Design of Reliable Communication Networks (DRCN) 2003, Banff, Alberta, Canada, October 19-22, 2003

Design models for Robust Multi-Layer Next Generation Internet core networks, carrying Elastic Traffic

Eligijus Kubilinskas[1], Michał Pióro[1,2] and Pål Nilsson[1]
[1]Department of Communication Systems, Lund University, BOX 118 SE-221 00, Lund, Sweden
[2]Institute of Telecommunications, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warszawa, Poland
Email: {eligik,mpp,paln}@telecom.lth.se

***Abstract*— This paper presents three mathematical formulations for designing robust two-layer networks carrying elastic traffic. The formulations differ by the way flow reconfiguration is performed in the case of link failures. An iterative algorithm to solve the problems is given and an extensive numerical study is provided comparing the effectiveness of the three reconfiguration approaches. The formulations can be applied for designing Next Generation Internet (NGI) core networks with two-layer IP-over-WDM structure.**

***Index Terms*— Design of robust networks, elastic traffic, fairness, Next Generation Internet, MPLS.**

I. INTRODUCTION

For historical reasons, telecommunications operators have deployed core networks composed of several resource layers, for example IP over ATM over SDH over WDM. The current trend, leading to Next Generation Internet (NGI), is to simplify this architecture in order to reduce network equipment and management costs, as well as network complexity. First of all, the two packet layers, i.e. IP and ATM, are being integrated into one resource layer based on MPLS. This leads to a single packet layer control plane instead of two. Secondly, IP packets will be transported directly over the optical WDM transport layer, enriched with a control plane. WDM will communicate with the packet layer by means of the G-MPLS[1] protocol. Hence, NGI core network will be most likely built as an IP-over-WDM network consisting of two layers: the upper IP-based packet layer equipped with IP/MPLS routers (the packet layer will be further subdivided into several MPLS sub-layers implied by the LSP hierarchy), and the lower WDM-based layer equipped with Optical Cross-Connects (OXC). The two network layers will be closely integrated. The integration and inter-working will be based on G-MPLS-like principles. Both, IP/MPLS routers and WDM OXCs will have to be G-MPLS-enabled.

Traffic routing in the IP (packet) layer will be evolving toward a constraint-based multi-path routing, based on (both in terms of bandwidth and paths) MPLS channels, reconfigurable by the routers. The amount of capacity allocated to IP links will be automatically modified (reconfigured) in order to adapt in real time to equipment or link failures, i.e. cable cuts. Reconfiguration of the IP links will be achieved by setting and releasing optical connections in the WDM layer.

Organization of the WDM (transport) layer will be based on an ASON-type architecture. ASON (Automatic Switched Optical Network) is a generic architecture (defined by the ITU-T) that will add the control/management plane to the "raw" transport plane of today's WDM network. Basic elements of the WDM layer will be optical reconfigurable connections (light-paths), interconnecting OXCs.

This paper considers two-layer (IP+WDM) NGI core network model with both layers potentially reconfigurable in a coordinated way. It has been assumed that demand volumes between Source-Destination (S-D) pairs (called demands in short) are imposed on the packet layer by elastic IP traffic and that they can consume any assigned capacity within certain bounds. Flows (bandwidth allocated to different paths of demands) in both upper (packet) layer and lower (optical) layer are potentially reconfigurable.

Three problem formulations for the two-layer network design (for flow reconfiguration in the lower layer only, in the upper layer only and in both layers simultaneously) and an algorithm to solve them are introduced in the paper. The formulations employ bandwidth allocation (among the flows realizing demands) according to Proportional Fairness (PF) rule in each considered (predefined) failure situation. Nodes and links of the lower layer are subject to failures. The resulting (logarithmic) total throughput in each failure situation (where flows are weighted by coefficients) is refered to as situation *revenue*. The revenues for the individual situations are forced to obey the Max-Min Fairness (MMF) principle. Flows assigned to demands' paths and link capacities (an uncapacitated network design problem is considered) are subject to maximization under a given budget constraint with respect to lower/upper bounds for each of the demand volumes.

II. BASIC NOTIONS

Informally, *elastic traffic* is the traffic induced by IP applications that can adapt, within certain bounds, to any volume of bandwidth assigned to them. The majority of traffic in today's Internet is approximately of this type. Several different ways exist to assign fairly bandwidth to demands' flows between each S-D pair in elastic traffic networks. The question to

[1]G-MPLS is a generic architecture (defined by the IETF), aimed at integrating control planes of adjacent resource layers.

answer is which of them is going to be used in NGI? One way for the fair bandwidth assignment is the well known Max-Min Fairness rule [1], which implies maximization of minimum bandwidth assigned to demands. Although the MMF method is the best in a pure fairness sense and has many different applications [2], it has a certain drawback when allocating bandwidth to elastic traffic: maximization of fairness decreases the total network throughput. The work of [3] shows that this problem could be alleviated if bandwidth allocated to demands is governed by the Proportional Fairness rule instead of MMF. PF implies maximization of the sum of logarithms of the total demands' flows. This method offers a trade-off between pure fairness (MMF) and Throughput Maximization, and therefore could be acceptable for both customers and operators. An effective algorithm for PF flow allocation for a single-layer robust network carrying elastic traffic is presented in [4]. This paper extends the considerations of [4] on fair networks to cover the multi-layer robust design case and provides an extensive numerical study, comparing effectiveness of restoration in different layers.

