Computer Science Department Technical Report
University of California
Los Angeles, CA 90024-1596

PROTECTION PLANNING IN TRANSMISSION NETWORKS

E. Pedrinelli
M. Barezzani
M. Gerla

June 1992
CSD-920031

Protection Planning in Transmission Networks.

M. Barezzani, E. Pedrinelli
Telettra S.p.A.
Cinisello Balsamo (MI)
Italy

M. Gerla
UCLA
Computer Science Dept.
Los Angeles
USA

Abstract.

In the present work we define several routing and protection problems which arise in transmission network planning and management. We then describe network planning algorithms and software tools aimed at providing network protection. Finally we introduce the concept of Automatic Network Protection System and briefly discuss some related problems.

Introduction.

In this paper we focus on the protection of transmission networks. We present algorithms and tools which are useful both to support the activity of network planners and to build an Automatic Network Protection System. The work is divided into four main sections. In the first section a definition of general and constrained routing and protection problems in a transmission network is given. The second section focuses on protection planning tools describing their functions and implementation. In this section a new algorithm for multi constrained routing and protection problems in presented. The concept of Automatic Network Protection System is introduced in the third section along with a discussion of its main features. Finally, the fourth section presents the results of a protection study carried out using the described algorithms.

1. Routing and Protection in Transmission Networks.

A transmission network can be defined as a set of transmissive media and equipment intended as a transport infrastructure for a generic telecommunications service network (e.g. PSTN, DDN, LLN, PSN, B-ISDN, WLLN, see also figure 1).

In this paper we restrict our analysis to transmission networks in which Digital Cross Connects (DXCs) are used. The flexibility and the manageability of such networks can be usefully exploited for routing and protection purposes (1). This is particularly true if the transmission net-

work is equipped with an Automatic Network Protection System. In a multi layer environment supported by DXCs, protection operations are usually performed at plesiochronous 140 Mbit/s level and/or synchronous VC-4 level.

In the following sections the transmission network will be modeled as an undirected graph.

1.1 Routing in a Transmission Network.

The routing problem in a transmission network can be generally stated as follows:

Given

G(N,L) :	a network with a finite set of nodes N and a finite set of undirected arcs L;
$c_i, i \in L$:	the capacity (possibly infinite) of arc i expressed as number of elementary transmission flows it can carry;
$k_i, i \in L$:	the incremental cost index for arc i;
T = {(nx, ny, t)}:	a finite set of traffic requirements: nx, ny ∈ N; t is the number of full duplex elementary traffic flows to be routed between nx and ny;

then

find a path for each traffic requirement minimizing the overall network cost

$$c = \sum_{i \in L} k_i \, q_i$$

where q_i is the number of elementary transmission flows to be provided on arc i to carry the given traffic;
subject to a set of constraints (e.g. capacity, line quality, etc.)

A specific routing problem on the transmission network can be formulated in many different ways depending on the definition of arc costs, and on the choice of constraints. For example, if arc cost = 1 for all arcs, we have the so called "min hop" routing problem. In a topology design problem, arc costs typically reflect the cost required to install and operate a communications trunk on that arc. As for the set of constraints, the most common cases are the following:

- unconstrained routing: in this general routing problem no constraint is imposed; each traffic re-

quirement is routed on the minimum cost path between source and destination nodes. Arc capacities are supposed infinite. (3, 7)

- arc capacity constrained routing; in this problem the capacity of the arcs is finite and it cannot be exceeded; routing a traffic requirement implies finding the minimum cost path traversing unsaturated arcs only.

- quality constrained routing; in this problem the transmissive quality of arcs (for example in terms of BER) is kept into consideration: a traffic requirement can only be routed on paths of quality greater than that it requires. That means that a traffic requirement must be routed on the minimum cost path having adequate quality.

- path length constrained routing; in this problem the maximum path length is fixed and cannot be exceeded.

- multi-constrained routing is a routing problem in which some or all of the above mentioned constraints simultaneously hold.

All the preceding routing formulations can also be extended to broadcast and multicast traffic as well as to multiple priorities. Broadcast traffic is from a source node to all the other nodes in the network; multicast traffic is from a source node to a subset of nodes.

The routing algorithm plays a key role in network protection since, after a failure, some of the requirements must be rerouted exploiting the extra capacity available in the network. This rerouting is typically performed using a constrained routing algorithm.

1.2. Protection Problems in Transmission Networks.

The problem of protection of a transmission network can be divided into two independent subproblems: the design problem and the analysis problem. The former involves the computing of the optimal number and location of the spare transmission resources in order to protect a given percentage of the traffic from a certain class of failures. The analysis problem consists of evaluating the degree of protection of a given network (for example counting the number of not-reroutable transmission flows for a given class of failures).

Design problem:

Given	
G(N, L):	a network with a finite set of nodes N and a finite set of undirected arcs L;
$k_i, i \in L$:	the cost index for arc i (i.e. the cost of installing spare capacity on arc i);
P:	a set of paths on which the transmission

	traffic has been routed;
$q_i, i \in L$:	the arc capacity utilized to carry the traffic (working capacity) for a given routing solution
F(N, L)	a failure class

then
minimize the overall cost of the spare capacity
that must be added to the network G to permit
rerouting on an alternate path,
by-passing the faulty elements, of all the
traffic affected by the given failure class

A failure instance is defined to be a set of out of service network elements (nodes and/or arcs). We will say that two failure instances are equivalent when their sets are composed of the same number of nodes and/or arcs. A failure class is defined to be the set of all the equivalent failures (e.g. all single node failures). For the purpose of protection evaluation, considering a failure class implies simulating the occurrences of all the individual failure instances, one at a time.

The design problem can be slightly modified to account for input spare capacity (i.e. the spare transmission capacity that is present in part of the network). In this case the design problem will take the initial spare capacity into account and will add other spare resources as necessary.

Multiple priority levels can also be considered in the rerouting of traffic after failures. In this case, the design problem may require that only a subset of the priority classes be able to survive certain types of failures.

Closely related to the Design Problem is the Analysis Problem. In fact, the analysis of a "protected" network allows us to verify if such network was properly designed. Apart from verification of acceptable performance in a preplanned network, the Analysis Problem must be solved when carrying out sensitivity studies (e.g. determination of robustness to changes in traffic pattern, or in line quality). As a difference from the design problem, the analysis problem assumes that arc capacities are given.

