OPTIMIZATION MODELLING IN MULTILAYER NETWORKS USING SQL AND OPLSCRIPTS

Leandro A. Rodrigues*, Saulo Bortolon**, Anilton Salles**
*Graduate Program in Electrical Engineering, **Department of Informatics
UFES – C.P. 01-9011, CEP: 29060-970, Vitória, ES – Brazil
Tel: +55 021 27-3352644, Fax: +55 021 27-3352737
lar@inf.ufes.br, bortolon@inf.ufes.br, anilton@inf.ufes.br

ABSTRACT

Multi-layer telecommunication networks are becoming the predominant transport architecture. T1/E1 circuits, SDH lower and higher layers, wavelength division multiplexing (WDM), optical fiber pairs and cables are all stacked one over the other. This paper proposes a relational database scheme to represent all these layers in such a way that makes it easier to define and build optimization design models. Two relevant optimization models are described: fiber pair routing and VC-over-STM routing. Multicommodity network flow models are built using OPLScript. The first problem is modelled using an arc-path and the second one, using a node-arc formulation. These formulations are mapped onto the proposed database.

Keywords: Database, Telecom Planning, Optimization

1. INTRODUCTION

The layering concept has been extensively used as a partitioning approach in telecommunications. A client-server layering approach may be used to describe the relationship among streets, trenches, fiber cables, fiber pairs, multiplexed wavelengths, and so forth. This paper describes such a relationship in a data model using Entity-Relationship scheme. It also builds optimization models using OPLScripts and solve planning problems in metropolitan transport network, getting data from that database. OPLScript is an optimization modeling language designed by ILOG. All optimization models are built upon a basic multicommodity flow network design problem, that is an usual model in telecommunications planning.

There are many questions to be answered by network planners when designing such a network: (1) ***How many fibers are to be deployed linking central offices A and B?*** (2) ***Which cables should be used to route these fibers?*** (3) ***Do they have spare fibers to be used?*** (4) ***New cables should be deployed?*** (5) ***Working and protection fiber pairs are routed through different cables?***

Figure 1 shows candidate SDH higher-order rings posing as possible ways to connect lower-order cross-connects (SDxC 4/1). There is a VC-4 demand matrix from SDxC-to-SDxC that is to be routed over the rings. These rings may be STM-1, -4, -16 or even STM-64, indicating how many VC-4 each one may carry. Network planners have to choose which rings are to be built in order to get minimum cost and minimum inter-ring traffic [2].

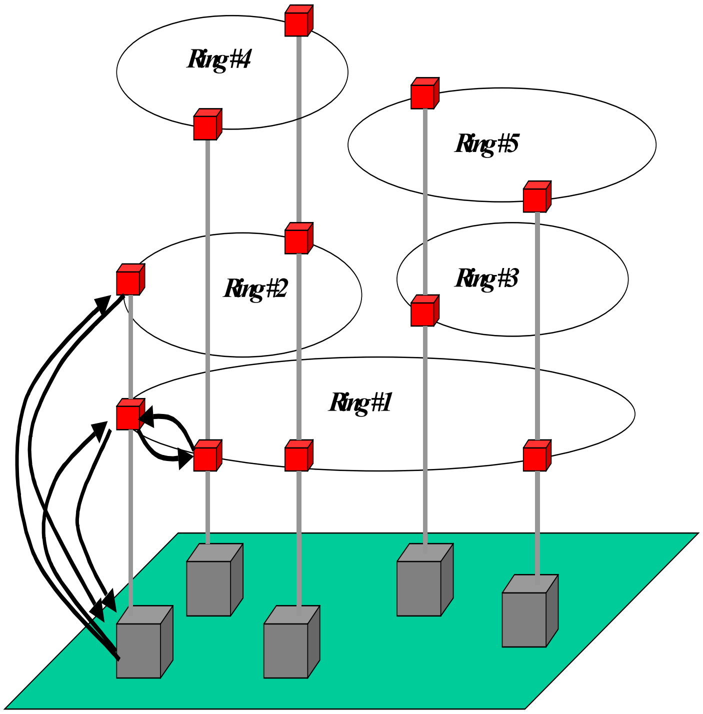

Figure 1. A SDH ring proposal.

All these questions may be answered with the help of mixed integer linear programming models, with hundreds or even thousands of variables. Two challenges are posed to telecommunications operators: (1) to keep up-to-date data about installed facilities and, (2) to have time enough to build these large optimization models. In order to help solve this challenges, this paper proposes that all information from an existent database and use an optimization programming language to build design models. This paper presents such models in OPLScript and apply to both network design problems described above.

