Quantifying Optimal Mesh and Ring Design Costs

Thomas Stidsen
Informatics and Mathematical Modelling
Technical University of Denmark
Lyngby, Denmark

Arne John Glenstrup
Research Center COM
Technical University of Denmark
Lyngby, Denmark

During the last decade telecommunication operators have been deploying WDM (Wavelength Division Multiplexing) technology to satisfy the exponential growth in global communication. While facilitating the advanced information society of today, this has also led to a higher dependency on the networks, and furthermore the high capacity utilisation of optical fibres means that a single link failure will influence many users and enterprises. For these reasons, protection of network connections has become a major competitive parameter for the operators.

Currently, the most popular protection method is ring protection, due to its simplicity, requiring only basic management functionality and operating with local restoration control. While many optical rings have been deployed, little work has been published on exactly what the cost of ring networks are, compared to general mesh networks.

In this article we perform a quantitative comparison between ring protection and mesh protection, using real world network data and realistic prices for network components. Extending classic LP flow models to take rings and node costs into account, and using a link-path based mesh network LP model, we are able to perform a total cost comparison of the two architectures, and of manual ring network design.

The results suggest that the price of mesh network components must be reduced significantly to be competitive with ring based networks, and also that manual network design does not necessarily lead to the most cost-efficient designs.

Keywords: Ring network, mesh network, node model, optimisation, routing

1. INTRODUCTION

A tremendous growth of the global communication networks has taken place in the last decade, both in terms of investment and capacity. This growth has made network planning a vital activity for the telecommunication companies. An important part of the network planning is assuring protection of the networks, i.e. that the communication flow can recover quickly from defined error scenarios. Protection against error scenarios is important, because companies and people increasingly depend on the network, and failures are thus more costly. Further, a single cable cut

Naval Research Logistics , Vol. preprint, pp. 1–14(2004)

This is a preprint of an article accepted for publication in Naval Research Logistics

may affect many people because of the large capacity of each fibre connection; hence reliability has become a competitive parameter for the telecommunication companies.

Even though the technological development has been fast in the recent years, standardisation has progressed at a much slower rate, making protection hard and expensive. The main protection methods today are 1+1 protection and ring protection [10]. Both these mechanisms are fairly expensive in the necessary hardware costs, but they have the common advantage over general $1{:}\,n$ path protected mesh networks of having a very fast protection switching time. Furthermore, the management of mesh network path protection has not yet been standardised. Recently ring protection has received most attention, since it is the cheapest approach, given its simplicity and sharing of protection capacity.

A number of articles have been written about the construction of ring networks [2, 4, 7, 8, 11]. In the work by Cosares et. al. the implemented model has been decomposed so that the solution cannot be claimed to be optimal. Morley and Grover give an optimal model of a ring network, but only a fraction of it is solved and hence the optimum remains unknown. Furthermore, the cost metric only concerns the link costs. In the work of Morales et. al. a heuristic for constructing a ring cover of a mesh network is given, and good results are reported; Luss et. al. present another scheme for the same task. All this work, except that of Morley and Grover and Wuttisittikulkij et. al., assumes that the traffic is given and the ring network has to be constructed. Morley and Grover consider the routing of the traffic and the ring construction at the same time—however, they also ignore the costs of the node equipment. Wuttisittikulkij et. al. construct several different heuristics for capacity design of mesh networks and of ring networks, given a mesh network. These heuristics do not take node costs into account, nor do they integrate the selection of paths and rings, but are promising algorithms for optimisation of larger ring networks.

In this work we focus on two aspects:

Quantitative comparison of *real world* data: We consider networks in operation, and perform experiments using real prices for switch node components and fibre links. Key questions include

- What are the differences between networks designed by human network planners and optimal networks designed by optimisation programs?
- How is the cost of the network distributed between link costs and node costs?
- How should the components be priced for protected mesh networks to be competitive with ring networks?

Comparison of IR and PR: We consider two methods for automatic ring design: IR (integrated routing), where path and ring selection are integrated into one optimisation, and PR (prerouting) where paths are routed prior to ring selection. Key questions include

- Quantitative: What are the cost differences?
- Qualitative: What do the solutions look like?

If we want to investigate these questions, we need a more detailed model of the networks than the models presented in the previously mentioned articles, and we need realistic examples to perform experiments.

In this paper we will formulate linear programs which model ring networks and path protected mesh networks; both models take routing as well as protection into account and further use realistic costs regarding both links and nodes. The ring network LP (Linear Programming) model is an extension of the classic LP flow model to include rings, and the mesh network model is a standard link-path LP model.

Since the problems are very hard and the models quite detailed, we should not expect the LP models to be solvable except for small networks. Both problems are inherently MIP (mixed integer linear programs), but we can only solve their relaxation—hence a simple rounding scheme is used to obtain integer solutions.