Analysis problem:

Given

G(N, L):	a network with a finite set of nodes N and a finite set of undirected arcs L;
$c_i, i \in L$:	the capacity of arc i;
P:	a set of paths on which the transmission traffic has been routed;
$q_i, i \in L$:	the capacity utilized to carry the traffic (working capacity) obtained from the solution of the routing problem;

F(N, L) a failure class

then
find the fraction of not-reroutable traffic.
This fraction is a measure of the degree of network protection.

Traffic priorities can also be considered in the protection analysis problem. Moreover, in this case, it is possible to protect higher priority traffic by preempting lower priorities. The preemption mechanism has no meaning in the design problem case, since in that case all the traffic affected by the failures must be rerouted, regardless of priority.

Both routing and protection problem can be solved by means of heuristic iterative methods based on a suitable shortest path algorithm. The protection problem, in fact, can be regarded as a special case of the general routing problem. Protecting the transmission network from a given failure means finding an alternative route on the spare network for each working traffic flow affected by that failure.

It should be noted that, for consistency of the results, the routing algorithm used in the analysis of a network should be very similar to the algorithm actually implemented in the network. Only under these conditions will the results of the analysis offer a realistic estimate of the protection to be expected from the network. This observation implies that the analysis problem and the implementation problem are closely intertwined. Moreover, in the selection of a solution to the analysis problem, one must account for the feasibility of implementing such solution in the actual network. For example, the computation time required for the analysis must fall within the response time constraints set by the applications in the case of recovery from network failures. This requirement is obvious in the case of a centralized implementation of the recovery algorithm. In a distributed implementation, other factors must be taken into account, as it shall be discussed later.

2. Planning for Protection: Network Algorithms.

If we try to match the protection problem formulation of the preceding section with the real world situation we immediately find that a real network is subject to many more constraints than we originally considered. To render the problem more manageable, in this paper we will focus only on the constraints imposed by transmission media capacity (arc capacity) and transmissive quality (path quality). Planning algorithms are required to manage at least these constraints if their results are to be used for dimensioning and protection of a real transmission network.

Based on our previous observations, we propose to attack this constrained protection problem using a constrained version of the shortest path algorithm. This algorithm will be the basic building block for the implementation of more complex heuristic protection procedures.

Broadly speaking, there are two classes of routing algorithms which can be proposed for the im-

plementation (and, therefore, analysis) of protection: (1) distributed algorithms, and; (2) centralized algorithms.

2.1. Distributed Routing Implementations

Starting with the first class, a distributed routing algorithm commonly used in computer networks is the Bellman-Ford algorithm (also known as the "Old ARPANET" Routing Algorithm, or the Distance Vector Algorithm) [2]. Without entering in the details, it will suffice to say that this algorithm basically permits each station to build the vector of distances (according to a well defined metric) to all destinations along with the vector of next nodes on the shortest routes to such destinations. The procedure is fully distributed, and requires the periodic exchange of distance vectors between adjacent nodes. In its original version, the Bellman-Ford Algorithm can find only unconstrained shortest paths. A simple extension, however, can generate also constrained shortest paths. For instance, a version of this algorithm called Bandwidth Routing Algorithm (14) finds, for a given origin/destination pair the list of shortest paths (based on an arbitrary arc cost metric) for all possible values of the bandwidth constraint (that is, the residual bandwidth available on the path). The list consist of pairs (L, B), Length and Bandwidth, and is ordered by increasing L. It turns out that B is also monotonically increasing. In fact, only the "dominant" (L,B) pairs need to be generated, for which both L and B are increasing. Namely, (L', B') is said to be dominated by (L_i, B_i) if, for instance, $L' > L_i$ and $B' \leq B_i$. Obviously, the dominated solution is of lower quality than the dominant solution, and thus need not be reported in the list. The list of dominant solutions was found to be typically quite small, even in large networks (say, on average, 2 or 3 paths in a 60 node network). Thus, the list of paths to be kept around is indeed quite manageable (14).

One drawback of the Bandwidth Routing algorithm is that it can take it into account only one constraint (e.g. bandwidth). In our protection application, we have two simultaneous constraints: bandwidth and quality (i.e. BER). This problem can actually be managed in an ad hoc, suboptimal fashion, by running two constrained algorithms in parallel: (1) shortest path and bandwidth, and; (2) shortest path and quality. For the former, we post, next to each length and bandwidth pair, the quality associated to the pair. For the latter, likewise, we post the bandwidth next to each length and quality pair. Then, we search the two tables in parallel, looking for the shortest path solution which satisfies both bandwidth and quality constraints.

Apart from the fact that the Bandwidth Routing algorithm requires ad hoc modifications to be made to work with two simultaneous constraints, there are reasons that limit its applicability to the protection problem:

(1) DXC nodes must be intelligent, i.e. must exchange and process update messages; and must maintain routing tables. Current DXCs do not have such capability.
(2) the algorithm is slow in reporting changes in bandwidth which occur as requirements are rerouted.

Namely, the second point suggests that the information in the routing tables may become obsolete as circuits are rerouted. Of course, the algorithm is dynamic. It will update its tables fol-

lowing the change in the bandwidth available; but, typically it will require several iterations to complete each update. For correctness, one should carry out a complete update after each circuit rerouting. This would be too time consuming for some applications which require that the entire reconfiguration be completed in a matter of seconds.

Another distributed routing algorithm which is a potential candidate for protection implementation is the Link State algorithm, also known as the "New ARPANET" algorithm (15). In this algorithm, each node (in our case, DXC) periodically broadcasts the state of its links to all other nodes in the network. In our case, the relevant information will include up/down state, bandwidth, quality and cost. From the periodic link state messages received from all other nodes, each node can thus build a network topology database. From this topology, it can then compute constrained shortest paths locally, using any of the conventional, centralized shortest path algorithms.

Again, the main drawback here is (apart the DXC intelligence) the fact that the network bandwidth changes dynamically as rerouting occurs, thus forcing the nodes to frenetically update the databases in order to catch up with rerouting. As a consequence, either the rerouting must be slowed down, or the line and processing overhead becomes very high (because of frequent updates), or the rerouting decisions are not very accurate (because they are based on obsolete information) and thus lead to blocking (and more rerouting).