2. MULTILAYER TRANSPORT NETWORK

From technological viewpoint, the transport network is built upon several different layers and each of these layers has their own nodes and arcs. There is a client-server (named client-support in this paper) relationship between the adjacent layers, for example: several fiber pairs use one fiber cable, a WDM cross-connect is located in a central office, and so forth.

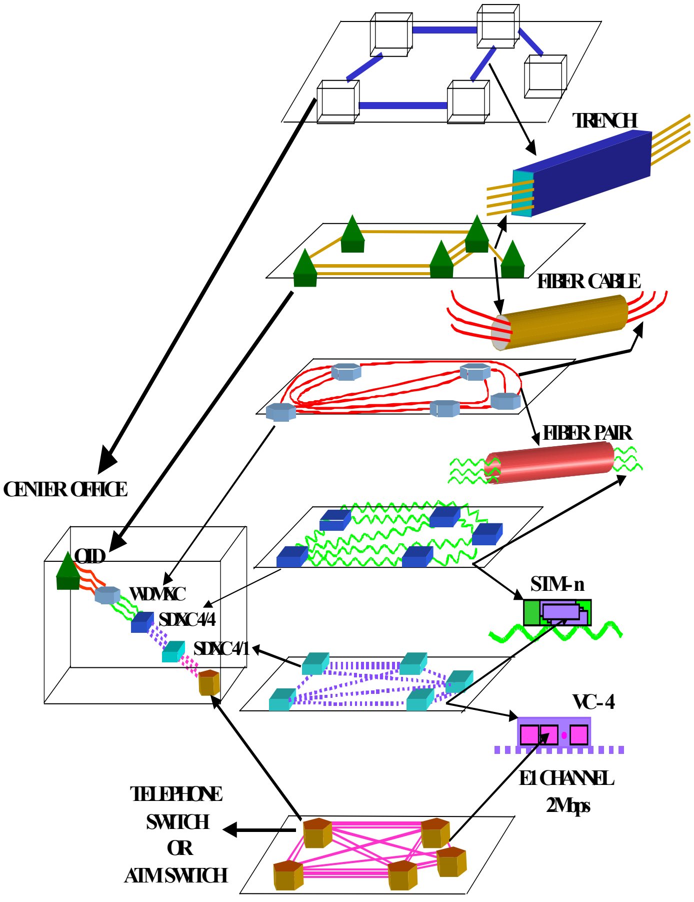

Figure 2. Multilayer transport network: client-layer is routed through its support-layer.

In a large scale network, the digitization process allows the usage of this layers with significative scale economy, although leading to serious survivability problems [3]. Different decisions, that need to be took by the planners are stated - is important to note that decisions upon one layer depends on decisions took on another one, as follows: (1) ***Where should be installed new nodes in each layer of the network and what is the capacity of them?*** When a designer talks about nodes, he may be talking about a Central Office (CO) or an Optical Intermediary Device (OID) or a WDM Cross-Connect (OxC) or a SDH Cross Connect (SDxC) or an Add-Drop Multiplexer (ADM) or an Optical Line Terminal Multiplexer (OLTM) or even, an IP switch. Each of those is to be located and dimensioned, in accordance to the design decisions; (2) ***Where should be installed new arcs in each layer of the network and what is the capacity of them?*** According to the layer of the network, an arc may be a trench or a fiber cable or a fiber pair or a WDM link carrying an STM-n signal (Synchronous Transport Module) or a VC-4 container (SDH Virtual Container) or a VC-12 or a 64Kbps link between two central offices. Given the nodes of the network in each layer, links connecting them have to be designed. Each link has its cost and capacity as well; (3) ***How should be assigned routes from a client-layer through its support-layer?*** This may be translated as (3.1) How should be carried a 64Kbps link between two central offices, routing them through the T1/E1 channels? (3.2) How should be routed the T1/E1 channels through the VC-4 containers? (3.3) How should be routed the VC-4 containers through the STM-n's?, and so on; up to the routing of the fiber cables through the trenches network. In all those decisions, the planner is looking for a manageable, efficient, resilient and low cost solution. Some circuits or some part of the network may demand an extra level of survivability due to special customer requirements.

Thus, the transport network has layered structure with links from one layer multiplexed and routed into links of the adjacent layer. That is, a support-layer is used to route demands from its client-layer (e.g., trenches are used to route a bunch of fiber cables and fiber cables are used to route the fiber pairs).