III. Network Model

A. Two-layer network

A network is modeled as an undirected graph with vertices representing nodes, and edges between vertices representing links. Links of the upper (IP) layer are labeled with e, where $e = 1, 2, \ldots, E$. Each link e is characterized by it's capacity y_e and *marginal cost* c_e, which is the cost of one capacity unit. A demand d ($d = 1, 2, \ldots, D$) between nodes of S-D is a requirement for certain amount of bandwidth, or in an elastic traffic case- a requirement for any available amount of bandwidth between lower bound h_d and upper bound H_d. D is the number of demands imposed on the upper layer. Demands d are realized by flows x_{dj}. Index $j = 1, 2, \ldots, m(d)$ labels paths for flows realizing demand d. The *total (aggregated) flow* X_d, realizing demand d, is the sum of flows assigned to all paths of the demand and is calculated as $X_d = \sum_{j=1}^{m(d)} x_{dj}$, $d = 1, 2, \ldots, D$.

Entities of the lower (WDM) layer are defined analogously as follows. The lower layer network is interconnected by optical links labeled with g of capacity u_g, where $g = 1, 2, \ldots, G$. Demands for the lower layer are the link capacities of the upper (IP) layer. Therefore demands of the lower layer are indexed with e and flows of WDM layer realizing demands e are denoted by z_{ek}. Index $k = 1, 2, \ldots, n(e)$ labels paths for flows realizing demand e. In this model all the nodes of the upper layer must exist in the lower layer as well. These nodes can be either the routers that have double functionality (they act as IP routers as well as WDM OXCs), or terminating nodes in WDM.

The two-layer network model, presented above could be easily extended to more layers, as well as the problem formulations and the algorithm presented below.

B. Failure situations

To represent network failures the notion of *failure situation* is introduced. A failure situation is a result of an event in which one or several links, nodes or any combination of links and nodes fully or partially fail. Such situation is represented by *availability coefficients* $\alpha_{gs} \in [0, 1]$, where s ($s = 1, 2, \ldots, S$) labels failure situations and S is the number of the considered situations. Availability coefficient is defined for the links of Layer 1 (WDM) only, since it is assumed that only physical links fail. $\alpha_{gs} = 0$ means that link g is totally broken (unavailable), whereas $\alpha_{gs} = 1$ implies that it is fully available. Fractional value of α_{gs} represents a partial link failure. Since availability coefficients are defined for links, to model a node failure α_{gs} has to be set to the value, representing level of the failure, for all links, that are incident to the failing node e.g., setting $\alpha_{gs} = 0$ for all links attached to a certain node, would mean that the node has completely failed. Node failure implies that demand set is reduced.

Having introduced failure situations, it's now possible to extend certain notions, presented in the earlier section, making them dependent on situations. Flows of the upper and lower layers now can be made situation-dependent and defined as x_{djs} (flow realizing demand d on path j in situation s) and z_{eks} (flow realizing demand e on path k in situation s) respectively. The total flow, realizing demand d in situation s is then defined as $X_{ds} = \sum_{j=1}^{m(d)} x_{djs}$, $d = 1, 2, \ldots, D$, $s = 1, 2, \ldots, S$. Similarly the capacity of the upper layer links can be defined as y_{es}. These new notions will be used together with the ones defined earlier (where applicable) in the problem formulations presented below.

IV. Mathematical formulations

Three mathematical problem formulations (central to this paper) for designing of robust two-layer network are presented in this section. The problem formulations presented are for flow reconfiguration in the lower layer only, in the upper layer only and in both layers simultaneously. All of them assume flow reconfiguration in the case of predefined failure situations, thus assuring network robustness to failures. This means that flows are rerouted on different paths, as well as the values of flows can be changed. In the third case reallocation is synchronized in both layers.

Given demands and predefined paths for each demand, the algorithms calculate maximum possible link capacities (uncapacitated network design problems are considered) and flows for each failure situation under the assumed budget constraint. In all the algorithms, flows in the upper layer are allocated according to the PF principle. Besides, the two latter formulations also assure that revenues are max-min fair among the situations, thus assuming kind of two-dimensional fairness.

The outcome of the algorithms is link capacities and the flow allocation for both layers in each failure situation, which then might be implemented using MPLS/G-MPLS/ASON technology.

All the algorithms presented further on use the notions introduced in the earlier sections. New constants and variables are introduced as necessary. All the variables in the formulations are continuous and non-negative, unless stated otherwise.