2.2. Centralized Routing Implementations

Because of the above mentioned limitations of the distributed procedures, we have opted for a centralized implementation. Namely, DXC's report failures to a center, which then recomputes shortest paths and coordinates the rerouting of the failed circuits. Several solutions are available for the centralized routing implementation.

A quasi optimal solution can be obtained using multicommodity flow optimization techniques (e.g. Flow Deviation method (16)). These techniques attempt to redistribute the requirements affected by the failure over the paths with extra capacity in an even way, thus making the best use of available resources. The solution is quasi optimal because the requirements are rerouted in discrete increments. It would be actually optimal if requirements could be subdivided into infinitesimal increments (which is obviously not possible). While the original Flow Deviation method does not handle quality constraints, it can however be modified to incorporate such constraints. Basically, the problem reduces to finding "constrained" shortest paths, rather than unconstrained ones, at each iteration of the Flow Deviation algorithm.

The advantage of the Flow Deviation method is to compute "in one shot" new feasible routes for all circuits. The drawback however is the fact that the optimization procedure is quite time consuming, since it typically requires several iterations to converge. Furthermore, this procedure does not lend itself easily to the case of multiple priority classes with preemption. Thus, it is not well suited for multi priority applications requiring complete reconfiguration within seconds.

To overcome these problems, we have decided to find a compromise between accuracy and

speed. Improving the speed basically implies running a shortest path algorithm as few times as possible. This led us to use a table of precomputed paths, as described in the next section.

2.3. Proposed Approach

The proposed approach is a centralized approach. The network control center maintains a K-shortest paths table which contains, for each pair of nodes, a set of K paths selected on a minimum cost basis (14). Whenever a new path is needed during rerouting, the list of K paths with the desired origin and destination is extracted from the K paths table. Cost values, bandwidth and transmissive quality parameters are computed in real time for the K paths using an up-to-date data base available at the center. The arc incremental cost is infinite when no residual arc capacity is available. The cost is unchanged otherwise. The least cost path which satisfies the bandwidth and quality requirement is chosen.

The probability of success in finding a feasible path with this scheme is a function of K. Values of K around 5 are suggested for medium size networks. When this methods fails to find a feasible path, the Dijkstra's algorithm (3) is performed on the topology data base to find the best quality path between the given origin and destination.

With this approach we need to run a simple algorithm (Dijkstra's) only once in a while. Still, a possible drawback stems from the fact that the K-shortest paths may not be disjoint, i.e. they may share a single network element (node or arc). Thus, a single failure affecting the common element may force us to run the algorithm quite often.

To overcome this problem we have carried out a modification in which the K-shortest paths table is substituted by a K-paths table composed by the first N (N <= K) arc disjoint shortest paths and by K-N paths, different from the first N (but not necessarily arc disjoint). In figure 2, a sample of the K paths table for the represented network is given; in figure 3 a flow chart of the procedure is shown.

So far we have described the approach to solve the protection problem for point-to-point circuits, but we are also interested in broadcast and multicast circuits. The use of a constrained minimum spanning tree algorithm (2) is not advisable because we wish to recompute only the failed part of the tree during the reconfiguration process, leaving untouched the unfailed part. The conventional min spanning tree algorithm recomputes the entire tree after failure. Moreover this algorithm is not suited for multicast circuits due to the presence of intermediate nodes that are not destinations of traffic. To overcome the above problems a shortest tree algorithm can be used. Namely, for each broadcast or multicast traffic requirement, the shortest tree getting to all the required destination nodes is built. The shortest tree results from the superimposition of the shortest paths connecting the source node to all the destination nodes. These paths are computed by means of the K paths table as described before. In general the shortest tree has higher cost than the min spanning tree. It is, however, easier to maintain. In fact, the shortest tree approach requires to rebuild only the branch of the tree in which the failure has occurred.

3. Protection Procedures.

Based on the K-shortest path routing algorithm, we have so far implemented four procedures for transmission network planning and protection. The procedures, now integrated into a software based package named "Network Planning Tools" (NPT), are the following:

- routing;

- network and traffic growth;

- protection degree analysis;

- spare network design.

These procedures can be applied both to point to point as well as to broadcast and multicast traffic.

3.1. Routing Procedure.

The routing procedure of the NPT solves the multi constrained version of the routing problem described in section 1.1 using the K paths approach. It is described in details in figure 4.

3.2. Network and Traffic Growth Procedure.

This procedure allows us to study the evolution of a transmission network both for network expansion (node or arc additions and/or capacity increases) and for traffic growth. Again the problem is essentially a routing problem in which the already existing network situation must be kept into account. The procedure is very similar to the routing procedure, but for the fact that it operates on the modified part of the network and on the added traffic only.

3.3. Protection Degree Analysis Procedure.

For this function as well as for the next one (Spare Network Design) it is necessary to introduce the concept of working network and spare network. The transmission network can be considered as composed of two logically separate networks. The first one, the working network, consists of the bearers normally used to carry the traffic. The second one, the spare network, is composed of the remaining bearers. So, the spare transmission resources are distributed over the network and are at disposal of the protection mechanisms.

The protection analysis procedure, also detailed in figure 5, evaluates the number of elementary traffic flows that cannot be protected (rerouted) in the occurrence of a certain failure. Thus, this procedure can be viewed as a routing procedure in which the traffic involved in the given failures has to be routed on the spare network only. Both single failures as well as a general class of failures can be handled (cfr. section 1.2). Furthermore, a preemption mechanism is provided to handle different priority levels. Namely, the working resources of lower priority cir-

cuits can be used to protect (reroute) higher priority circuits when no spare resources are available. Preempted circuits will be rerouted when their priority level will be processed.

3.4. Spare Network Design Procedure.

This module builds an optimal spare network such that a given percentage of the traffic would be protected from a certain class of failures. Figure 6 summarizes the entire procedure. The percentage of the traffic to be protected can be expressed per priority.

4. Managing Protection.

Up to now we have discussed algorithms and tools for the analysis and design of protection in transmission networks. In general, these problems can be considered independently from the way protection is actually implemented within the real network. For instance, it is possible, in principle, to carry out the reconfiguration of a transmission network manually from an operator console, following the instructions supplied by the routing module. However, the extra effort we made to design a fast reconfiguration algorithm is clearly justified only in a fully automated process. This is the situation we address in this paper.