This layered structure is shown in Figure 2. The right side of the figure shows how arcs of the client-layer are routed through arcs of its support-layer. The left side shows how the switching nodes of adjacent layers are connected within a CO (implementing connections among nodes of different layers). Note that these links connect nodes on the same layer or on different layers. In the first case, the pair of nodes is located at different COs; in the second case, the pair of nodes is located at the same CO.

Each node and each arc of this multilayer network have cost, capacity and modularity associated with it. Capacity is usually based on standards: an optical fiber cable may carry up to 8, 16 or 32 fiber pairs; an SDxC device may switch up to 64, 256 or 1024 E1 channels. Cost structure have linear, fixed-charge and modular components to be considered: fiber cable installation costs are linear with its length; a CO has a fixed-charge cost to be built; SDxC switches may have its capacity upgraded in modules.

For reliability reasons, there might also be requirements of link-disjoint routing, taking many layers into account simultaneously. Two fiber pairs (working and protection) used by APS (Automatic Protection Switching) [13] must have its protection fiber routed through cables and trenches diverging from its working fiber pair. A STM-16 ring protected by line switching (bi-directional ring) must have all its backup STM-16 modules routed through wavelengths using diverse fiber pairs, diverse fiber cables and diverse trenches.

All those network requirements and all those multi-layered routing procedures demand a computational planning tool with graphical user interfaces and plug-in/ready-to-use network flow modeling. Both these computational facilities are data-intensive applications that should be connected to a network planning database. Such database scheme will be described at next section.

3. A DATABASE SCHEME FOR NETWORK PLANNING

In order to represent all features described at previous section, this paper proposes the database scheme shown in Figure 3. Looking at this entity-relationship (E-R) scheme, one mat say that each layer in Figure 2 has a correspondent “data layer” in Figure 3. In addition, Figure 4 depicts the lower order SDH layer highlighted in Figure 3 [9].

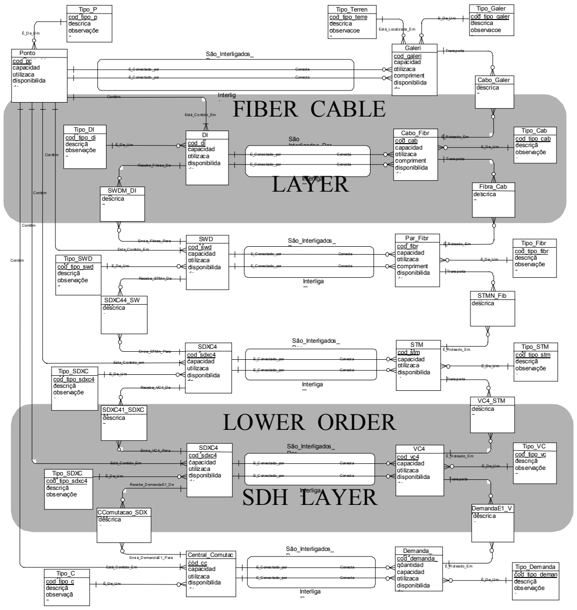

Figure 3. Multilayer network database scheme.

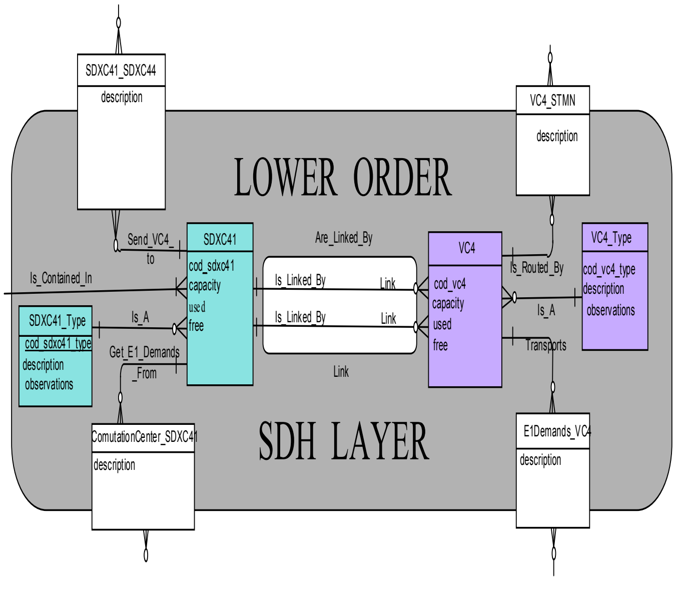

Figure 4. Database scheme of lower order SDH layer - extracted from Figure 4.