2. MODELS

We consider a design problem where the traffic demand is static and given in a matrix as a number of VC-4 connections (end-to-end bandwidths), for each node pair. The topology is given as a set of nodes and links, but the problem is *uncapacitated* in the sense that each link can be assigned as many fibres as necessary, and each node as many switch components as necessary. The cost of a link is proportional to the capacity used, so we do not impose an initial cost whenever a new fibre is needed. The objective is to design a routing and a network with node and link capacities that support the routed connections and provide 100% protection against single link failures.

The ring network technology considered in this paper is the MS-SPRing (Multiplex Section Shared Protection Ring) system [10]. To compare MS-SPRing protection with mesh network protection switching, we formulate several linear programs: First, we formulate two ring models: one where the routing of traffic demands is integrated with the ring selection, and one where the routing is performed prior to ring selection. Then, for comparison, we describe a mesh network model for routing paths and allocating protection capacity.

2.1. Ring model—integrated routing

In this section we present a linear program which models a ring network. The characteristic feature of this model is that the optimisation of the paths is performed *concurrently* with the selection of which rings to use. Note that we will not model protected connections *between* the rings, e.g. Drop and Continue [10], in the present work. For this reason, the ring network solution is only protected against link failures and not against ring-connecting node failures.

The overall structure of the model is that of a standard network flow model [1], where the constraints have been refined by parameterising them over a set of potential rings. Constraints are added for modeling the ring architecture, where all links in a ring must have the same capacity, and also constraints that relate the paths with the amount of traffic that is switched between rings. The total cost of the network is then calculated on the basis of the necessary link and switch capacity.

There are two specific components in the MS-SPRing model: nodes and rings:

Nodes Each node consist of three parts, cf. Figure 1. A number of cores (frame and switching fabric), the input/output equipment (tributary cards), and the transit equipment (line cards). Each line card is connected to a fibre.

For each node we define three quantities: The *transit capacity,* z^T, the *I/O flow,* y, and the *number of cores,* z^C. The transit capacity is the total number of fibres on the rings of which the node is a part. Thus, twice as many line cards are necessary, because each node is connected to two other nodes in each ring. The I/O flow is the amount of traffic that switches from one ring to another, plus the amount of traffic that is started or terminated at the node.

The cost of a node is then the sum of the costs of the three different parts. For the experiments we use real-world costs, but due to confidentiality concerns, we only give the relative costs of the various node components here (note that one STM-1 link unit can support one VC-4 connection):

MS-SPRing component	*Max # per core*	*Fractional cost for one component*
Core	1	$C^{core} = 1.00$
Line card (4 × STM-1)	8	$C^{linecard} = 0.36$
Tributary card (4 × STM-1)	4	$C^{tribcard} = 0.38$

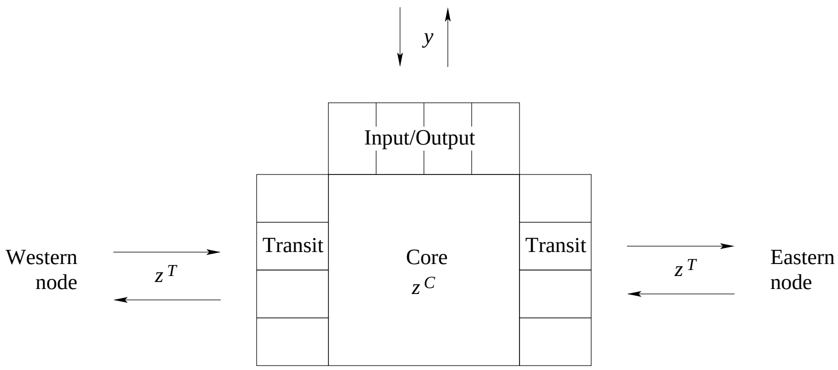

Figure 1. MS-SPRing node model

Rings The MS-SPRing architecture comprises fully bidirectional rings [10]. In an MS-SPRing system, each link of a ring has the same number of fibres for transporting the nominal, bidirectional flow. An identical number of fibres are allocated as backup on each link, and when a link fails, the nodes at each end of the failure switch over to use all the backup links around the ring in the opposite direction. All the connections carried by the ring share the backup capacity, and protection switching is done locally, making it very fast. We model the capacity of each ring as twice the maximal flow on any of the links of the ring.

Figure 2 gives an example of how the protection mechanism works. In the ring, north-south and east-west-going signals are interrupted by a north-east link break. The ring then automatically reroutes the signals the opposite way round the ring on the backup fibres. The paths that are affected by the failure are not entirely rerouted from end to end, but only around the broken link, so this is an example of *link protection*.