In operational transmission networks, in fact, the reconfiguration must be performed in a very fast and effective manner after a failure occurs. That is the reason why so many different automatic reconfiguration systems have been proposed in recent years (5, 6, 9, 13). Such systems, often called Automatic Network Protection Systems (ANPS), can be defined as the set of hardware devices and software products which implement network protection strategies in an automatic way, with the goal of improving the overall availability of the network.

An ANPS should perform at least the following key functions:

1. Alarm Management and Fault Localization;
2. Network Reconfiguration.

In turn, the Network Reconfiguration function consists of:

2.1. Finding an alternative path for each circuit involved in the localized failure;
2.2. Setting up the alternative path reconfiguring the appropriate DXCs.

The effectiveness of an ANPS is widely dependent on its capability to respond in very short time to a large class of failures. For this reason, fast multi-constrained shortest path algorithms must be developed. This is the main justification for the choice of the K paths table method. This table, in fact, provides a set of pre-computed paths thus reducing the processing time required to find an alternative path.

This solution is ideally suited to a centralized ANPS architecture in which the whole transmission network is managed from a single Operations Center. Such a Center, equipped with power-

ful computing systems, will store the K paths table an will perform all the rerouting algorithms. In this case the DXCs only collect alarm messages from the network, send them to the Operations Centre and finally execute the cross-connecting commands they receive.

A different approach would have been to use a distributed ANPS architecture where the processing of the alarm messages and the preparation of the rerouting tables are performed at the various nodes. This approach was evaluated in Sect. 2.1., but was abandoned because of its heavy DXC processing requirements and of its high control traffic overhead.

A number of intermediate solutions are also possible. For example, in a hybrid centralized/distributed solution, the DXCs reconfigure the network using routing tables which are precomputed in the Operations Center and are shipped out to the DXCs after each topology update.

In general, in the choice of a particular approach many aspects should be evaluated, including the following:

- the required processing capability within the DXCs;
- the overall network survivability;
- the personnel-related costs;
- the ANPS complexity, effectiveness and manageability;
- the ANPS reconfiguration speed;
- the amount of ANPS-related service information;
- the dimension of the transmission network to be protected;
- the complexity of the routing and reconfiguration algorithms.

5. Case Study.

In this section, the application of the Network Planning Tools to a sample network is presented. Input data are reported in figure 5. They consist of:

- list of network nodes;

- description of the physical transmission network (figure 5.a);

- description of the logical transmission network (at 140 Mbit/s level, figure 5.b);

- traffic requirements to be routed on the network (figure 5.c). As shown in the figure, for each traffic requirement a priority class and the required quality class are given. Three priority classes have been used in this case. The minimum quality class was 1E-3.

Starting from these input data, we apply the routing module and obtain the results reported in figure 6. In this figure, we show the paths on which traffic requirements were routed according to the quality constraints and the capacity available in the transmission network. A multicast

traffic requirement was also routed. The used transmission capacity (i.e. working capacity) is represented in figure 7. Note that several arcs have not been used by the routing function. Their spare capacities will be used for protection.

After the execution of the routing function it is possible to analyze the degree of protection of the network. The planner is required to define a certain failure or a certain class of failures. The analysis function will simulate the occurrence of this failure and it will attempt to reroute the traffic affected by it. Figure 8 reports the results for the simultaneous failure of two links. As stated before, the degree of protection is expressed in terms of the number of non reroutable 140 Mbit/s circuits.

Finally, an example of application of the protection network design function is given in figure 9. The planner is required to specify the protection goal, that is, the class of failures the network is required to manage without loss of traffic (indeed, in some cases, loss of traffic is unavoidable due, for example, to the disconnection of a node from the network). Figure 9 reports the capacity required to survive the class of failures "single cable cut" (i.e. any single link failure). The column labelled "increment" indicates where new transmission resources are needed (positive increments) and where resources are still redundant (negative increments).

6. Conclusions.

Multi-constraint routing and protection network design are two complex and challenging problems. Here we have proposed heuristic iterative methods to find a good solution satisfying the fast response time requirements posed by the operation of a transmission network.

Future work in this area will proceed in several directions. First, exact solution techniques (for small networks) and bounds (for large networks) will be developed to evaluate the accuracy of the proposed heuristics. Then, distributed algorithms will be probed further. We believe that, in the future, DXC's will became more intelligent and powerful, and will thus be able to process sophisticated distributed algorithms. Line speeds will became higher, thus making line overhead, due to the control traffic generated by distributed algorithm, less of a critical issue. Furthermore, networks will become larger, thus will be more difficult to control from a central node (in fact, the traffic overhead at this node grows linearly with network size). Also, a distinct advantage of the distributed approach is its fault tolerance (note that in the centralized approach the protection scheme fails if the center fails or becomes disconnected from the network). In view of these consideration, we plan to re-examine the distributed solution, and in particular to study the trade offs between prompt response time, accuracy of the routing solution, and line and processing overhead.

We are also planning to explore novel techniques for the implementation of protection. In particular, we are currently investigating the possibility to apply concepts from the Compartmental Analysis (11) to the multi constrained routing problem. Preliminary results are encouraging (8), but much work needs still to be done in this direction. We are also considering the application of neural networks and, in particular Hopfield networks which have been shown to be extremely

effective in the solution of optimization problems (10).

7. References

(1) P. Asnaghi, R. Cislaghi, S. Rigobello, "Transmission Network Protection And Reconfiguration By Means of Digital Cross Connect Systems", Proceedings of the Fourth International Network Planning Symposium, Palma de Mallorca, pg. 365-372, Sept. 1989.