A. Problem RLL: flow Reconfiguration in Lower Layer

This problem formulation allows for flow reconfiguration only in the lower layer.

constants

w_d	revenue coefficient from demand d
h_d, H_d	lower and upper bound, respectively, for total flow of demand d
ψ_{edj}	$= 1$ if link e belongs to path j realizing demand d, 0 otherwise
φ_{gek}	$= 1$ if link g belongs to path k realizing demand e, 0 otherwise

variables

x_{dj}	fixed flow allocated to path j of demand d
X_d	total flow allocated to demand d
y_e	capacity of link e
z_{eks}	situation-dependent flow allocated to path k of link e in situation s
u_g	capacity of link g

Problem RLL:
objective

$$\text{maximize} \quad R = \sum_d w_d \log(X_d) \tag{1}$$

subject to

$$\sum_{j=1}^{m(d)} x_{dj} = X_d, \quad d = 1, 2, \ldots, D \tag{2}$$

$$h_d \le X_d \le H_d \quad d = 1, 2, \ldots, D \tag{3}$$

$$\sum_g c_g u_g = B \tag{4}$$

$$\sum_d \sum_j \psi_{edj} x_{dj} = y_e, \quad e = 1, 2, \ldots, E \tag{5}$$

$$\sum_k z_{eks} = y_e, \quad e = 1, 2, \ldots, E, \quad s = 1, 2, \ldots, S \tag{6}$$

$$\sum_e \sum_k \varphi_{gek} z_{eks} \le \alpha_{gs} u_g, \quad g = 1, 2, \ldots, G, \quad s = 1, 2, \ldots, S. \tag{7}$$

Objective function (1) maximizes sum of the logarithms of the total upper layer flows thus implementing their PF allocation. Total (aggregated) flows for each demand are calculated in 2 and are forced to attain values within certain bounds by constraints (3). Constraints (5) force the sums of all the flows of the upper layer x_{dj}, that are routed on paths traversing link e, to be equal to the capacity allocated for link e. Constraints (6) assure that sums of the flows of the lower layer (z_{eks}) are enough to implement capacity requirements y_e in all the predefined failure situations. Constraints (7), similarly to (5), force the sums of all the flows of the lower layer (z_{eks}), that are routed on the paths traversing link g, not to exceed the available (remaining) capacity ($\alpha_{gs} u_g$) of link g in situation s. Budget constraint (4) assures that the cost of lower layer links doesn't exceed the budget B.

Problem RLL is a Convex Problem and can be treated approximately as a Linear programming (LP) problem, using the piece-wise linear approximation discussed in section VI.

B. Problem RUL: flow Reconfiguration in Upper Layer

This problem formulation allows for flow reconfiguration only in the upper layer. RUL uses lexicographical maximization. Recall that a vector $(a_1, a_2, \ldots, a_S)$, sorted in the non-decreasing order, is lexicographically greater than a sorted vector $(b_1, b_2, \ldots, b_S)$ if and only if there exists s' such that $0 \le s' < S$, $a_s = b_s$ for $s = 1, 2, ..., s'$, and $a_{s'+1} > b_{s'+1}$ (it is possible that $a_i < b_i$ for some $i > s' + 1$).

constants

w_{ds}	revenue coefficient from demand d in situation s
h_{ds}, H_{ds}	lower and upper bound, respectively, for total flow of demand d in situation s
θ_{eks}	availability coefficient of path k realizing link e in situation s , $\theta_{eks} \in \{0, 1\}$, $\theta_{eks} = min(\alpha_{gs} : \varphi_{gek} = 1)$, where $\alpha_{gs} \in \{0, 1\}$

variables

x_{djs}	situation-dependent flow allocated to path j of demand d in situation s
X_{ds}	total flow allocated to demand d in situation s
y_{es}	capacity of link e in situation s
z_{ek}	fixed flow allocated to path k of link e

definitions

$R_s = \sum_d w_{ds} \log(X_{ds})$ is a revenue in situation s.
$\boldsymbol{R} = (R_1, R_2, \ldots, R_S)$ is a vector of revenues sorted in non-decreasing order.

Problem RUL:
objective

$$\text{maximize lexicographically} \quad \boldsymbol{R} \tag{8}$$

subject to

$$\sum_{j=1}^{m(d)} x_{djs} = X_{ds}, \quad d = 1, 2, \ldots, D, \quad s = 1, 2, \ldots, S \tag{9}$$

$$h_{ds} \le X_{ds} \le H_{ds} \quad d = 1, 2, \ldots, D, \quad s = 1, 2, \ldots, S \tag{10}$$

$$\sum_g c_g u_g = B \tag{11}$$

$$\sum_d \sum_j \psi_{edj} x_{djs} = y_{es}, \quad e = 1, 2, \ldots, E, \quad s = 1, 2, \ldots, S \tag{12}$$

$$\sum_k \theta_{eks} z_{ek} \ge y_{es}, \quad e = 1, 2, \ldots, E,$$

$$s = 1, 2, \ldots, S \quad (13)$$

$$\sum_e \sum_k \varphi_{gek} z_{ek} = u_g \ , \quad g = 1, 2, \ldots, G. \quad (14)$$

