer, but non-modular, fashion. Although OC-n modules can only approximate the exact design capacities, this has been acceptable in practice where networks are rapidly growing and demand forecasts have uncertainty in any case, or in research studies where modularization only confuses the comparison of underlying theoretical effects. However, the modularity and economy-of-scale effects in near-term dense wave division multiplexing (DWDM) transmission systems may be so large (consider 40 λ at OC-192 on a single fiber) that we need to study and exploit these effects in the optimal capacity design problem. To do so, we introduce and compare results with two approaches for modular optimal capacity design to non-modular reference designs which are subject to conventional post-modularization. Significant shifts in the network structure, and total costs savings up to 20% are seen in the test results.

I. INTRODUCTION

Today society, commerce, and nations are highly dependent on the reliable and continuous functioning of our telecommunications infrastructure. Debit card transactions, credit card purchases, computer data networks, financial institutions, or 911 calls can all be disrupted by a single fiber break unless some sort of restoration mechanism is in place to immediately reroute signals around the failure. An outage of a few minutes or even several seconds can easily cost a telco and its customers millions of dollars, and may result in even more serious consequences if for example an emergency system is rendered inoperable. Even a short disruption in the physical layer may cause routers in the IP layer to issue thousands of link-state updates, possibly entering a ‘routing flap’ as it is called in the Internet. A key part of stabilizing the Internet and assuring confidence in its performance as an infrastructure for e-commerce will be to effect almost un-interruptible SONET / WDM physical layer transport.

In the past decade, two distinct restoration methods have emerged for ensuring such survivability. These are the ring- and mesh-based network restoration concepts.

Ring-based restoration relies on predetermined switching mechanisms within closed ring-like sub-networks. These rings function such that in the event of failure of a part of the ring between two nodes on the ring, traffic is very quickly rerouted in the other direction around the ring on spare capacity built into the ring. The rings can work on either the bi-directional line-switched (BLSR) or unidirectional path-switched (UPSR) ring principle. Sets of rings are then developed either to cover all physical fiber spans of the network graph, or a routing of demands is found so as to transit end-to-end through a given set of transport rings. More on ring-based networks can be found in [1] and in the companion paper [2] at this conference.

In mesh-restorable transport networks, the idea is that a much more generalized and efficient path-finding mechanism will restore failed signal units by multi-path rerouting through relatively much smaller allocations of spare capacity, distributed throughout the network. The main advantage of mesh-based restoration is that far less total spare capacity is required for full survivability, mainly due to the spare capacity being shared by many different possible failure scenarios. The problem of deciding how to route working demands and correspondingly place spare capacity so that 100% restoration is feasible with a minimum of total capacity (or total spare capacity) is called the mesh capacity design problem. That is the topic of this paper.

Mesh restoration schemes are subdivided into two main categories. In span restoration, failed signal units are restored by substituting a logical detour or restoration path, created out of spare capacity units or ‘links’, from one immediate endpoint of the failed span to the other node adjacent to the break. From here, working signals continue their original routing. This is also variously called line, patch, or local restoration. In path restoration all signals affected by a span cut are effectively re-provisioned end-to-end between their individual origin
destination (O-D) node pairs. Rerouting is thus viewed as a global network reconfiguration problem. Path restoration is accordingly more capacity-efficient than span restoration, but also considerably more complex in both real-time routing requirements and capacity design. Recent work on path-restorable networks can be found in [3,4,5]. For purposes of this paper, however, our interest is in span restoration.

Grover et al. [6] gave one of the first heuristics for span-restorable mesh spare capacity planning. This was an iterative hill-climbing approach in which the greatest increments in network restorability are sought as spare capacity units are added one-at-a-time until the target restorability level is achieved. A secondary ‘tightening’ phase would improve the overall design. Modularity was handled to an extent in this method by defining the basic building unit as an OC-n module. The method is not optimal however and tightening phase could take a long time. Moreover, only a single modularity could be handled.