(2) D. Bersekas, R. Gallager, Data Networks, Prentice Hall, Inc.,, Englewood Cliffs, N.J. (1987)

(3) E.W. Dijkstra, "A Note On Two Problems in Connection with Graphs", Numer. Math., Vol. 1, pg. 269-271 (1959)

(4) M. Elzas, T. Oren, B. Zeigler (Eds.), Modeling and Simulation Methodology: Knowledge Systems Paradigms, North Holland (1989)

(5) W.E. Falconer, "Service Assurance in Modern Telecommunications Networks", IEEE Communications Magazine, pg. 32-39, June 1990

(6) T. Flanagan, "Fiber Network Survivability", IEEE Communications Magazine, pg. 46-53, June 1990

(7) L.R. Ford Jr., D.R. Fulkerson, Flows in Networks, Princeton University Press (1962)

(8) M. Garuffo, "Un Nuovo Algoritmo per la Determinazione dei Minimi Percorsi con Applicazioni alle Reti di Telecomunicazioni", (in Italian), Thesis. Catholic University - Brescia - Italy (1991)

(9) S. Hasegawa, A. Kanemasa, H. Sakaguci, R. Maruta, "Dynamic Reconfiguration of Digital Cross Connect Systems With Network Control and Management", Proceedings of the GLOBECOM 87

(10) J.J. Hopfield, D.W. Tank, "Neural Computation of Decisions in Optimization Problems" Biol. Cybern. 52, pg. 141-152 (1985)

(11) F. Kajiya, S. Kodama, H. Abe (Eds.), Compartmental Analysis, Karger, 1984

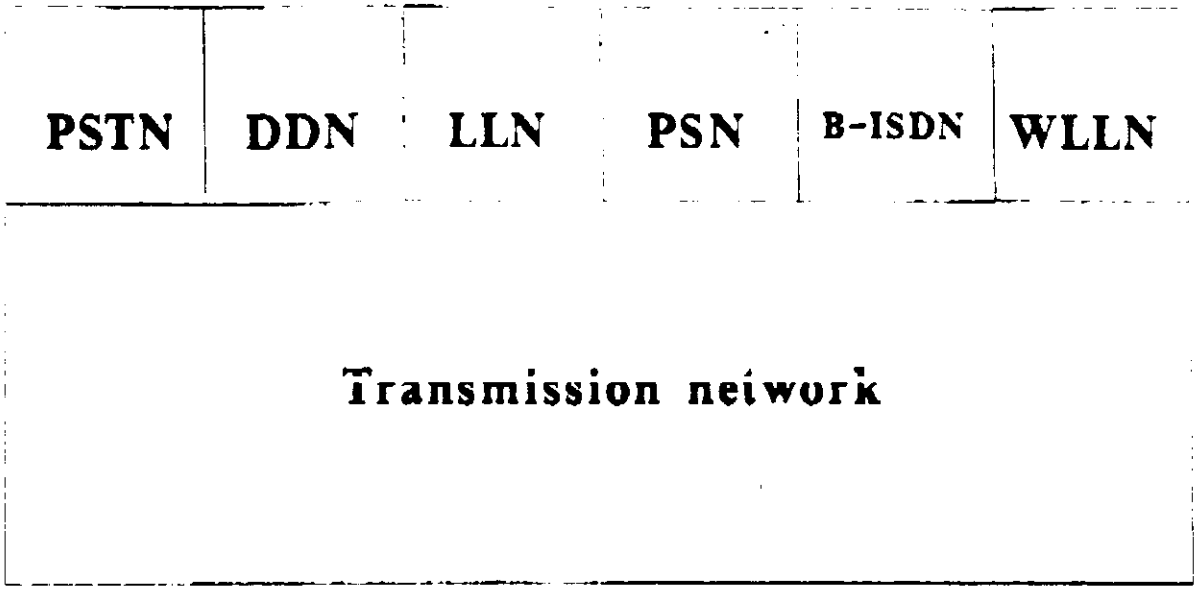
(12) F. Lenoir, M. Guiheneuf, A. Monteillet, H. Fava, "Securisation Du Reseau Interurbain De Transmission", Commutation & Transmission, N. 1, 1988

(13) C.C. Skiscim, B.L. Golden, "Computing K-Shortest Path Lengths in Euclidean Networks", Networks, Vol. 17, pg. 341-352 (1987)

(14) M. Gerla, "Routing and Flow Control in ISDNs", ICCC86 Proceedings, Munich, Germany, Sept. 1986.

(15) J. McQuillan et al, "The New Routing Algorithm for the ARPANET", IEEE Trans. on Comm., May 1980

(16) L. Fratta, M. Gerla and L. Kleinrock, "The Flow Deviation Method: An Approach to Store-and-Forward Communications Network Design", Networks, April 1973, Vol. 3, pp. 97-133



PSTN Public Switched Telephone Network

ISDN Integrated Services Digital Network

DDN Digital Data Network

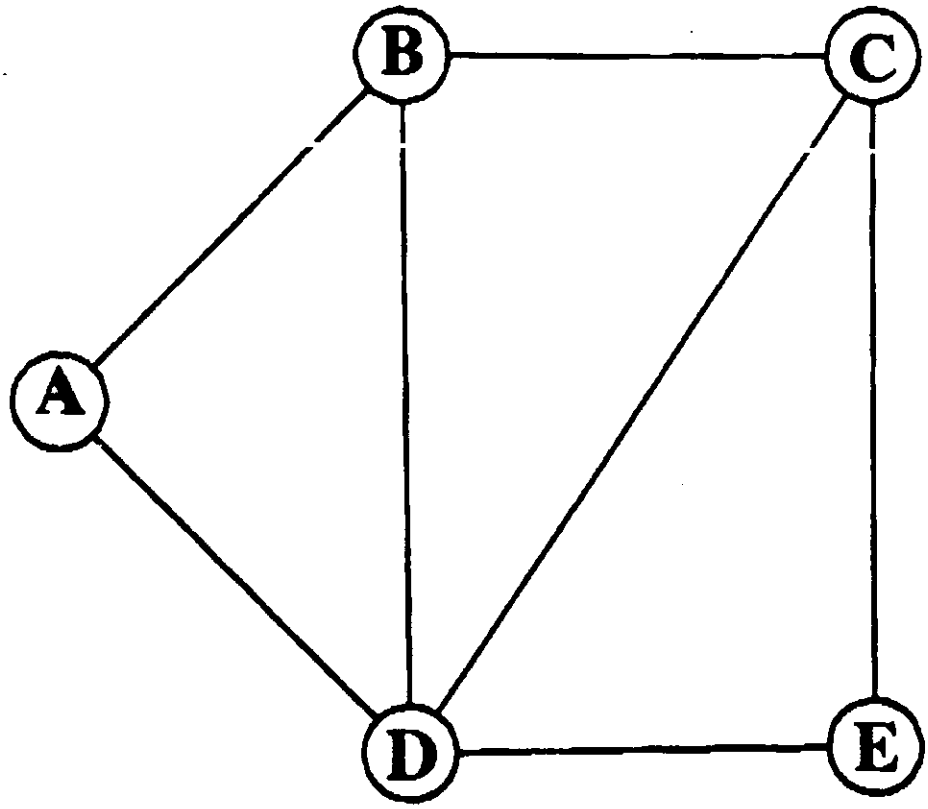
LLN Leased Line Network

PSN Packet Switched Network

B-ISDN Broadband ISDN

WLLN Wideband Leased Line Network

Figure 1: the transmission network as a transport infrastructure for multiple service networks.