Objective function (8) assures that the problem results in lexicographically maximal (unique) solution vector of revenues. This implies the MMF allocation of revenues among situations, while in each failure situation flows are allocated in a proportionally fair way among the demands. Total (aggregated) flows for each demand in each situation are given in (9) and are forced to attain values within certain bounds by constraints (10). Constraints (12) force the sums of all the upper layer flows x_{djs}, that are routed on paths traversing link e, to be equal to the allocated capacity for link e in the situation s. Constraints (13) assure that the capacity of the upper layer links y_{es} doesn't exceed the total available (remaining) flows of the lower layer ($\theta_{eks} z_{ek}$) that implement y_{es} in each of the failure situations. Constraints (14), similarly to (12), force the sums of all the flows of the lower layer (z_{ek}), that are routed on paths traversing link g, not to exceed the capacity allocated for link g.

Problem RUL is not a mathematical programming problem (since it uses lexicographical order maximization) and must be solved by the algorithm presented in section V.

C. Problem RBL: flow Reconfiguration in Both Layers

This problem formulation allows for flow reconfiguration in the upper and lower layers simultaneously. It's the most flexible flow reconfiguration option, but also the most complicated. Like RUL, it also uses lexicographical maximization. Therefore it is not mathematical programming problem and must be solved by the algorithm presented in section V.

Problem RBL:
objective

$$\text{maximize lexicographically} \quad \boldsymbol{R} \quad (15)$$

subject to
constraints (9)-(12) and:

$$\sum_k z_{eks} = y_{es} \ , \quad e = 1, 2, \ldots, E, \quad s = 1, 2, \ldots, S \quad (16)$$

$$\sum_e \sum_k \varphi_{gek} z_{eks} \leq \alpha_{gs} u_g \ , \quad g = 1, 2, \ldots, G \quad s = 1, 2, \ldots, S. \quad (17)$$

Most of the constraints are analogous to those of problem RUL, except (16)-(17). Constraints (16) assure that sums of the lower layer flows (z_{eks}) are sufficient to implement capacities y_{es} in each of the failure situations. Constraints (17) force the sums of the lower layer flows (z_{eks}), that are routed on paths traversing link g, not to exceed the available (remaining) capacity of link g in situation s ($\alpha_{gs} u_g$).

D. Introducing modularity

The three problem formulations presented above allow flows and link capacities to be assigned any continuous non-negative values. Since in the backbone optical networks link capacities are installed in modules, all the problem formulations should be adjusted to account for the modularity. For example, problem RBL can be adjusted by modifying the constraint (17) as follows:

$$\sum_e \sum_k \varphi_{gek} z_{eks} \leq \alpha_{gs} u_g M \ , \quad g = 1, 2, \ldots, G \ , \quad s = 1, 2, \ldots, S \quad (18)$$

where:
- u_g capacity of link g in modules (integer variable)
- M size of the link capacity module.

Analogous changes can be made in other problem formulations. The modularity requirement makes the problems NP-hard. For small (and sometimes medium) size networks they can be solved using MIP (Mixed Integer Programming) solvers, equipped with Branch-and-Bound or Branch-and-Cut procedures [7].

V. Algorithm

Problem RLL is a LP problem, so it can be solved by LP solvers [7]. Problems RUL and RBL are not that simple to solve, since they involve lexicographical maximization (cf. [2]). The efficient iterative algorithm for solving RUL and RBL is given below. It is based on a general algorithm for convex lexicographical maximization introduced in [6]. The algorithm is an improved version of the MMF algorithm given for another application in [5], and is based on ideas described in [2] (see also references there).

A. Algorithm for solving RUL and RBL

Step 1:
Put $n := 0$, $Z_0 := \emptyset$, $Z_1 := \{1, 2, \ldots, S\}$, $t_s := 0$ for all s.

Step 2:
Solve the following Convex Programme:

maximize t
subject to (9)-(12) and
(13)-(14) for RUL or (16)-(17) for RBL, and

$$R_s = \sum_d w_{ds} \log(X_{ds}) = t_s \ , \quad s \in Z_0 \quad (19)$$

$$R_s = \sum_d w_{ds} \log(X_{ds}) \geq t \ , \quad s \in Z_1. \quad (20)$$

Let t^* be the optimal solution of the above task and $\lambda_s^*, s \in Z_1$ be the optimal dual variables corresponding to constraints (20).

Step 3:
Put $n := n + 1$, $Z := \{s \in Z_1 : \lambda_s^* > 0\}$ and $t_s := t^*$ for each $s \in Z$. Put $Z_0 := Z_0 \cup Z$ and $Z_1 := Z_1 \setminus Z$.

If $Z_1 = \emptyset$ then STOP. The vector $\boldsymbol{R} = (R_1, R_2, \ldots, R_S) = (t_1, t_2, \ldots, t_S)$ is the solution of the problem.