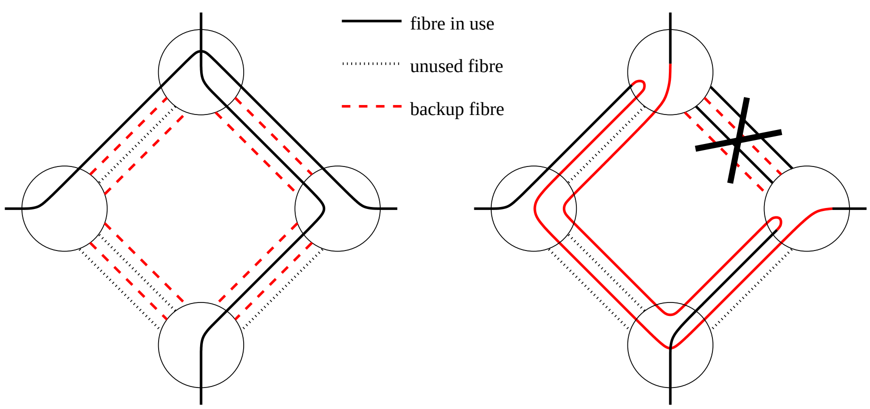

Figure 2. Ring protection: when a link failure occurs, its end nodes reroute the connections in the opposite direction along backup fibres

Based on the prices for the node components we calculate:

- Transit cost per VC-4: $C^T = \frac{2}{4} \cdot C^{linecard}$: One transit connection (either a normal or protection) takes up 1 of 4 STM-1 units connected to the preceding node in the ring and 1 of 4 STM-1 units connected to the next node.

- I/O costs per VC-4: $C^{IO} = \frac{1}{4} \cdot C^{tribcard}$: When entering or exiting a node, 1 of 4 ports on the tributary card is used.

These costs are used in the linear program, which is given in detail in the following. First we describe the indices, constants and variables, and then we present the objective and constraints.

INDICES

$d \in \{1, \ldots, D\}$	demands between (oriented) pairs of nodes.
$s_d \in \{1, \ldots, N\}$	starting node of demand d.
$t_d \in \{1, \ldots, N\}$	terminating node of demand d.
$i, j \in \{1, \ldots, N\}$	node index.
$(ij) \in \{1, \ldots, N\}^2$	oriented link from node i to node j.
$\{i, j\} \subseteq \{1, \ldots, N\}$	non-oriented link $\{i, j\}$, corresponding to oriented links (ij) and $(j\,i)$.
$r \in \{1, \ldots, R\}$	potential rings which may be used.

CONSTANTS

v_d	volume of demand d (in number of VC-4s) to be realised.
$C^{link}_{\{i,j\}}$	transmission costs for each STM-1 on link $\{i, j\}$.
$C^{transit}$	transit costs for each VC-4, for all nodes.
C^{IO}	I/O (ring shift) costs for each VC-4, for all nodes.
C^{core}	core costs for each core part.
$l^r_{(ij)}$	set to 1 if ring r uses oriented link (ij), otherwise 0.
$m^r_{\{i,j\}}$	set to 1 if ring r uses non-oriented link $\{i, j\}$, otherwise 0.
n^r_i	set to 1 if ring r uses node i, otherwise 0.

VARIABLES

$x^{d,r}_{(ij)}$	flow for demand d on the oriented link (ij) in ring r, in VC-4 units
y_i	I/O flow at node i, in VC-4 units
$y^{d,r}_{i+}$	flow for demand d on ring r into node i, in VC-4 units
$y^{d,r}_{i-}$	flow for demand d on ring r out of node i, in VC-4 units
z^R_r	the necessary fibre capacity for ring r including protection capacity, in STM-1 units
z^T_i	necessary transit capacity (including protection) at node i, in STM-1 units. The necessary number of ports is twice this number, one for each direction in the ring
z^C_i	necessary number of cores in node i
$z^L_{\{i,j\}}$	total necessary capacity on link $\{i, j\}$, in STM-1 units

All variables are non-negative reals, as opposed to integers, which enables the linear program solver to find an optimal solution within reasonable time.

LINEAR PROGRAM

Based on the above definitions we now specify the linear program for minimising the costs of installing a ring network in an existing topology. The total cost is a sum of costs for links, cores, line cards and tributary cards:

objective: minimise

$$cost^{ring} = \sum_{\{i,j\}} C^{link}_{\{i,j\}} \cdot z^L_{\{i,j\}} + \sum_i C^{core} \cdot z^C_i + \sum_i C^{transit} \cdot z^T_i + \sum_i C^{IO} \cdot y_i$$

subject to:

$$\sum_r \sum_j l^r_{(ij)} \cdot x^{d,r}_{(ij)} - \sum_r \sum_j l^r_{(ji)} \cdot x^{d,r}_{(ji)} = \begin{cases} v_d & i = s_d \\ -v_d & i = t_d \\ 0 & i \neq s_d, t_d \end{cases} \quad \forall i, d \qquad (1)$$