...	...
A B	A B A D B A D C B A D E C B
A C	A B C A D C A D B C A D E C
...	...

Figure 2: sample network and part of its K-paths table relative to paths between nodes A, B and A,C.

FOR EACH ORIGIN / DESTINATION PAIR

FIND UP TO K BSG-DISJOINTED

ALTERNATIVE PATHS

BETWEEN ORIGIN AND DESTINATION

USING DIJKSTRA ALGORITHM

IF ONLY N<K SUCH PATHS EXIST

THEN

FIND UP TO K-N OTHER SHORTEST PATHS

(DIFFERENT FROM THE PREVIOUS N)

BETWEEN ORIGIN AND DESTINATION

USING K-SHORTEST PATHS ALGORITHM

Figure 3: summary of the K-paths table construction procedure.

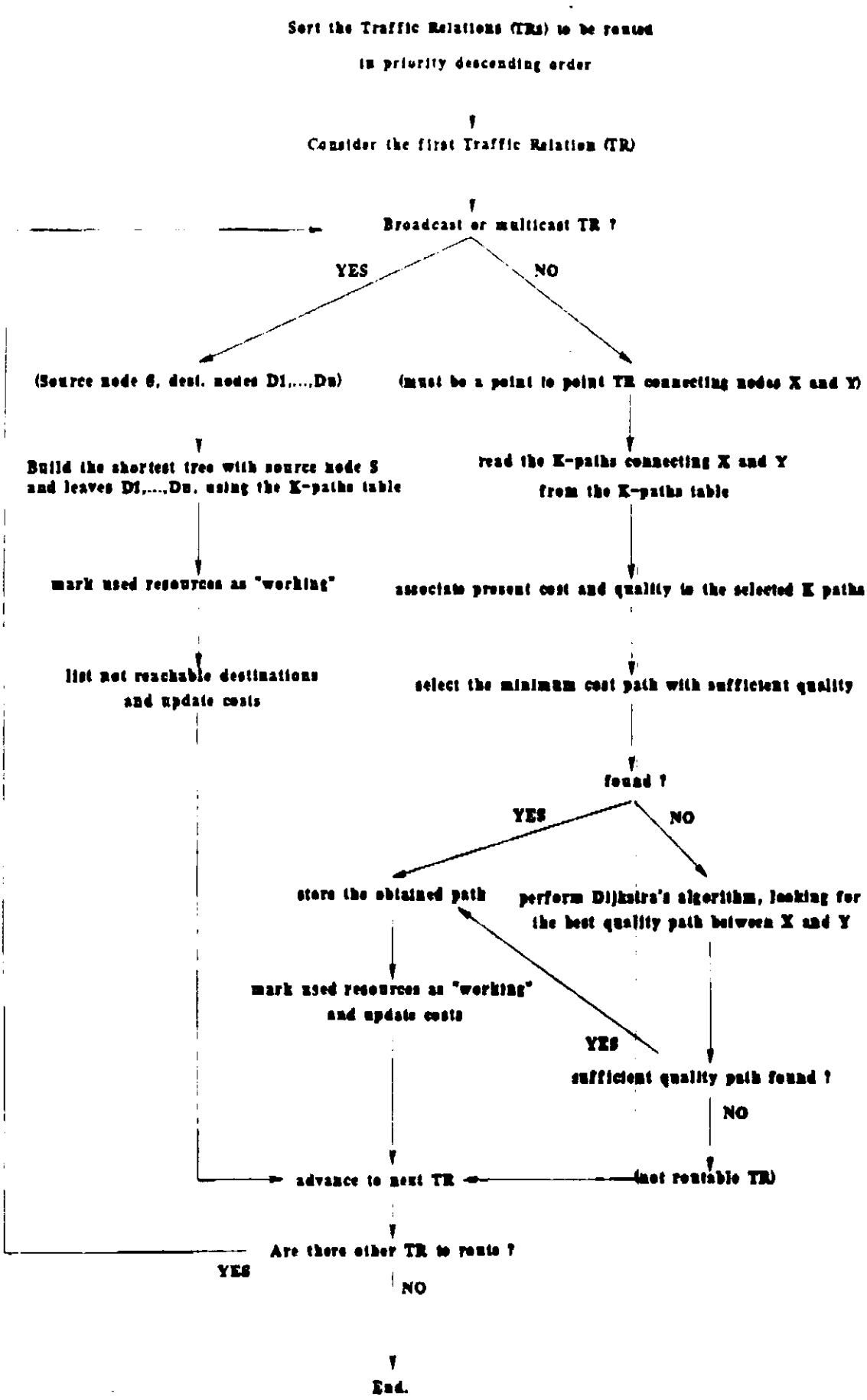


Figure 4: flow chart of the routing algorithm.