Else go to Step 2.

B. Comments

In the algorithm n is the iteration counter. Sets Z_0 and Z_1 have the following interpretation after completion of Step 3:

- Z_0 : The current set of situations for which the current bound t_s is the maximal value for R_s.
- Z_1 : The current set of situations for which it is not known if t_s is the maximal value for R_s.

The algorithm (in Step 3) uses values of dual variables corresponding to constraints (20) to check, weather values of revenues R_s can be further increased (this is possible in the case when $\lambda_s^* = 0$), or the current value $R_s = t_s$ is the maximum possible for a given situation (this is the case when $\lambda_s^* > 0$). A situation s for which value of the revenue can't be increased any further is called *blocking*. It should be noted that $\lambda_s^* = 0$ doesn't necessary mean that the value of the revenue for situation s can be further increased, as it is shown in [5], [6]. If after Step 2 the value of t^* doesn't increase (in comparison to the previous value), this means that there is one (or more) blocking situation in Z_1 that prevent this. The algorithm will automatically find such situation(s) in the next iteration, thanks to the fact, that in each iteration $\sum_{s \in Z_1} \lambda_s^* = 1$.

VI. Numerical Example

A. Linear approximation

All the Convex Problems considered in the previous sections can be converted to their approximative Linear Programming (LP) counterparts using the piece-wise linear approximation of the logarithmic function (cf. Figure 1). The LP approximation makes the problems solvable with standard LP solvers. In the numerical experiments, reported in the next section, the following approximation $G(z)$ of the logarithmic function $log(z)$ has been used:

$$G(z) = min\{F_p(z) = a_p z + b_p : p = 1, 2, ..P, \} \ . \quad (21)$$

The linear approximation consists in introducing one auxiliary variable f_{ds} and a set of P constraints corresponding to the linear pieces of approximation (21), which replace the logarithm of the flow e.g. $\log(X_{ds})$. Hence the optimization part in Step 2 of the algorithm for solving RBL becomes:

$$\begin{aligned} &\textbf{maximize} && t && (22)\\ &\textbf{subject to} && (9)-(12),(16)-(17) \text{ and} &&\\ &&& R_s = \textstyle\sum_d w_{ds} f_{ds} = t_s \ , \ s \in Z_0 && (23)\\ &&& R_s = \textstyle\sum_d w_{ds} f_{ds} \geq t \ , \ s \in Z_1 && (24)\\ &&& f_{ds} \leq a_p X_{ds} + b_p \ , \ p = 1, 2, ..P, \ , &&\\ &&& d = 1, 2, \ldots, D \ , \ s = 1, 2, \ldots, S. && (25) \end{aligned}$$

The consecutive pairs of coefficients (a_p, b_p) of the five ($P = 5$) linear pieces used for the approximation (cf. Figure 1) are as follows: (3.9805,-2.7170), (0.7774,-0.7910), (0.2380,0.3761), (0.0731,1.5649), (0.0180,2.9318).

The other problems can be modified analogously.

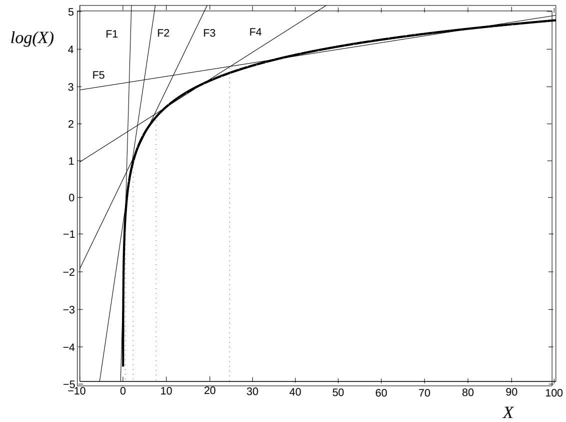

Fig. 1. Piece-wise approximation of the logarithmic function.

TABLE I
NETWORKS USED FOR EXPERIMENTS

ref. code	layer	# nodes	# links	# paths per demand	# demands	# failure situations
N_{12}	L_2	12	22	6-14	66	-
	L_1	12	18	2-3	22	19
N_{41}	L_2	21	37	6	209	-
	L_1	41	72	3	37	21

B. Example networks

A number of experiments have been performed with two different network models: mid-size (N_{12}) and large (N_{41}). The models are presented in Table I and the network topologies of both layers are shown in Figures 2-5. The aim was to find out which reconfiguration option is the most profitable for multi-layer networks in terms of lexicographically ordered revenues (that reflect total realizable throughput in each failure situation), and what is the impact of the network topology on this judgment. The affect of imposing lower and upper bounds on total flows X_{ds} has been also examined.