The Figure 4 describe that:

- Each SDxC 4/1 origins one or more VC-4 and each VC-4 connects two SDxC 4/1;
- Each VC-4 is routed by a sequence of one or more STM-n (its higher order path) and each STM-n routes up to n VC-4 (its payload);
- Each VC-4 may be sent from an origin SDxC 4/1 to a higher order SDxC 4/4;
- Each E1 may be sent from an origin voice switch to an SDxC 4/1;
- Each E1 is routed by a sequence of one or more VC-4 (its lower order path) and each VC-4 routes up to 63 E1 (its payload); and
- Each SDxC 4/1 is located in a central office building.

All those relations have their counterparts in each level shown in Figure 2. The advantage of such scheme is its capability to incorporate new layers whenever needed. A network planner may include ATM or IP layers in the present model with little changes in other layers.

4. MULTICOMMODITY NETWORK FLOW MODELS

The use of optimization models in telecommunications is an industry practice for interoffice network planning since [6]. In multi-layered networks, all nodes and arcs have associated costs and capacities. There are many useful optimization models, as follows:

- How many fiber pairs should be installed to connect COS?, where should new cables be placed, if any?, and how should be connected existing spare fiber pairs extremities in OID's, in order to build optical paths from origin-CO to destination-CO?
- How many VC-4s should be installed to connect SDxC 4/1?, which rings should be installed, if any?, and how should be routed VC-4s over them in order to minimize the overall cost and the inter-ring connections?

Those and many other interoffice network design optimization models are based on multicommodity network flow problem formulations [1][7]. There are two possible approaches to multicommodity flow modeling: arc-path and node-arc. The first one require that a set of possible candidate paths be described prior to decide how to route the demand over them. The second one allows all possible paths to be used, regardless of which routes they use. Figure 5 describes a multicommodity network sample that transports k = 3 commodities throughout n = 6 nodes using m = 10 arcs. Each arc has its capacity and its cost to install, and each commodity has a demanded volume.

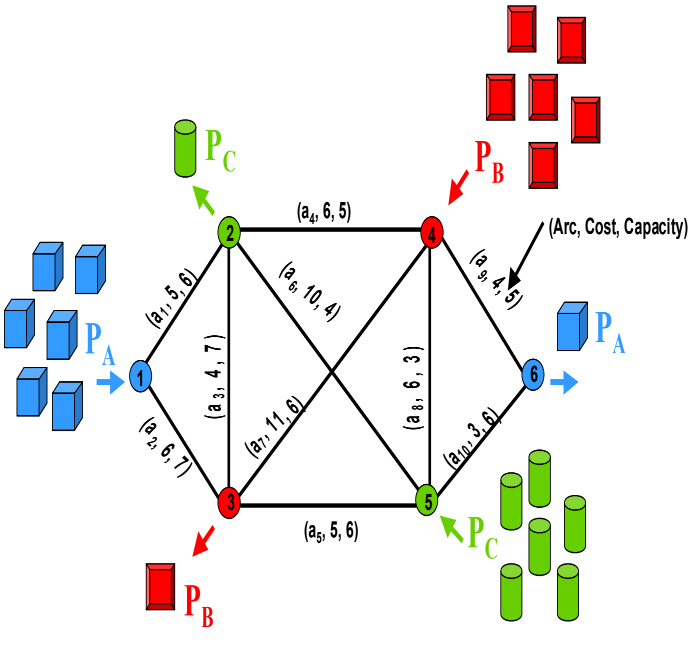

Figure 5. Multicommodity flow network sample.

The following arc-path mathematical model described hereafter was extracted from[7]:

$$\text{Minimize} \qquad {}_{j} C^{j} x_{j}$$

subject to:

$$ {}_{k} \quad {}_{r} a_{j}^{r,k} \cdot f_{r,k} \leq x_{j} b_{j} \quad \forall\, j = 1, \ldots, m \qquad (1)$$

$$ {}_{r} f_{r,k} = d_{k} \quad \forall\, k = 1, \ldots, p \qquad (2)$$

$$ f_{r,k} \geq 0 \quad \forall\, r = 1, \ldots, n_{k}, k = 1, \ldots, p \qquad (3)$$

$$ x_{j} = 0/1 $$

where:

$$ a_{i}^{j,k} = \begin{cases} 1, & \text{if arc } i \text{ belongs to route } R_{j}^{k} \\ 0, & \text{otherwise} \end{cases} $$

R_{j}^{k} route *j*, used only by commodity *j*

p is the number of commodities to be routed

$x_{\mathrm{j,\,k}}$ is the amount of commodity *k* passing by arc *j*

$C^{j,k}$ is the cost to flow one unit of commodity *k* by arc j.

b_i is the maximum flow through arc *i*.

d_k is the total amount of commodity *k* to be transported.