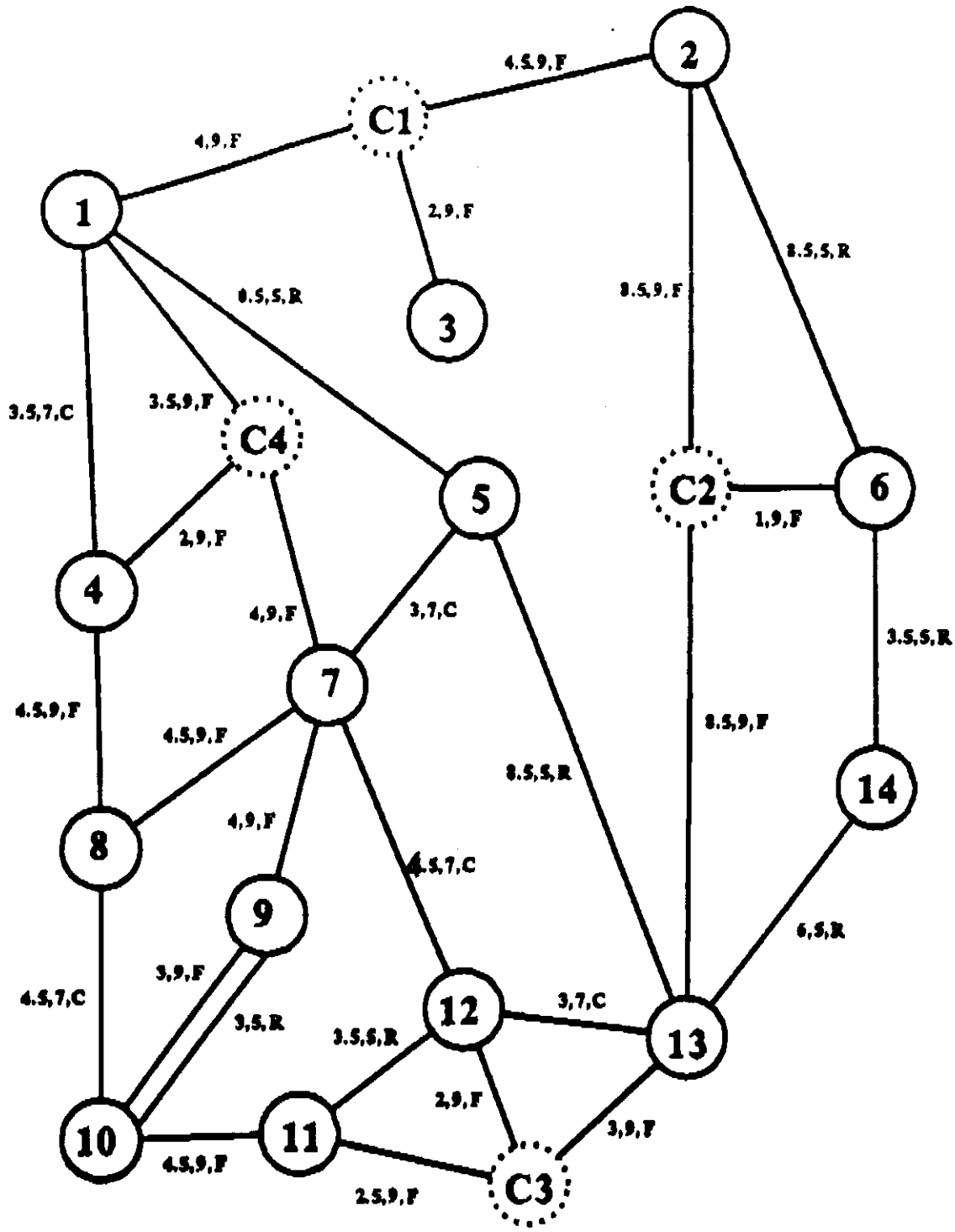


Figure 5.a: Physical transmission network

C1,C2,C3,C4 : Cable Joints

(X,Y,Z) :

X: cable length

Y: nominal transmission quality (8 means BER = 1E-8)