Links' costs for networks N_{12} and N_{41} are given in the Tables II and III respectively. Failure situations have been generated according to the following rule: in situation $s = 1$ (called the nominal situation) all links are fully available. In each of the remaining situations two randomly selected links

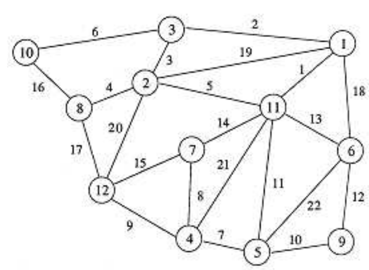

Fig. 2. Topology of the lower layer network for N_{12}.

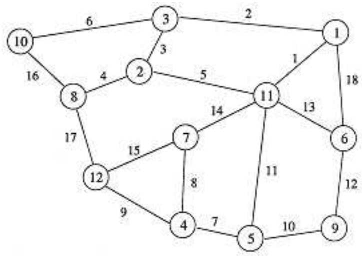

Fig. 3. Topology of the upper layer network for N_{12}.

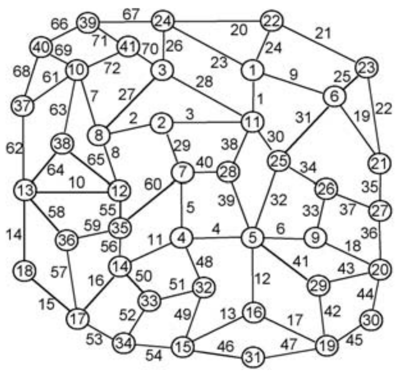

Fig. 4. Topology of the lower layer network for N_{41}.

are assumed to fail entirely, so that their link availability coefficients α_{gs} become equal to 0 (the coefficients for the remaining links are equal to 1). It has been assured that the situations are unique, and that they do not result in disjoint graphs. The pairs of links that fail in each situation are given in Tables IV and V. The experiments have been performed with $S = 19$ situations for the network N_{12} and $S = 22$ situations for the network N_{41}. For all experiments all revenue coefficients w_{ds} have been set to 1 and budget B to 10^6.

C. Numerical results

Resulting revenues of the three reconfiguration options have been compared in the unbounded (when X_d or X_{ds} could take any value between 0 and $+\infty$) and bounded (when X_d or X_{ds} could be assigned any values from the intervals $h_d \le X_d \le H_d$ or $h_{ds} \le X_{ds} \le H_{ds}$, respectively) cases. Imposing upper bound H_{ds} limits the highest value for the flows X_{ds} at a certain value, though, if allowed by the budget, it would be possible to increase it even more. In this way the resulting vector of revenues is lexicographically less than in the unbounded case. Imposing Lower Bound (LB) is of more interest, because it usually results in different flow allocation scheme. Therefore, because of the space limits, only the results for the unbounded case and with lower bounds ($LB = 1000$ for N_{12} and $LB = 10$ for N_{41}) are given. The upper bound in these experiments was always set to $+\infty$.

Revenue for problem RLL is not situation-dependent, so for different situations it is the same and equal to 6007.04 (in the unbounded case) for the network N_{12} and 2072.19 for the network N_{41}. Revenues of RUL and RBL are situation-dependent. Figures 6 and 7 illustrate lexicographically ordered revenue

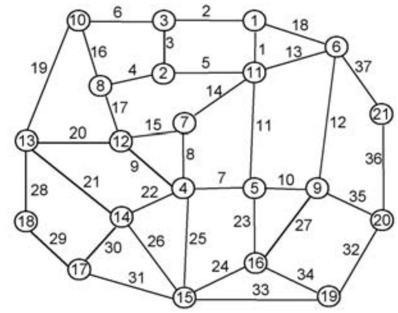

Fig. 5. Topology of the upper layer network for N_{41}.

TABLE III

LINK MARGINAL COSTS FOR NETWORK N_{41}

g	1	2	3	4	5	6	7	8	9
c_g	7.8	9.4	13.7	10.6	9.7	9.3	10.1	8.9	12.9
g	10	11	12	13	14	15	16	17	18
c_g	14.2	10.2	11.1	11.8	11.9	10.7	9.9	12.2	10.9
g	19	20	21	22	23	24	25	26	27
c_g	12.0	16.4	15.6	14.6	15.2	7.8	6.6	7.3	13.5
g	28	29	30	31	32	33	34	35	36
c_g	15.5	8.1	6.9	12.6	12.0	7.3	8.3	7.1	8.7
g	37	38	39	40	41	42	43	44	45
c_g	8.5	8.3	10.3	6.9	12.1	9.2	9.7	7.8	7.8
g	46	47	48	49	50	51	52	53	54
c_g	10.7	11.1	8.3	9.3	6.8	8.2	7.6	7.7	8.9
g	55	56	57	58	59	60	61	62	63
c_g	5.7	5.7	11.7	9.7	8.2	12.5	9.9	12.5	11.4
g	64	65	66	67	68	69	70	71	72
c_g	8.8	10.9	8.1	11.2	9.3	6.4	6.2	7.0	8.3