Herzberg and Bye [7] later gave an integer programming (IP) formulation to find the strictly optimal (non-modular) spare capacity placement, given working capacities already assigned. Herzberg’s key to solving the formulation was to use LP relaxations, to accept hop-limits which restrict the maximum length of restoration routes, and to pre-process the network graph to find all distinct “eligible” restoration routes, up to the hop limit, for each prospective failure. Spare capacity placement is then formulated as a problem of optimal assignment of restoration flows to the eligible routes. The aspects of hop limits and flow assignment are inherited by the formulations here.

Both [6,7] first route working paths (usually through shortest path routing) and then place the spare capacity to restore failure of spans carrying working paths. In other words, spare capacity is optimized to suit the working capacity. They are not jointly planned. Obviously, however, the amount of working capacity on a span will affect the amount of spare capacity needed elsewhere in the network to effect its restoration, so manipulating the working path routing could provide a lower total capacity requirement. Iraschko et al. [3] give an IP formulation for joint working and spare capacity placement (for span and path restoration) which may route working paths other than in a shortest path manner so that total capacity is minimized, not merely spare capacity. A more extensive review of work on this problem is available in [8].

All of these prior works, however, solve for capacity variables in an integer but non-modular way. For example, 17 units of total capacity may be called for on a span, even though SONET (modulo-12) capacities must be used, so that in this case OC-24 would be required, implying 7 units of capacity more than was needed on that span. This has of course long been recognized but, given other research priorities, modularity issues have been generally given only passing mention. Post-modularization of unit-capacity designs has been the norm. With the arrival of wavelength division multiplexing (WDM), capacity module sizes may, however, become much larger than with SONET alone. In addition we may see very significant economy-of-scale effects, e.g., where a doubling of module capacity cost significantly less than twice as much. Indeed in today’s SONET community a typical ratio is that one obtains “four times the capacity for only 2 – 3 times the price”. If capacity placement is designed to exploit modularity, this economy-of-scale effect can also be exploited. Thus, primarily due to advent of DWDM technology, we think it is timely to study the differences in cost and architecture between post-modularized unit-bandwidth designs and truly modular optimal designs incorporating non-linear cost-capacity coefficients.

To do so, we have developed integer programming (IP) formulations, which include modularity constraints and minimize the total cost of modules placed. A complicating aspect of the truly modular problem is that working path routing and spare capacity placement really have to be considered jointly. Intuitively, if an OC-192 module costs only twice as much as an OC-48 module, it may be economic to detour the routing of smaller working flows and restoration paths to take advantage of a large relatively low-cost module. Not only should this result in lower costs compared to post-modularization, it may theoretically result in spontaneous elimination of a span if the advantage of detouring paths over larger modules is so great that some spans will not carry any capacity at all.

A. Outline

In Section II. we first re-visit a conventional (Herzberg-like) post-modularized spare capacity placement formulation which is used as a benchmark. We then introduce two modular capacity design methods. The first is a compromise approach that is of interest: it could be called a modular-aware spare capacity solution. It is a compromise in that working paths are still routed independently on shortest paths. The second is a true modular joint optimization. Several network models, which serve as test cases, are provided in Section III. Results are presented and explained in Section IV, and in concluding in Section V.
II. MODULARIZED DESIGN STRATEGIES

A. Post-Modularized SCP Benchmark (PMSCP)

A typical method used for capacity placement in mesh-restorable networks is to first route working paths, and then optimally place spare capacity required to restore the working paths [7] followed by modularizing the resulting span \((s_i+w_i)\) quantities. In our model, working paths are first routed between O-D end node pairs on the shortest path between those two nodes. This generates the \(w_i\) quantities, as inputs to the baseline formulation. The spare capacity quantities, \(s_i\) are then solved with the following formulation. The resulting \((s_i+w_i)\) totals on each span will be subsequently modularized to serve as the post-modularized spare capacity placement (PMSCP) results.