Constraints 1, 2 and 3 may be read as:

(1) Choose arcs that will be build, in order to minimize total arc costs;

(2) Each arc $\mathrm{a_i}$, *i* = 1, ..., *m*, must have its flow equal to or less than b_i; and

(3) Each commodity *k* must have all its d_k units transported from origin and destination nodes.

An alternative node-arc formulation is:

$$\text{Minimize} \qquad {}_{k} \quad {}_{j} C^{j} x_{j}$$

subject to:

$$ {}_{j} a_{j}^{i} g_{j,k} = d_{i,k} \quad \forall\, i = 1, \ldots, n, \forall\, k = 1, \ldots, p \qquad (4)$$

$$ {}_{k} g_{j,k} \leq b_{j} \quad \forall\, j = 1, \ldots, m \qquad (5)$$

$$ g_{j,k} \geq 0 \quad \forall\, j = 1, \ldots, n, k = 1, \ldots, p \qquad (6)$$

$$ x_{j} = 0/1 $$

where:

$$ a_{i}^{j} = \begin{cases} 1, & \text{if arc } j \text{ starts at node } i \\ -1, & \text{if arc } j \text{ ends at node } i \\ 0, & \text{otherwise} \end{cases} $$

p is the number of commodities to be routed

$g_{\mathrm{j,\,k}}$ is the amount of commodity *k* passing by arc *j*

$C^{j,k}$ is the cost to transport one unit of commodity *k* through arc j

b_i is the maximum flow through arc *i*

d_k is the total amount of commodity *k* to be transported.

Constraints 4, 5 and 6 may be read as:

(4) Choose arcs that will be build, in order to minimize total arc costs;

(5) Each arc a_i, *i* = 1, ..., *m*, must have its flow equal to or less than b_i; and

(6) Each commodity *k* must have all its d_k units transported from origin and destination nodes.

5. ROUTING FIBERS OVER CABLES

The fiber-routing problem may be modelled as an arc-path multicommodity problem following the table below:

Telecom Network	Modelled Network
Fiber pair	Commodities
Fiber cable	Arcs
Optical Intermediary Device	Nodes