TABLE II

LINK MARGINAL COSTS FOR NETWORK N_{12}

g	1	2	3	4	5	6	7	8	9
c_g	1.85	3.4	1	1.45	2.3	2.9	1	1.6	2.2
g	10	11	12	13	14	15	16	17	18
c_g	1.5	2.3	1.4	1.65	1.25	2.3	1.55	1.35	1.7

TABLE IV

LINKS THAT FAIL IN EACH SITUATION FOR NETWORK N_{12}

g	1	2	3	4	5	6	7	8	9	10
s	-	5,16	8,12	6,15	3,8	11,15	6,11	3,5	2,17	4,14
g	11	12	13	14	15	16	17	18	19	
s	11,13	2,16	7.10	2,7	17,18	1,17	7,18	6,13	1,9	

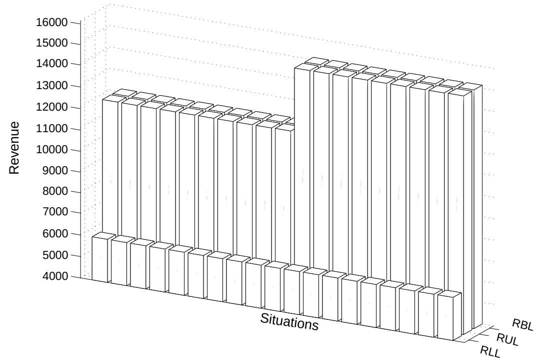

Fig. 6. Revenue values for RLL, RUL and RBL in the unbounded case (N_{12}).

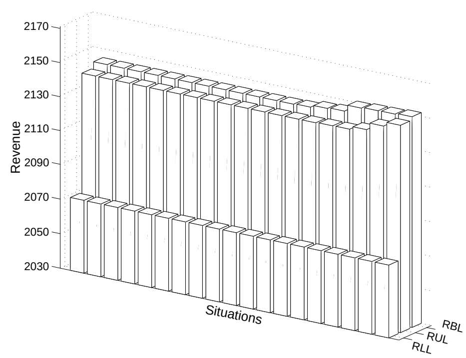

Fig. 7. Revenue values for RLL, RUL and RBL in the unbounded case (N_{41}).

vectors for the three reconfiguration options in the unbounded case for the networks N_{12} and N_{41} respectively. Because of the lexicographical ordering the numberings of situations may not coincide for different reconfiguration options. Therefore the situations are not numbered in the figures. As shown by the figures, revenue vectors for RUL and RBL are almost the same for the network N_{12}, while for N_{41} RBL is clearly better. It should be noted, that for N_{12} the differences between revenue vectors for RUL and RBL are negligible, although the vector for RBL is still lexicographically (marginally) bigger than for RUL:

$R^{RBL} = \{12132.37887293740\mathbf{8}, 12132.378872937410, 12132.378872937414, \ldots, 15132.09314945\mathbf{4997}\} >$
$R^{RUL} = \{12132.37887293740\mathbf{6}, 12132.378872937408, 12132.378872937410, \ldots, 15132.0931494\mathbf{55350}\}.$

TABLE V

LINKS THAT FAIL IN EACH SITUATION FOR NETWORK N_{41}

g	1	2	3	4	5	6	7	8	9	10	11
s	-	8,15	9,11	17,55	19,5	33,3	23,4	61,31	12,63	38,2	17,8
g	12	13	14	15	16	17	18	19	20	21	22
s	13,16	19,6	48,53	40,5	31,51	49,11	60,16	30,39	14,2	8,70	10,1

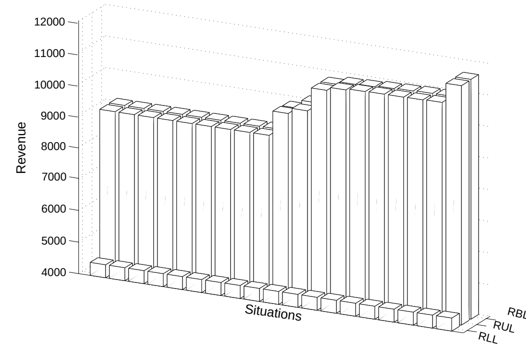

Fig. 8. Revenue values for RLL, RUL and RBL when $LB = 1000$ (N_{12}).

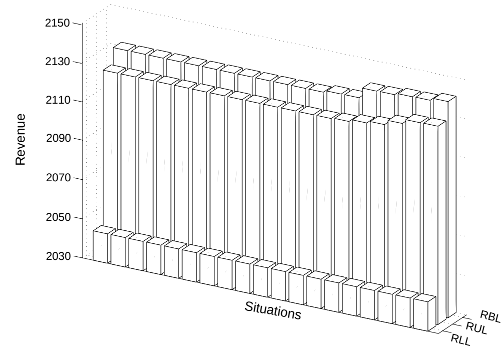

Fig. 9. Revenue values for RLL, RUL and RBL when $LB = 10$ (N_{41}).