$$2 \cdot \sum_d m^r_{\{i,j\}} \cdot \left(x^{d,r}_{(ij)} + x^{d,r}_{(ji)} \right) \leq z^R_r \quad \forall r, \{i,j\} \qquad (2)$$

$$\sum_r m^r_{\{i,j\}} \cdot z^R_r = z^L_{\{i,j\}} \quad \forall \{i,j\} \qquad (3)$$

$$\sum_r n^r_i \cdot z^R_r = z^T_i \quad \forall i \qquad (4)$$

$$\sum_j l^r_{(ij)} \cdot x^{d,r}_{(ij)} - \sum_k l^r_{(ki)} \cdot x^{d,r}_{(ki)} = y^{d,r}_{i+} - y^{d,r}_{i-} \quad \forall i, r, d \qquad (5)$$

$$\sum_d \sum_r \left(y^{d,r}_{i+} + y^{d,r}_{i-} \right) = y_i \quad \forall i \qquad (6)$$

$$z^C_i \geq \frac{z^T_i}{16} \ \wedge \ z^C_i \geq \frac{y_i}{16} \quad \forall i \qquad (7)$$

$$x^{d,r}_{(ij)}, y_i, y^{d,r}_{i+}, y^{d,r}_{i-}, z^R_r, z^T_i, z^C_i, z^L_{\{i,j\}} \in R_+ \qquad (8)$$

Constraints (1) are the standard flow constraints: the input flow of demand d to node i must be balanced by the output flow, unless i is the starting or terminating node of the demand. Constraints (2) ensure that the capacity of a ring is twice (because of protection) the maximal flow on any of its links. Constraints (3) and (4) calculate the necessary link and transit node capacities. Note that for given i, r we have $l^r_{(ij)} \neq 0$ for at most one j (and similarly for $l^r_{(ji)}$), because there is exactly one ingoing and one outgoing link of each node in ring r. Thus, all the right hand sides of constraints (5) consist of no terms if node i is not in ring r, and exactly two terms if node i *is* in ring r: $x^{d,r}_{(ij)} - x^{d,r}_{(ki)}$. This is exactly the volume of flow for demand d entering ring r at node i. If this volume is positive, $y^{d,r}_{i+} > 0$, and if it is negative, $y^{d,r}_{i-} > 0$. Either way, the necessary I/O port capacity is calculated by constraints (6). Finally, constraints (7) ensure that there are enough switch cores to hold the transit and I/O ports in each node.

2.2. Ring model—prerouting

The model in the preceding section optimises the routing and the ring selection concurrently. However, a popular way of designing ring networks is to split the optimisation into two stages:

first, all the demands are routed, and then based on these paths the rings are selected and their capacities determined [2, 11].

It is straightforward to extend the linear program to handle prerouted traffic: We let the shortest route for each demand be given in a two-dimensional binary matrix p such that $p^d_{(ij)} = 1$ indicates that the shortest route for demand d uses link (ij). Then we replace constraints (1) with the constraints

$$\sum_r x^{d,r}_{(ij)} = v_d \cdot p^d_{(ij)} \qquad \forall (ij), d. \tag{9}$$

Constraints (9) ensure that if traffic is flowing through a link, given by $p^d_{(ij)}$, it is assigned to one or more rings, given by $x^{d,r}_{(ij)}$.

2.3. Mesh model

To compare the cost of the ring based solution we formulate a linear program which finds an optimal mesh based solution. In this mesh network model, each node is a general DXC (digital cross-connect), and traffic is protected from single-link failures by path protection: when a link fails, broken paths are re-routed end-to-end. The remaining working link capacity on the broken paths is released and can be used for other connections in the re-routing.

The linear program we formulate is a standard link-path model which requires pre-calculated paths for each demand [5, 9]. It is also possible to formulate a linear program using an arc-flow model, which does not require pre-calculation of all the paths [3]. Even though the latter model is more similar to our ring model, it is not optimal in the sense that when link capacities are given it may not find the optimal solution since several backup paths may be needed.

The idea in the link-path model is the following: For each node pair which has a connection demand $v_d > 0$, all possible paths between the nodes are pre-calculated. The linear program then satisfies the connection demand by assigning it to some of the paths.

A number of failure situations are defined, in our case one for each link in the network, corresponding to each of these links being broken. We define further a “nominal situation,” where none of the links are broken, and require for each failure situation that there are enough backup routes assigned to cover for the loss when the nominal routes are broken.

The link model is the same as for the ring architecture, but the nodes are slightly different because they are modelled as DXCs, cf. Figure 3. The DXC looks somewhat similar to the add-drop node in figure 1. The difference is that we do not distinguish between transit traffic and I/O traffic, as they are both handled by the same port type. If there is so much traffic that more than one DXC core is necessary, we assume that the connections can be made in such a way that each path that passes through the node only uses one of the cores, so there is no traffic between them.