An OPLScript program that translates the arc-path mathematical model described at section 4 is presented in Figure 6. Note that the script uses database connections in order to gather data from a real network and to update the network design with the optimization output. This script generates candidate routes with less than 2 fiber melting points at intermediate OIDs. This is an attempt to generate minimum-hop routes.

```
/* Database structures */
struct Node {int+ node;int+ capacity;int+ utilization;};
struct Arc {int+ arc;int+ capacity;int+ utilization;int+ length;
            int+ oid1;int+ oid2;float+ cost;};
struct Fiber {int+ fiber;int+ length;int+ oid1;int+ oid2;};
struct Fiber_Cable {int+ fiber;int+ cable;};
/* Artificial structures */
struct Demand {int+ demand;int+ oid1;int+ oid2;
               int+ volume;};
struct Route {int+ demand;int+ capacity;};
struct Route_Arc {int+ route;int+ arc;int+ demand;};
/* Database connection */
DBconnection bdNet("mssql", "sa//bdNet/TRUTA");
{Node} node from DBread(bdNet, "select cd_oid,capacity_oid,
                        utilization_oid from oid order by cd_oid ");
{Arc} arc from DBread(bdNet,
"select cd_cable,capacity_cable,utilization_cable,length_cable,
cd_oid_1,cd_oid_2,vl_cost from Cable order by cd_cable ");
DBexecute(bdNet, "create table
#demand(demand integer identity,oid1 integer NOT NULL,
    oid2    integer NOT NULL,volume  integer NOT NULL) ");
DBexecute(bdNet, "insert into #demand (oid1,oid2,volume)
            select SWDM_DIO1.cd_oid,SWDM_DIO2.cd_oid,
count(Fiber.cd_fiber) from Fiber,SWDM_DIO SWDM_DIO1,
                        SWDM_DIO SWDM_DIO2
where Fiber.cd_swdm_1 = SWDM_DIO1.cd_swdm and
      Fiber.cd_swdm_2 = SWDM_DIO2.cd_swdm
group by SWDM_DIO1.cd_oid,SWDM_DIO2.cd_oid
order by SWDM_DIO1.cd_oid,SWDM_DIO2.cd_oid ");
```

```
{Demand} demand from DBread(bdNet,
                              "select * from #demand ");
DBexecute(bdNet, "drop table #demand ");
DBexecute(bdNet, "create table #fiber(
fiber integer nodeT NULL,length integer nodeT NULL,
oid1 integer nodeT NULL,oid2 integer nodeT NULL) ");
DBexecute(bdNet, "insert into #fiber (fiber,length,oid1,oid2)
Select Fiber.cd_fiber,Fiber.length_fiber,SWDM_DIO1.cd_oid,
SWDM_DIO2.cd_oid from Fiber,SWDM_DIO SWDM_DIO1,
SWDM_DIO SWDM_DIO2
where Fiber.cd_swdm_1 = SWDM_DIO1.cd_swdm and
      Fiber.cd_swdm_2 = SWDM_DIO2.cd_swdm
order by Fiber.cd_fiber ");
{Fiber} fiber from DBread(bdNet, "select * from #fiber ");
DBexecute(bdNet, "drop table #fiber ");
{Fiber_Cable} fiber_cable from DBread(bdNet,
"select cd_fiber,cd_cable
from Fiber_Cable where st_fiber = 'P'
order by cd_cable ");
/* Routes generation */
int+ routes_por_demand = 3;
// (direct path + 1 hop + 2 hops) = 3 routes
int+ max_routes = routes_por_demand*card(demand);
int+ arcs_por_demand = 6;
// (direct path + 1 hop + 2 hops) = 6 arcs
int+ max_route_arcs = arcs_por_demand*card(demand);
range qtd_routes 1..max_routes;
Route  route[qtd_routes]; // Each demand has 3 possible routes
range     qtd_route_arcs 1..max_route_arcs;
Route_Arc route_arc[qtd_route_arcs];
// All arcs that compose routes
int+ ind_route = 1;
int+ ind_route_arc = 1;
int+ flag = 1;
int+ oid_org = 0; int+ oid_dst = 0;
initialize {
  forall(ra in qtd_route_arcs) {route_arc[ra].route = 0;
         route_arc[ra].arc = 0;route_arc[ra].demand = 0;};
  forall(ordered d in demand) {
    forall(i in 1..routes_por_demand) {
      ind_route = i + routes_por_demand*(d.demand - 1);
      route[ind_route].demand    = d.demand;
      route[ind_route].capacity = 1000;
     }; // <demand,capacity>
    forall(i in 1..arcs_por_demand) {
      // Escolhe as routes dos arcs
      if i = 1 ∨ i = 2 then ind_route = i +
                       routes_por_demand*(d.demand - 1) endif;
      if i = 4 then ind_route = (i-1) +
                       routes_por_demand*(d.demand - 1) endif;
      // Choosing arcs into routes
      if i = 1 then
        forall(a in arc: ((d.oid1 = a.oid1 & d.oid2 = a.oid2) ∨
                       (d.oid1 = a.oid2 & d.oid2 = a.oid1))){
          route_arc[ind_route_arc].route    = ind_route;
          route_arc[ind_route_arc].arc     = a.arc;
          route_arc[ind_route_arc].demand = d.demand;
          ind_route_arc = ind_route_arc + 1; }
     endif; // direct arc
};//Create"max_routes"routes and "max_route_arcs"route_arcs};
/* Functions */
int f_cap_route[i in qtd_routes] = route[i].capacity;
int f_cap_arc[a in arc] = a.capacity-a.utilization;
{int} f_fibers_da_demand[d in demand] =
```

```
          {f.fiber | f in fiber : d.oid1 = f.oid1 & d.oid2 = f.oid2};
/* Decision variables */
var int x[arc] in 0..1,
      int routeflow[i in qtd_routes] in 0..f_cap_route[i];
/* Mixed integer-linear programming model */
minimize sum(a in arc) a.length * a.cost * x[a]
subject to {
  forall(d in demand) // Demand satisfaction
    sum(i in qtd_routes : d.demand = route[i].demand)
      routeflow[i] = d.volume;
  forall(a in arc) // Arc capacities
    sum(i in qtd_route_arcs : a.arc = route_arc[i].arc)
      routeflow[route_arc[i].route] <= f_cap_arc[a] * x[a];};
/* Database update -> Fiber_Cable table */
{Fiber_Cable} proc_routing = {<f.fiber,a.arc> | a in arc & f in
fiber & ra in qtd_route_arcs & d in demand :
               routeflow[route_arc[ra].route] > 0 &
               route_arc[ra].demand = d.demand &
               f.fiber in f_fibers_da_demand[d] &
               f.fiber < first(f_fibers_da_demand[d]) +
               routeflow[route_arc[ra].route] &
               route_arc[ra].arc = a.arc & x[a] = 1 };
DBupdate(bdNet, "delete from Fiber_Cable where cd_fiber = ?
and cd_cable = ? ")(fiber_cable);
DBupdate(bdNet, "insert into fiber_Cable(cd_fiber,cd_cable,
st_fiber) values (?,?,'P') ")(proc_routing );
```