Z: Fibre/Radio/Coax

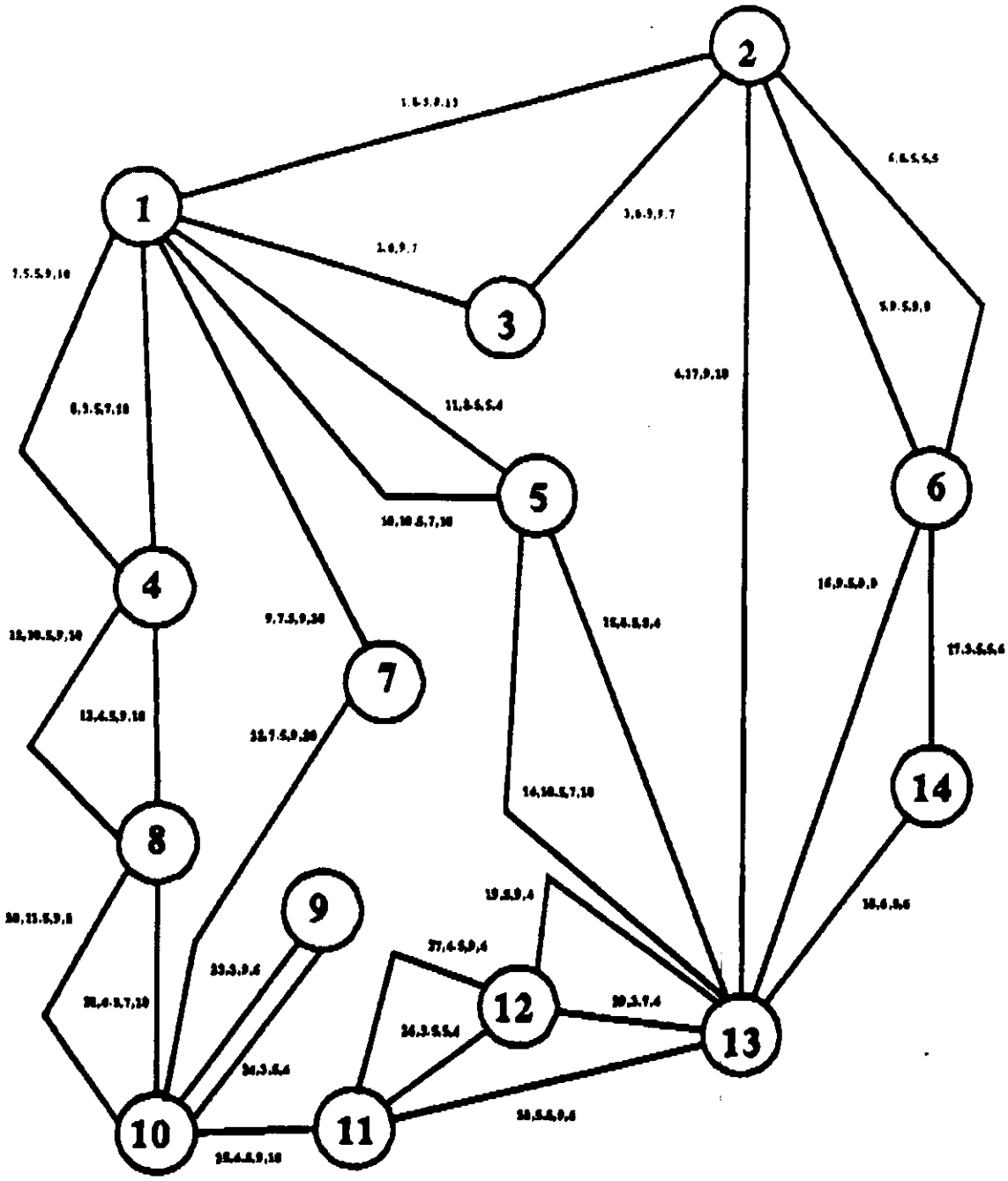


Figure 5.b: Logical transmission network

Legenda:

(W, X, Y, Z) :

W: name of the group of bearers

X: 140 Mbit/s bearer length

Y: actual transmission quality (0 means BER = 1E-8)

Z: existing capacity (number of 140 Mbit/s bearers)

TR Num.	First node	Second node	Number of 140 Mbit/s flows	Priority	Required quality (BER)
1	1	2	5	1	8
2	1	3	5	2	6
3	1	4	6	1	9
4	1	5	3	3	8
5	1	7	6	3	5
6	1	10	3	1	8
7	1	12	3	2	6
8	1	13	2	2	6
9	2	6	6	3	4
10	2	7	2	3	4
11	2	12	2	1	7
12	3	11	1	2	5
13	4	8	2	3	4
14	5	13	4	1	9
15	6	7	1	3	4
16	6	8	1	3	4
17	6	14	2	1	7
18	7	9	2	2	5
19	7	14	1	2	5
20	8	9	1	1	8
21	9	10	1	3	4
22	10	11	1	2	5
23	11	12	1	1	7
24	12	13	1	3	4
25	4	1	1	2	3
		3			
		6			
		8			
		10			
		12			

Figure 5.c: traffic relations to be routed on the network of figure 5.a and 5.b. Traffic relation number 25 is a broadcast traffic relation with source in node 4 and destinations in nodes 1,3,6,8,10 and 12.

TR num.	First node	Second node	Prior.	Number of 140 Mbit/s flows	Path (traversed arcs)
1	1	2	1	5	1
14	5	13	1	4	NOT ROUTED (insufficient quality)
6	1	10	1	3	9 22
20	8	9	1	1	20 23
17	6	14	1	2	NOT ROUTED (insufficient quality)
11	2	12	1	2	4 19
3	1	4	1	6	7
23	11	12	1	1	27
2	1	3	2	5	2
22	10	11	2	1	25
19	7	14	2	1	NOT ROUTED (insufficient quality)
18	7	9	2	2	22 23
12	3	11	2	1	2 8 13 21 25
7	1	12	2	2	8 13 21 25 27
8	1	13	2	2	10 14
7	1	12	2	1	10 14 19
15	6	7	3	1	5 1 9
21	9	10	3	1	23
4	1	5	3	3	NOT ROUTED (insufficient quality)
5	1	7	3	6	9
13	4	8	3	2	13
9	2	6	3	5	6
10	2	7	3	2	1 9
9	2	6	3	1	5
24	12	13	3	1	29
16	6	8	3	1	16 28 25 21
25	BROADCAST....		2	1	8 2 1 5 13 21 25 27

Figure 6: paths on which the given traffic has been routed by the routing algorithm. Traffic relations 14, 17, 19, 4 have not been routed because it has not been found a path satisfying their quality requirement.

Arc name	Worker capacity (140 Mbit/s)	Protection capacity (140 Mbit/s)
1	9	4
2	7	0
3	0	7
4	2	8
5	3	6
6	5	0
7	6	4
8	4	6
9	12	8
10	3	7
11	0	4
12	0	10
13	6	4
14	3	7
15	0	4
16	1	8
17	0	6
18	0	6
19	3	1
20	1	7
21	5	5
22	5	9
23	4	2
24	0	4
25	6	4
26	0	4
27	4	0
28	1	5
29	1	3

Figure 7: worker capacity used by the routing algorithm and remaining protection capacity; PROTECTION CAPACITY = EXISTING CAPACITY - WORKER CAPACITY.

IR num.	First node	Second node	Prior.	Number of 140 Mbit/s flows	Path (traversed arcs)
1	1	2	1	5	2 3
12	3	11	2	1	2 10 14 28
7	1	12	2	1	10 14 19
7	1	12	2	1	10 14 28 27
2	1	3	2	1	10 14 4 3
22	10	11	2	1	22 9 10 14 28
2	1	3	2	4	NOT RE-ROUTED
16	6	8	3	1	17 18 15 11 9 22 20
15	6	7	3	1	16 15 11 9
10	2	7	3	1	4 15 11 9
10	2	7	3	1	4 14 10 9
25	BROADCAST....		2	1	8 2 13 21 3 5 10 14 29

Figure 8: analysis of the protection degree.
The considered failures were: arc 1 and arc 25 simultaneously
The percentages of not protected traffic are the following:
Priority 1 0 %
Priority 2 80 % (due to the preemption mechanism)
Priority 3 0 %
Protection paths for the protected traffic are described in the table above.

Arc name	Existing protection capacity (140 Mbit/s)	Required protection capacity (140 Mbit/s)	Increment
1	4	9	+5
2	0	0	0
3	7	6	-1
4	8	9	+1
5	6	5	-1
6	0	0	0
7	4	3	-1
8	6	11	+5
9	8	6	-2
10	7	7	0
11	4	2	-2
12	10	8	-2
13	4	9	+5
14	7	7	0
15	4	2	-2
16	8	6	-2
17	6	3	-3
18	6	3	-3
19	1	1	0
20	7	7	0
21	5	10	+5
22	9	9	0
23	2	0	-2
24	4	4	0
25	4	13	+9
26	4	1	-3
27	0	2	+2
28	5	12	+7
29	3	6	+3

Figure 9: protection network design; required capacity for the protection network to protect the traffic against any single cable failure.