For this and subsequent formulations, we use the notation:

**Parameters (inputs):**
- \(C_j\): Cost of each unit of capacity on span \(j\)
- \(H^2_i\): Maximum length (in number of spans) of any eligible restoration route for failure of span \(i\)
- \(L_i\): Target Restoration level for span \(i\) \((L_i = 1\) assumed\)
- \(S\): Number of spans in the network
- \(P_i\): Number of eligible routes for restoration of span \(i\)
- \(w_j\): Number of working links (capacity units) on span \(j\)
- \(\delta_{i,j}^p\): Equal to 1 if \(p^{th}\) eligible route for span \(i\) uses span \(j\)

**Variables:**
- \(f_i^p\): Restoration flow assigned to \(p^{th}\) route for span \(i\)
- \(s_j\): Number of spare capacity units placed on span \(j\)

**Objective Function:**

\[
\text{IP1:} \quad \text{Minimize}\left\{\sum_{j=1}^{S} C_j \cdot s_j\right\}
\]

Subject to:

2. \(\sum_{p=1}^{P} f_i^p \geq \left[w_i \cdot L_i\right] \quad \forall i = 1,2,\ldots,S\)

3. \((s_j) \geq \left(\sum_{p=1}^{P} \delta_{i,j}^p \cdot f_i^p\right) \quad \forall(i,j) = 1,2,\ldots,S. \quad i \neq j\)

The constraints in Equation 2 ensure that restoration for span failure \(i\) meets the target level. The system of constraints in Equation 3 ensures sufficient spare capacity on every span \(j\) so that the sum of the simultaneously instantiated restoration paths routed over span \(j\) is met for every failure span \(i\). The largest simultaneously imposed set of restoration paths imposed on a span effectively sets the minimum \(s_i\) value on each span in the solution.

Once the link-by-link spare capacity placement is determined, modularity is implemented by rounding the total working and spare capacities up to the smallest module of sufficient size. Results for this post-modularized design method are used as the benchmark against which we compare the next two modular capacity design approaches.

B. Modular Spare Capacity Placement (MSCP)

The first modular capacity formulation continues to route working paths on shortest routes, but then places spare capacity with an inherently modular orientation. This is a compromise of sorts in that capacity decisions are inherently modular, but any advantages due to modularity impacts on working path routing are not captured. The interest is, however, in seeing how much of the benefit of modularity is attained by this intermediate formulation relative to the more complex fully modular joint formulation which follows. We reuse the prior notation plus:

**New Parameters (inputs):**
- \(C_j^m\): Cost of a module, of the \(m^{th}\) size, on span \(j\)
- \(M\): Number of different module capacities
- \(Z^m\): Number of capacity units for the \(m^{th}\) module size

**New Variables:**
- \(n_j^m\): Number of modules of \(m^{th}\) size placed on span \(j\)

Since we are now placing capacity in a modular fashion, our objective function is modified to minimize the total cost of all modules placed on the network:

\[
\text{IP2:} \quad \text{Minimize}\left\{\sum_{n=1}^{M} \sum_{j=1}^{S} C_j^m \cdot n_j^m\right\}
\]

We must also add a new set of constraints to those above:

4. \(s_j + w_j \leq \sum_{m=1}^{M} n_j^m \cdot Z^m \quad \forall j = 1,2,\ldots,S\)

The constraint system assigns a sufficient number of modules of each size so that the total of working and spare capacity required will be met. Note that this does not replace Equation 3, but is required in addition.
C. Modular Joint Capacity Placement (MJCP)

We now look at a no-compromise modular optimal capacity placement model. Here the routing of working paths and the required sparing on other spans for restoration are jointly decided, with the available suite of modularities in mind. This can be considered the most complete and desirable approach to the problem, which fully exploits economy-of-scale and modularity effects. It is however the most computationally complex and generates the largest IP tableaus for the solver. The main additional complexity comes from having now to define an eligible route set for routing each working demand in the pre-failure network, as well as the prior eligible route-sets for restoration of each failure scenario. Our interest is in seeing how much added benefit there would be to a network operator if the added preprocessing and IP solver complexity were accepted, however. Again, we build on the IP1 formulation and introduce new notation as needed:

Parameters (inputs):

- \( D \) Total number of O-D pairs with non-zero demand
- \( d^r \) Number of demand units for O-D pair \( r \)
- \( H^r \) Maximum length (in number of spans) of any eligible working route for demand pair \( r \)
- \( Q^r \) Number of eligible working routes available for demand pair \( r \)
- \( \zeta_{j,r,q} \) Equal to 1 if \( q \)th eligible route for demands between O-D pair \( r \) uses span \( j \)

New Variables:

- \( g^{r,q} \) Working capacity required by \( q \)th eligible route for demand pair \( r \)
- \( w_j \) Number of working capacity units on span \( j \)

Three new constraint systems are needed:

5. \( \sum_{q=1}^{Q^r} g^{r,q} = d^r \quad \forall r = 1,2,..., D. \)

6. \( \left( \sum_{i=1}^{D} \sum_{q=1}^{Q^r} \zeta_{j,r,q} g^{r,q} \right) = w_j \quad \forall j = 1,2,..., S. \)

The first set of added constraints ensures that all working demands are routed. The next generates the logically required working capacity on each span \( j \) to satisfy the sum of all pre-failure demands routed over it.

III. EXPERIMENTAL DESIGN

A. Software

The three formulations were implemented in the AMPL Modeling System 6.0.2 (Bell Laboratories, 1998) and solved using CPLEX Linear Optimizer 6.0 with Mixed Integer & Barriers Solvers (ILOG, 1997-1998). Data files containing all required input data, including the eligible route sets for restoration and working path routing for each network model were generated using custom designed C programs written at TRLabs (ModCap1.exe, ModCap2.exe, and ModCap3.exe).

B. Network Models

The different design methods were tested in four test networks of varying size. JED9807b is a small hypothetical network topology constructed as a simple test case that is suitable for manual verification and checking exercises. The Bellcore1 test case is from Yang and Hasegawa [9]. The Bellcore2 test case is based on another network used in a BellCore publication [10]. The EuroNeta topology is based on that given by Lardies and Aguilar [11] representing a backbone fiber optic route structure between major European cities.

In the absence of demand matrices for most of these networks, we used their topologies and generated demand patterns using what we call Gravity-Based Demand. We begin by assuming that all possible node pairs may exchange some amount of demand. The number of units of demand any node pair exchanges is generated in proportion to the product of the degrees of the two nodes and inversely proportional to the distance between the two nodes. Specifically, the demand is calculated as:

\[
\text{demand}(a,b) = \text{int} \left[ \frac{\text{nodal degree}_a \times \text{nodal degree}_b \times \text{constant}}{\text{distance}} \right]
\]

The constant is set empirically so that the individual demand quantities are realistic in comparison to other data sets we have from industry sources. Additionally, there is an aspect of experimental design involved; we want in some cases to ensure that the largest available module size is indeed fairly large relative the average baseline \( w_j \) quantity. The point being that unless this is true, the hypothesized attraction of working routes towards large strategically placed modules is inhibited. Very similar gravity demand models have been used in other work such as [11,12]. The model tends to reproduce plausible expectations about the real world in that large centres have strong communities of interest and an inherent hubbing tendency is apparent. The resulting test case summary data is provided in Table 1:
Table 1 – Test Network Characteristics

<table>
<thead>
<tr>
<th>Network</th>
<th>Nodes</th>
<th>Spans</th>
<th>Demand Pairs</th>
<th>Total Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>JED9807b</td>
<td>6</td>
<td>14</td>
<td>15</td>
<td>75</td>
</tr>
<tr>
<td>Bellcore1</td>
<td>11</td>
<td>23</td>
<td>5</td>
<td>341</td>
</tr>
<tr>
<td>Bellcore2</td>
<td>15</td>
<td>28</td>
<td>105</td>
<td>465</td>
</tr>
<tr>
<td>EuroNeta</td>
<td>19</td>
<td>37</td>
<td>171</td>
<td>841</td>
</tr>
</tbody>
</table>

Table 2 – Eligible Routes Provided to the Formulations

<table>
<thead>
<tr>
<th>Working Hop-Limit</th>
<th>Eligible Routes</th>
<th>Average Restoration Hop-Limit</th>
<th>Eligible Restoration Routes</th>
</tr>
</thead>
<tbody>
<tr>
<td>JED9807b</td>
<td>3.0</td>
<td>210</td>
<td>3.0</td>
</tr>
<tr>
<td>Bellcore1</td>
<td>3.5</td>
<td>691</td>
<td>3.6</td>
</tr>
<tr>
<td>Bellcore2</td>
<td>4.1</td>
<td>1045</td>
<td>4.3</td>
</tr>
<tr>
<td>EuroNeta</td>
<td>4.2</td>
<td>2098</td>
<td>4.5</td>
</tr>
</tbody>
</table>