Figure 6. The arc-path OPLScript sample.

The route generation algorithm uses an enumerative scheme to generate all routes (only one arc or two arcs or, at most, 3 arcs) to connect origin and destination nodes of each demand. This algorithm is not completely shown in the present paper because of its size. Only direct paths generation is shown.

6. ROUTING VC-4s OVER STM-n RINGS

This problem may be modeled as a node-arc multicommodity problem, as presented by the following table:

Telecom Network	Modelled Network
VC-4	Commodities
STM-n	Arcs
SDxC 4/4	Nodes

An OPLScript program that translates the node-arc mathematical model described at section 4 is presented in Figure 7. Note that the script uses database connections in order to gather data from a real network.

```
/* Database structures */
struct Node {int+ node;int+ capacity;int+ utilization;};
struct Arc {int+ arc;int+ capacity;int+ utilization;
             int+ org;int+ dst;float+ cost;};
/* Artificial structures */
struct Demand {int+ demand;int+ org;int+ dst;int+ volume;};
struct Ring {int+ ring;int+ capacity;int+ utilization;};
struct Ring_Arc {int+ ring;int+ arc;};
/* Database connection */
DBconnection bdNet("mssql", "sa//bdNet/TRUTA");
{Node} node from DBread(bdNet,
      "select cd_sdxc44,capacity_sdxc44,utilization_sdxc44
       from SDXC44 order by cd_sdxc44 ");
DBexecute(bdNet, "create table #STMN(node_seqn_stmn
integer identity,cd_stmn integer NOT NULL,
capacity_stmn smallint NOT NULL, utilization_stmn smallint
NOT NULL, cd_sdxc44_1 integer NOT NULL,cd_sdxc44_2
integer  NOT NULL,vl_cost float NULL,id_status char(1)
NULL) ");
DBexecute(bdNet, "create table #RING45_STMN(cd_ring45
          integer  NOT NULL,cd_stmn   integer  NOT NULL) ");
DBexecute(bdNet, "execute sp_load_arc ");
{Arc} arc from DBread(bdNet, "select cd_stmn,capacity_stmn,
              utilization_stmn,cd_sdxc44_1,cd_sdxc44_2,vl_cost
              from #STMN ");
DBexecute(bdNet, "create table #demand(
              demand integer  identity,origin  integer  NOT NULL,
              destination integer NOT NULL,
              volume smallint NOT NULL) ");
DBexecute(bdNet,"insert into
#demand(origin,destination,volume)
select table1.cd_sdxc44,table2.cd_sdxc44,VC4.qt_demand
from VC4,SDXC41_SDXC44 table1,SDXC41_SDXC44 table2
where VC4.cd_sdxc41_1 = table1.cd_sdxc41 and
VC4.cd_sdxc41_2 = table2.cd_sdxc41 order by VC4.cd_vc4 ");
{Demand} demand from DBread(bdNet,
                                    "select * from #demand ");
{Ring} ring from DBread(bdNet,
              "select cd_ring45,capacity_ring45,utilization_ring45
              from RING45 order by cd_ring45 ");
{Ring_Arc} ring_arc from DBread(bdNet,
             "select cd_ring45,cd_stmn from #RING45_STMN ");
DBexecute(bdNet, "drop table #demand ");
DBexecute(bdNet, "drop table #RING45_STMN ");
DBexecute(bdNet, "drop table #STMN ");
/* Initializations */
int supply[node,demand];
// Indicators to represent whether a node is supply, sink, or
//intermediate
initialize {
forall(n in node&d in demand:n.node=d.org) supply[n,d] = -1;
forall(n in node&d in demand:n.node=d.dst) supply[n,d] =  1;
forall(n in node&d in demand: n.node <> d.org & n.node <>
d.dst) supply[n,d] = 0;};
/* Functions */
int+ capacity[a in arc] = a.capacity-a.utilization;
int+ volume[d in demand] = d.volume;
/* Decision variables */
var float+ traffic[arc,demand],
     int+ x[arc] in 0..1,int+ y[ring] in 0..1;
/* Mixed integer-linear programming model */
minimize
  sum(a in arc) a.cost * x[a]
subject to {
  forall(n in node & d in demand) // Flow conservation
    sum(a in arc : n.node = a.dst) traffic[a,d] -
    sum(a in arc : n.node = a.org) traffic[a,d] =
                                       supply[n,d] * volume[d];
  forall(a in arc) // Arc capacities
    sum(d in demand) traffic[a,d] <= capacity[a] * x[a];

  forall(aa in ring_arc & r in ring & a in arc :
        aa.ring = r.ring & aa.arc = a.arc) // Ring assignments
    x[a] = y[r];};
display(a in arc, d in demand : traffic[a,d] > 0.1) traffic[a,d];
display x;display y;
```