It can be seen, that only because the smallest revenue values (in the lexicographical listing) attained for RBL are higher than the ones for RUL, it makes RBL marginally better. But the maximal revenue achieved in the RUL case is higher. This similarity of RUL and RBL for N_{12} can be explained by the very similar network topologies of the upper and lower layers. The network N_{41} with different network layers' topologies shows obvious superiority of RBL.

For the bounded case ($LB = 1000$ for N_{12} and $LB = 10$ for N_{41}), as it can be seen from the figures 8 and 9, the situation is the same as in the unbounded case. Lexicographically ordered revenue vector for the problem RBL is again greater than the one for the RUL. In this case, the difference is non-negligible for both N_{41} and N_{12} as it can be seen from the revenue values (for N_{12}) below:

$R^{RBL} = \{9078.90, 9078.90, 9078.90, \ldots, 11592.75\} >$
$R^{RUL} = \{9069.50, 9069.50, 9069.50, \ldots, 11581.00\}.$

In this case revenue values for RBL are significantly higher than for RUL, because the former has more reconfiguration capabilities (on both layers simultaneously) which are especially useful under the tight LB constraints. It can also be seen from the figures, that revenue values for the bounded case are, as expected, smaller than in the unbounded case.

Much higher values of revenues are achieved in RUL and

RBL cases in comparison to RLL, because the first two algorithms involve lexicographical order maximization. Higher revenue values also mean higher values of aggregated flows (X_{ds}), which are beneficial for elastic traffic networks. Besides, even a small difference in total logarithmic flows makes much bigger difference between individual flows X_{ds}. Figures, showing total (non-logarithmic) values X_d for each failure situation, have the similar character to those for revenues. It can be seen from the results, that both RUL and RBL are almost equally good, as compared to RLL, although RBL performs better, especially for the networks with different topology of the layers (e.g. N_{41}) or with tight lower bounds.

VII. Conclusions

The paper presents three different problems (RLL, RUL and RBL) for robust/fair design of two-layer networks and an iterative algorithm to solve them. The problems differ by the reconfiguration option in the case of link failures. All the three problem formulations assure fair allocation of resources to demands. The efficient design algorithm for solving the formulated problems is given. It should be noted, that the presented problem formulations and the algorithm can be extended to networks with more than two layers.

Problem RLL allows to perform reconfiguration only in the lower network layer, RUL reconfigures flows only in the upper layer and RBL- simultaneously in both layers. All the problems assure Proportionally Fair bandwidth allocation among different demands. Besides, the formulations assure Max-Min Fair revenue allocation among different failure situations, resulting in a “two-dimensional” fairness. Some comments on modular dimensioning of link capacities are also given (in IV-D).

A numerical case study of two network examples (N_{12} and N_{41}) is presented. It shows that RBL is clearly superior when topologies of layers are not similar (N_{41}), while RBL and RUL are almost equally good for the networks with similar layers' topologies (N_{12}), though RBL is still marginally better, as any feasible solution of RUL is also a feasible solution of RBL. RBL also performs better than RUL when high lower bounds are imposed. It is also interesting, that in the case of N_{12}, the highest revenue value (in the unbounded case) has been attained for RUL. Both RUL and RBL perform much better than RLL. These observations favor RUL option for the networks with similar topologies of the layers, as it is considerably simpler than RBL. For the networks with different topologies of the layers, however, RBL is significantly better than RUL.

In this study, full reconfiguration has been assumed in the case of failures, which is not too realistic, especially in the lower layer. However it shows what results could be achieved in that case and weather it is worth to use some kind of (coordinated) two-layer reconfiguration. More realistic reconfiguration strategies, as e.g. link protection, will be the subject of future work.

References

[1] D. Bertsekas, R. Gallager, *Data Networks*, Prentice Hall, 1987.

[2] W. Ogryczak, “On the lexicographic minimax approach to location problems”, *European Journal of Operational Research*, Vol. 100, pp. 566-587, 1997.

[3] F.P. Kelly, A.K. Mauloo, D.H.K. Tan, “Rate control for communication networks: Shadow Prices, Proportional Fairness and Stability”, *Journal of the Operational Research Society*, Vol. 49, pp. 2006-2017, August 1997.

[4] M. Pióro, E. Kubilinskas, P. Nilsson, M. Matuszewski, “Robust dimensioning of Proportionally Fair networks”,*2nd Polish-German Teletraffic Symposium (PGTS 2002)*, pp. 235-251, September 23-24 2002.

[5] M. Pióro, P. Nilsson, E. Kubilinskas, G. Fodor, “On efficient Max-Min Fair routing algorithms”, *8:th IEEE International Symposium on Computer Communications (ISCC'2003)*, Antalya, June 2003.

[6] M. Pióro and D. Medhi, *Routing, Flow and Capacity Design in Communication and Computer Networks*, USA: Morgan Kaufmann Publishers, ISBN 0125571895, to appear in March 2004.

[7] *ILOG CPLEX 7.1 User's Manual*, France, March 2001.