Table 3 - Post-Modularized SCP (benchmark) Results

By comparing the required capacity column to the total modular capacity columns, it is apparent that 14% to 37% of the total capacity on the network is excess due to modularity effects in the benchmark designs. As might be expected it is generally lower for large networks than for small networks. Results also show that the required (logical) capacity using MSCP method increases but this is expected. MSCP allows less direct restoration routing of failed signals to take advantage of large modules elsewhere in the network, so there is an increase in total capacity-hop units required. However, in most cases, total module capacity remained the same or decreased, relative to simple post-modularized designs. And in all cases, there was a reduction in the proportion of unused module capacity. Thus, we are, indeed, seeing the exploitation of modular capacity placement opportunities. Under MSCP only 9% to 30% of the capacity placed on the networks is in excess of design use. Cost savings of up to 6% were also realized with MSCP.

Table 4 – MSCP Design Results

* Sub-optimal solutions for cases indicated.

The greatest improvements in design efficiency and costs are, however, seen with MJCP. Cost savings of between 6% and 21% were realized. Excess capacity was decreased to between 1% and 4%, and most networks showed decreased total modular capacity, as much as 12%, even though the internal logical design capacity continued to rise. This is all consistent with the notion that in MJCP both working and restoration paths are being carefully tucked together into well filled, cost-effectively chosen module sizes.

Two of the four networks also experienced spontaneous span elimination under MJCP. Five spans on JED9807b carried no capacity, as did four spans on Bellcore1. This is a significant research finding as it admits that topology reduction may be cost-effective in a mesh restorable network, despite the general tendency for higher
connectivity to increase the spare capacity sharing efficiency of a mesh network.

<table>
<thead>
<tr>
<th>Network</th>
<th>Req’d Capacity</th>
<th>OC-12 Cost</th>
<th>OC-24 Cost</th>
<th>OC-48 Cost</th>
<th>Module Capacity</th>
<th>% Cost Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>JED9807b</td>
<td>174</td>
<td>3</td>
<td>6</td>
<td>0</td>
<td>330</td>
<td>180</td>
</tr>
<tr>
<td>Bellcore1</td>
<td>883</td>
<td>0</td>
<td>1</td>
<td>18</td>
<td>940</td>
<td>888</td>
</tr>
<tr>
<td>Bellcore2</td>
<td>1436</td>
<td>1</td>
<td>2</td>
<td>29</td>
<td>1560</td>
<td>1452</td>
</tr>
<tr>
<td>EuroNeta</td>
<td>2355</td>
<td>2</td>
<td>5</td>
<td>48</td>
<td>2660</td>
<td>2448</td>
</tr>
</tbody>
</table>

* Sub-optimal solutions for cases indicated.

Table 5 – MJCP Results

Figure 1 graphically summarizes the relative costs of the solutions achieved for each network using the three methods. The bars represent the costs of each particular solution as a proportion of the costs of the solutions using PMSCP. Observe that MSCP gives a clear improvement over PMSCP, while MJCP gives an even greater improvement. As the footnotes on Tables 4, and 5 indicate, these results may as yet not even show the full extent of the benefits obtained as, at the paper deadline, some CPLEX runs were not yet complete. The best feasible solutions found to date are reported.

![Figure 1 - Relative Total Network Capacity Costs](image)

V. CONCLUDING DISCUSSION

This is an interim report on an ongoing project. Further work will investigate the effects of varying the module costs and sizes. The current analyses employ OC-12, OC-24, and OC-48 module sizes with arbitrarily chosen costs of 30, 40, and 50 respectively. Continuing work will explore the effects of OC-96 and OC192 modules and varying cost-capacity nonlinearities that may characterize future transmission systems. The significance of the work so far is that we have shown two formulations to advantageously exploit modularity rather than allowing it to be a post-design noise-like excess capacity effect. Moreover, we have given the first demonstration that one can design span-restorable mesh networks to exploit economy-of-scale effects, and to use a modular design formulation as a tactic to identify spans for elimination from a future network topology.

VI. REFERENCES