Figure 7. The node-arc OPLScript sample.

7. CONCLUSION

This paper presents data and optimization models to multilayer telecommunications network design. Based on a single database scheme and on multicommodity flow optimization models, it is shown how to find good design solutions to many network planning problems. This approach is being used as the kernel of a transport network planning & project decision support tool.

The main contribution of the proposed database/optimization-based architecture is to provide flexibility and reliability – because of the database support – and fast optimization models development. As long as this architecture is used, a library of optimization models may be built and used to many different design problems.

The next step is incorporate this “transport network planning & project decision support tool kernel” to a graphical environment in order to simplify the network designer analysis.

REFERENCES

[1] Ahuja R. K., Magnanti T. L., Orlin J. B. "Network Flows Theory - algorithms and applications", Prentice Hall, Englewood Cliffs, New Jersey 1993

[2] Bortolon S. “Planejamento Otimizado de Redes de Transporte na Hierarquia Digital Síncrona”. Dissertação de Doutorado, FEE - UNICAMP, Setembro de 1996.

[3] Garcia A. S. “Planejamento do Entroncamento em Redes Telefônicas Urbanas em Processo de Digitalização”. Dissertação de Doutorado, FEE - UNICAMP, 1987.

[4] Guerra S. “Um Modelo de Alocação de Canais E1 em Containers SDH”. Tese de Mestrado, PPGEE, Depto. de Engenharia Elétrica – UFES, Julho de 1999.

[5] Northern Telecon, “Synchronous Transmission Systems”, Nortel (Northern Telecon) Limited, 1995.

[6] Rapp Y. "Planning of Junction Network in a Multi-Exchange Area: General Principles". Ericsson Technics, vol.20, nº 1, 1964, pp. 77-130.

[7] Santos I. P. "A Primal Partitioning Approach to the Minimum Cost Multicommodity Network Flow Problem", Tese de Mestrado, PPGI, Depto. de Informática – UFES, Agosto de 1999.

[8] Silva R. P. "Uma Metodologia para Alocação de Canais E1 (2Mbps) em Virtual Containers VC-4 em Redes Metropolitanas". Tese de Mestrado, PPGEE, Depto. de Engenharia Elétrica – UFES, Dezembro de 1999.

[9] Trevisol A. G. “Desenvolvimento de uma Base de Dados para Redes Multicamadas” Monografia de Graduação, Colegiado de Ciências da Computação, Depto. de Informática – UFES, Setembro de 1999.

[10] Zanandrea, G.B; Garcia, A.S. “A Heuristic Approach for SDH Transmission Network Evolution”, IV International Conference on Telecommunication Systems, pp. 620-626, Nashville, TN, USA, Março/96

[11] Moura, A. F.; Garcia, A.S. “A Heuristic Procedure for Multilayer Network Planning”, 5th INFORMS TELECOM, Boca Raton, USA, Março/2000

[12] Pereira, J.; Garcia, A.S.; Bortolon, S. “Optimizing the Routing/Allocation of E1 into VC-4 in SDH Netwotks”, First International Workshop on the Design of Reliable Communication Network, Brugge, Belgium, pp., Maio 1998

[13] Wu, Tsong-Ho. Fiber Network Service Survivability, 1992, Artech House