Again we do not present absolute costs, but give the costs of the various DXC components relative to the cost of one MS-SPRing core:

DXC component	*Max # per core*	*Fractional cost of one*
Core	1	$C^{core} = 9.24$
STM-1 magasin (capacity of 8 cards)	16	$C^{magasin} = 1.10$
Electrical STM-1 card	128	$C^{card} = 0.14$

As can be seen, networks based on DXC nodes can be significantly more expensive, as each component is more expensive than its MS-SPRing counterpart.

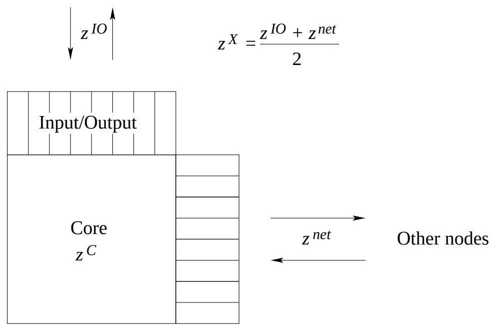

Figure 3. DXC node model

Based on these prices we now calculate the switching costs per VC-4: $C^{switching} = 2 \cdot (\frac{1}{8} \cdot C^{magasin} + C^{card})$. Again we use a capacity of two STM-1 per VC-4 signal passing through the node, but this time we do not discriminate between signals which pass through and signals which enter or exit at the node. This price is used in the linear program given in the following sections.

INDICES

$d \in \{1, \ldots, D\}$	demands between (oriented) pairs of nodes.
$r \in \{1, \ldots, R\}$	routes for each demand.
$s \in \{0, \ldots, S\}$	failure (broken link) situations.
$i, j \in \{1, \ldots, N\}$	node index.
$\{i, j\} \subseteq \{1, \ldots, N\}$	non-oriented link $\{i, j\}$, corresponding to oriented links (ij) and (ji).

CONSTANTS

v_d	volume of demand d (in number of VC-4s) to be realised.
$C^{link}_{\{i,j\}}$	transmission costs for each VC-4 on link $\{i, j\}$
$C^{switching}$	switching costs for each VC-4, for all nodes
C^{core}	core costs per VC-4
$p^{d,r}_{\{i,j\}}$	set to 1 if path (d, r) uses link $\{i, j\}$, otherwise 0.
$n^{d,r}_i$	set to 1 if path (d, r) uses node i, otherwise 0.
$b^{d,r}_s$	set to 0 if path (d, r) is broken in failure situation s, otherwise 1.

VARIABLES

$x^{d,r}$	capacity used on path (d,r) when no links are broken, in number of VC-4s.
$y_s^{d,r}$	protection capacity added on path (d,r) in situation s, in number of VC-4s.
z_i^X	necessary switching capacity (number of switched connections) for node i for all failure situations, in number of VC-4s.
z_i^C	necessary number of cores in node i
$z_{\{i,j\}}^L$	necessary link capacity for link $\{i,j\}$ in all failure situations, in number of VC-4s.

LINEAR PROGRAM

objective: minimise

$$cost^{mesh} = \sum_i C^{switching} \cdot z_i^X + \sum_i C^{core} \cdot z_i^C + \sum_{\{i,j\}} C_{\{i,j\}}^{link} \cdot z_{\{i,j\}}^L$$

subject to:

$$\sum_r x^{d,r} = v_d \qquad \forall d \tag{10}$$

$$\sum_r b_s^{d,r} \cdot \left(x^{d,r} + y_s^{d,r}\right) = v_d \qquad \forall d, s \tag{11}$$

$$\sum_{d,r} n_i^{d,r} \cdot b_s^{d,r} \cdot \left(x^{d,r} + y_s^{d,r}\right) \leq z_i^X \qquad \forall i, s \tag{12}$$

$$z_i^C \geq \frac{z_i^X}{64} \qquad \forall i \tag{13}$$

$$\sum_{d,r} p_{\{i,j\}}^{d,r} \cdot b_s^{d,r} \cdot \left(x^{d,r} + y_s^{d,r}\right) \leq z_{\{i,j\}}^L \qquad \forall \{i,j\}, s \tag{14}$$

$$x^{d,r}, y_s^{d,r}, z_i^X, z_i^C, z_{\{i,j\}}^L \in R_+ \tag{15}$$

Constraints (10) ensure that the connection demand is fulfilled when no links are broken, and constraints (11) ensure sufficient backup capacity in link failure situations. Constraints (12) calculate the necessary switching capacity, and as each switched connection uses two ports, dividing this capacity by 64 in constraints (13) yields the number of necessary DXC cores. Finally, constraints (14) ensure sufficient capacity on each link.

2.4. Obtaining integer solutions

In general, the solutions obtained from the linear programs are not integral, i.e. solutions with an integral number of switches and fibres. To obtain this, we round up the values found by the linear programs.

For the ring network model, we round up the values of variables $x_{(ij)}^{d,r}$, and then re-calculate $z_{\{i,j\}}^L$, z_r^R, z_i^T and y_i, rounding the latter two up to a multiple of 4 because each port handles 4 STM-1 links. Then we calculate the integer number of cores needed for each node: Each core may support only one ring and each ring can have at most 16 VC-4 connections, hence for each ring r in which the node participates, the traffic z_r^R is divided by 16 and rounded up. The number of needed cores is then the total sum of these core numbers.

For the mesh network, the flows $x^{d,r}$ and $y_s^{d,r}$ are rounded up; the resulting link costs can then be calculated in a straightforward manner. For each node we need to calculate the number of cores, the number of magasines and the number of electrical STM-1 cards. Whereas in the ring architecture each node requires different core components for the different rings in which it is participates, one DXC can handle connections to all the neighboring nodes. The number of cores needed is thus the traffic through the node divided by 64 and rounded up, and the number of magasines is the traffic divided by 4 and rounded up. Finally, the needed number of STM-1 cards is simply the traffic rounded up.

3. EXPERIMENTAL DESIGN

We now use the linear programs to perform a series of experiments which can give us some answers to the key questions mentioned in the introduction. The major characteristics of the solutions that we are interested in are

- the cost of the network and its components
- the number of rings in the ring architecture
- the mean and variance of the number of nodes per ring

NETWORK TOPOLOGY & TRAFFIC

In the experiments we use a real world network topology, the backbone network of the Danish telecom operator TDC, using realistic link costs. The network has 12 nodes and 25 links, giving an average node degree of 3.8, and it must satisfy 22 demands with a total demand volume of 1734 VC-4s. The total number of rings with at least 3 and no more than 16 nodes is 1167, with an average length of 9 non-oriented links, which are modeled by 18 oriented links.

Even though this network is fairly small, the LP model for the ring network is large; this can be seen by looking at the number of flow variables $\left|x_{(ij)}^{d,r}\right| = 22 \cdot 1167 \cdot 18 \approx 460{,}000$.

When a network specialist was given the task of manually designing a ring network in this topology, he selected 3 rings, which covered the entire network, using only 14 of the potential 25 links. Using only these 3 rings an optimal routing was found using the ring LP and the costs were compared with the automatic ring design. To make a fair comparison, we also perform experiments using only the same 14 links.

DESIGN METHOD

We perform experiments for the two architectures: ring and mesh network, and for two ways of routing paths: IR or PR.

NETWORK ELEMENT COSTS

There is a significant cost difference between the necessary node equipment in the ring network and the mesh network. To study the influence of this price difference we simply reduce the combined node costs by a factor α that is varied, e.g. $\alpha = 0.5$ or $\alpha = 0.75$.

EXPERIMENTAL SERIES

Using the various experimental parameters described in the preceding paragraphs, we perform a series of experiments:

Manual vs. automatic ring design: Considering only IR path design, we vary the architecture (ring/mesh), the number of links available (14/25) and the ring design method (manual/automatic).

Break-even cost factor: Considering only the mesh architecture, we vary the number of links available (14/25) and the network element cost factor α, to determine when the total network costs are the same for ring and mesh architectures.

Integrated routing vs. prerouting: Considering only ring architecture, we vary the ring design method (manual/automatic), the number of links available (14/25) and the path design method (IR/PR).

4. RESULTS & DISCUSSION

The experiments were performed with the LP/MIP optimiser CPLEX 6.61 [6] on a 440MHz HP Series 9000 Model J7000 with 4Gb RAM, each experiment running for 1–20 minutes. We note that only the barrier solver of CPLEX was able to cope with the larger problems (IR path design with 25 links); all the other optimisation methods would either run out of memory or would not finish within a 2 hour limit.

MANUAL VS. AUTOMATIC RING DESIGN

The results from the first set of experiments are shown in Table 1, with the integer-rounded values in Table 2. Note that in both tables, the cost has been split out on the various network components: links, transit equipment (line cards), I/O equipment (tributary cards) and cores. The "*% Cost*" column gives the total cost relative to that of the manual ring design with optimal routing.

Table 1. Optimal LP solutions. The price unit is the cost of one thousand MS-SPRing cores

Arch	*Links*	*Ring design*	*Link*	*Transit*	*I/O*	*Core*	*Total*	*%*	*Rings*	*Nodes/Ring avg.*	*std. dev.*
ring	14	manual	1.21	1.32	0.36	0.45	3.34	*100.0*	3	5.33	0.82
		auto	1.09	1.18	0.33	0.41	3.02	*90.4*	6	7.50	6.10
	25	auto	0.68	0.70	0.35	0.24	1.97	*59.0*	20	4.20	6.60
mesh	14	—	0.54	2.52	—	0.65	3.71	*111.1*	—	—	—
	25	—	0.39	2.03	—	0.52	2.94	*88.0*	—	—	—

Table 2. Optimal LP solutions rounded up to integral values. The price unit is the cost of one thousand MS-SPRing cores

Arch	*Links*	*Ring design*	*Link*	*Transit*	*I/O*	*Core*	*Total*	*%*	*Rings*	*Nodes/Ring avg.*	*std. dev.*
ring	14	manual	1.20	1.32	0.36	0.46	3.35	*100.3*	3	5.33	0.82
		auto	1.09	1.18	0.35	0.42	3.04	*91.0*	6	7.50	6.10
	25	auto	0.68	0.70	0.35	0.29	2.02	*60.5*	20	4.20	6.60
mesh	14	—	0.54	2.53	—	0.73	3.81	*114.1*	—	—	—
	25	—	0.39	2.04	—	0.58	3.01	*90.1*	—	—	—

We observe that the rounding up does not change the figures significantly, which is due to the fact that the demand volumes are large in relation to the unit size of the network elements. From

Table 3. Integer solutions for mesh network costs as a function of the mesh node cost factor α.

Links	α	*Link cost*	*Transit cost*	*Core cost*	*Total cost*	*%*
14	0.85	0.54	2.16	0.62	3.32	*99.4*
	0.80	0.54	2.03	0.58	3.15	*94.3*
	0.75	0.54	1.90	0.55	2.99	*89.5*
25	0.65	0.39	1.33	0.38	2.10	*62.9*
	0.60	0.39	1.22	0.35	1.97	*58.9*
	0.55	0.39	1.12	0.32	1.84	*55.1*

the figures we further find the most cost-effective network is the automatically designed ring-network, using 25 links. This result is somewhat surprising, since one would normally expect a better utilisation of the network using the mesh design [11]. The reason can clearly be seen in the cost of the node equipment which is the dominating cost for the mesh network. Even though the savings on the links are substantial in the mesh case, up to 75%, the mesh solution is still more expensive than the ring solution. Notice further that using many rings pays off: The automatic solutions contain 6 and 20 rings. Hence, the LP solver achieves significant savings, up to 40%, by using the appropriate rings and routing the demands optimally on these rings.

BREAK-EVEN COST FACTOR FOR MESH NODE COMPONENTS

Perhaps surprisingly, the results from the preceding section indicate that a network based on the ring architecture is cheaper to establish than one based on the protected mesh architecture—the reason being that the node costs (i.e. core and transit costs) are significantly higher in the mesh case.

Given a solution $(\vec{z}^X, \vec{z}^C, \vec{z}^L)$ to the mesh network optimisation problem, a rough approximation of the cost of a mesh network where node costs are multiplied by a factor α is

$$\begin{aligned} cost^{total}_{mesh}(\alpha) &= \sum_i \alpha C^{switching} \cdot z^X_i + \sum_i \alpha C^{core} \cdot z^C_i + \sum_{\{i,j\}} C^{link}_{\{i,j\}} \cdot z^L_{\{i,j\}} \\ &= \alpha \cdot (cost^{transit}_{mesh} + cost^{core}_{mesh}) + cost^{link}_{mesh}. \end{aligned}$$

Thus, an approximate break-even factor can be determined by solving a equation for α:

$$\begin{aligned} & cost^{total}_{ring} = cost^{total}_{mesh}(\alpha) = \alpha \cdot (cost^{transit}_{mesh} + cost^{core}_{mesh}) + cost^{link}_{mesh} \\ \Updownarrow \\ & \alpha = \frac{cost^{total}_{ring} - cost^{link}_{mesh}}{cost^{transit}_{mesh} + cost^{core}_{mesh}} \end{aligned}$$

so we find $\alpha_{25\text{links}} = \frac{2.02-0.39}{2.04+0.58} \approx 0.6$ and $\alpha_{14\text{links}} = \frac{3.04-0.54}{2.53-0.73} \approx 0.8$. However, the changed node costs may influence the optimal solution, so we must perform some experiments where we vary α around these values, the results of which are shown in Table 3. By comparing with the costs for ring networks given in Table 2 (3.04 using 14 links, and 2.02 using 25 links), it can be seen that protected mesh networks indeed are competitive when $\alpha \leq 0.6$.

INTEGRATED ROUTING VS. PREROUTING OF TRAFFIC DEMANDS

The effect of integrating the optimisation of the traffic routing with the ring selection can be seen in Table 4. It is interesting to note that there is a 10–15% loss because of the prerouting.

Table 4. The effect of the path design (IR/PR) on ring network costs, LP solution rounded up

Links	*Ring design*	*Path design*	*Link*	*Transit*	*I/O*	*Core*	*Total*	*%*	*Rings*	*Nodes/Ring avg.*	*std. dev.*
			Cost								
14	manual	IR	1.21	1.32	0.36	0.46	3.35	*100.3*	3	5.33	0.82
		PR	1.45	1.53	0.26	0.53	3.77	*112.8*	3	5.33	0.82
	auto	IR	1.09	1.18	0.35	0.42	3.04	*91.0*	6	7.50	6.10
		PR	1.36	1.45	0.28	0.50	3.60	*107.8*	5	7.20	5.90
25	auto	IR	0.68	0.70	0.35	0.29	2.02	*60.5*	20	4.20	6.60
		PR	0.93	0.91	0.29	0.32	2.44	*73.1*	18	4.00	4.20

5. CONCLUSION

In this article we have investigated the costs for optimised protected networks. We have compared path protected mesh networks and ring networks, using the same links, and surprisingly the ring networks are more than more than 25% cheaper. The cost difference is solely due to the higher cost of the switches used in the path protected solution: If the prices for these switches were reduced by a factor of 0.6 they would be competitive. Further, the routing of the demands plays a significant role in the costs: Simply using a standard shortest path routing followed by optimal ring selection results in a loss of 10 to 15%. Finally, it is not trivial to design ring networks: A manual design was improved more than 40% by the optimal LP design.

The conclusions drawn in this article are based on only one network, the backbone network of TDC. They are, however, so significant that TDC will consider re-evaluating the way the ring networks are designed, with an aim to reduce costs. Obviously it would be interesting to apply the presented design methods to other networks, but unfortunately the confidentiality of detailed information about backbone networks of telecommunication operators has prevented us from doing this.

More investigations should be carried out to determine exactly the network size limits for using this LP-based ring design approach. For networks that exceed this limit several modifications could be applied to obtain approximations to the optimal design. Since the number of potential rings is the fastest growing factor of the problem, it seems reasonable to apply some heuristic to reduce the number of rings actually considered in the linear program. By looking at the solutions derived in this work, it seems that the smallest rings containing each demand are preferred; smaller capacities on the links might change this, though. Another possibility could be to reduce the number of demands by dividing the network into two or more parts, optimising each part individually and only exchanging aggregate demand information at the dividing nodes—this would also reduce the number of rings considered. Finally, for very large networks good heuristics will surely be needed [11].

ACKNOWLEDGMENTS

The authors would like to thank Bjarke Skjoldstrup and Søren Blaabjerg for close cooperation in the modelling phase and for the necessary network data.

REFERENCES

[1] William J. Cook, William H. Cunningham, William R. Pulleyblank, and Alexander Schrijver. *Combinatorial Optimization*. Wiley, 1998.

[2] Steven Cosares, David Deutsch, Iraj Saniee, and Ondria Wasem. Sonet toolkit: A decision support system for designing robust and cost-effective fiber-optic networks. *Interfaces*, 25, 1995.

[3] Bharat Doshi, Subrahmanyam Dravida, P. Harshavardhana, Oded Hauser, and Yfei Wang. Optical network design and restoration. *Bell Labs Technical Journal*, January–March 1999.

[4] Linda Morales Gardner, I. Hal Sudborough, and Ioannis G. Tollis. NetSolver: A software tool for the design of survivable networks. In *IEEE Global Telecomunication Conference.*, volume 2, pages 926–930. IEEE, IEEE, 1995.

[5] Arne John Glenstrup, Christian Fenger, and Thomas J.K. Stidsen. Full design of robust optical networks. In *Fifteenth Nordic Teletraffic Seminar*, 2000.

[6] ILOG CPLEX 6.5, users manual. ILOG Cooperation, March 1999.

[7] Hannan Luss, Moshe B. Rosenwein, and Richard T. Wong. Topological network design for sonet ring architecture. *IEEE Transactions on Systems, Man, and Cybernetics*, 28(6):780–790, November 1998.

[8] G. Dave Morley and Wayne D. Grover. Current approaches in the design of ring-based optical networks. In *Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering*, page 6. IEEE, 1999.

[9] Michał Pióro, Thomas J.K. Stidsen, Arne John Glenstrup, Christian Fenger, and Henrik Christiansen. Design problems in robust optical networks. In *Networks 2000*, September 2000.

[10] Mike Sexton and Andy Reid. *Broadband Networking: ATM, SDH, and SONET*. Artech House, Boston • London, 1997.

[11] Lunchakorn Wuttisittikulkij, Charoenchai Baworntummarat, and Thanyaporn Iamvasant. A comparative study of mesh and multi-ring designs for survivable WDM networks. *IEICE Transactions on Communications*, E83-B(10):2270–2277, 2000.