Flow optimization in IP networks with fast proactive recovery

Mateusz Dzida, Michal Zagozdzon, Mateusz Zotkiewicz, Michal Pióro

Institute of Telecommunications, Warsaw University of Technology, Poland
{mdzida, mzagozdz, mzotkiew, mpp}@tele.pw.edu.pl

Keywords: protection, rerouting, fast reroute, LFA, MHRP

The post-failure convergence of shortest path routing (SPR) protocols used in IP networks can be too slow to meet the restrictive requirements (i.e. maximum allowable delay, jitter, etc.) of the multimedia services and therefore new protection/restoration mechanisms combined with IP routing are of interest. The article addresses the optimization of three potential rerouting mechanisms based on the IP fast reroute mechanism proposed by Shand and Bryant [1].

The first mechanism takes advantage of multiple equal-cost (shortest) paths (ECMP) where two or more ECMP paths outgoing from one router can be used to protect one another in the IP fast reroute mechanism. However, due to a limited number of the ECMP paths, ECMP protection cannot be used as a stand-alone protection mechanism assuring protection against link failures. Therefore, two other mechanisms, called loop-free alternate (LFA) and multi-hop repair path (MHRP) are considered. LFA protection consists of determining an alternative next-hop address used in the case of a link failure. MHRP is a generalization of LFA using multi-hop tunnels to redirect packets from the failing link to a router that is able to send them to the destination based on a shortest path based forwarding. For each mechanism we formulate an optimization problem as a mixed integer programme (MIP).

We also consider a combined approach where protection is assured through ECMP paths, LFA next-hop addresses, or MHRP paths. Thanks to the variety of such a protection mechanisms, the IP fast reroute technique is able to provide protection for any single link failure. The associated optimization problem (consisting in a simultaneous optimization of a weight system, LFA alternative next-hop addresses and MHRP paths) is difficult and is thus approached with a heuristic method. In our numerical experiments we evaluate the effectiveness of this heuristic method.

1. Introduction

Shortest-path routing (SPR) is a widely deployed intradomain routing mechanism in today's Internet. Interior gateway protocols (IGP) like Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS) determine routing paths as shortest paths with respect to predefined link weight systems. Hence, selection of proper link weights is a traffic engineering task in the networks using SPR.

An SPR weight setting problem is NP-hard (cf. [2]) and is typically approached with heuristics. Although the problem can be resolved with exact methods (e.g., formulated as a MIP problem and solved with a branch-and-cut algorithm) such an approach allows for resolving the problem only for small instances with up to, say, 10 nodes. An SPR routing mechanism accompanied with periodic broadcast of the link weights can be used to restore traffic flows on paths affected by link failures. Once a failure of a link is detected in its adjacent router, the router broadcasts unavailability of the link, assigning a large weight to it, e.g., the maximum possible value, which in the case of OSPF is equal to 2^{16}-1. Each time a router obtains a weight update message, it recalculates the routing paths and in this way all failed links are eliminated from the routing paths. However, link weight updates and recalculation of the shortest paths introduces delay in traffic restoration.

According to [1] typical delays in OSPF post-failure convergence are as follows:

1) **Detection time**

Delay of failure detection is of the order of a few milliseconds (in the case of the detection on the level of a physical layer) to tens of seconds (using hello packets of routing protocols).

2) **Failure local reaction**

Delay related to generation and broadcasting of link state update messages that depends on some internal hold-down delays.

3) **Broadcasting**

Delay resulting from the link state update messages propagation through the network. Typically of the order of tens of milliseconds per each hop.

4) **Shortest paths update**

Delay of the order of a few milliseconds due to calculations of the shortest paths with respect to the updated weight system.

5) **Forwarding table update**

Delay introduced by an update of the forwarding table in forwarding hardware. For some routers it is up to several hundred milliseconds.

It follows that the total path-update delay is not negligible. Although recent developments in tuning the timeouts of the IGP protocols allowed to reduce the delay of the SPR re-convergence to second(s) [3], new multimedia services based on streaming transmission, e.g., VoIP or VoD, are typically delay (jitter) sensitive and still may not accept this relatively slow SPR re-convergence after a failure. Therefore, many current research activities focus on developing other restoration mechanisms applicable in SPR networks and satisfying the requirements of the streaming services (see [4,5]). A promising idea of an alternative recovery mechanism is a proactive IP Fast Rerouting (IFR) mechanism, based on the concept of MPLS Fast Rerouting. The IFR recovery mechanism is described in IETF draft [1].

Typically, network operators set SPR link weights proportional to the inverse of link capacities or use uniform link weights equal to 1 (in the latter case routing paths are the shortest paths with respect to the number of hops). In fact, these two weight systems do not take into account volume of the traffic demands (actual or approximated). Therefore, the resulting routing patterns may lead to otherwise avoidable link overloads or to unbalanced network utilization. To overcome these drawbacks, the SPR weight setting problem can instead be approached with network optimization techniques. An appropriate routing optimization problem is formulated (as an instance of multi-commodity flow optimization taking into account network topology and traffic demands) and resolved with proper optimization methods. Moreover, while formulating and solving the problem, network resilience to failures can be explicitly taken into account.

Assuming a set of significant failures that can occur in the network, we can formulate an optimization problem which allows for calculating the SPR routing paths in the normal (failure-free) state as well as in any considered failure state. Resolving such a formulation, we may try to find a weight system that does not cause link overloads in any failure state (of course if such a solution exists) and optimizes certain traffic engineering criteria, e.g., maximal residual capacity. Hence, approaching the problem with network optimization techniques is a suitable approach to tackle the problem of weight selection.

In the article we discuss three proactive IFR mechanisms:

- equal-cost multiple paths,
- loop-free alternates,
- multi-hop repair paths,

that are taken into account in the recovery scenario assuming single link failures.

Let *e* be the failing link and w^o be the current weight system. If there exists at least one surviving shortest path from the starting router of link *e* to a particular destination, affected packets are redirected to the surviving SPR paths (with appropriate ECMP split in the case of multiple surviving paths). Otherwise (all shortest paths from the starting node of *e* to the destination are using link *e* in the normal state), the router checks if the failing link *e* is assigned an alternate next-hop address (LFA) for the required destination. If this is the case, the packets are redirected to the router associated with this address and then are forwarded on the shortest paths from this router to the considered destination. As both recovery mechanisms do not guarantee protection in all failure states (all single link failures), a third mechanism seems unavoidable. This is the case when for some router neither ECMP split is used nor there exits an LFA next-hop address such that when used does not result in flow loops. One such mechanism is to use multi-hop repair paths (MHRPs). A repair path in this case is a pre-established MPLS tunnel from the starting router of link *e* to a remote router. Similarly, as in the case of LFA, end-router of the repair path forwards redirected packets to the destination.

The considered IFR mechanisms allow for the compensation of the negative impact of relatively slow convergence of the routing protocol after a failure is detected. In fact, IFR framework, if deployed in the network, eliminates the need for the post-failure routing protocol re-convergence. The detecting routers simply locally reroute the affected flows using either the surviving ECMP paths, alternative LFA next-hop addresses, or multi-hop repair paths (MHRPs). This can be done very quickly as no exchange of information with other routers is needed. Moreover, it guarantees that the rerouted flows will reach the destination and will not be lost, e.g., due to transient flow loops that are most likely to occur while the re-convergence of the conventional IP routing is concerned.

An important aspect related to the considered IFR restoration mechanisms is the fact that they should be taken into account while designing the distribution of network flows, e.g., by the appropriate selection of link weights, LFA addresses, and routes for MHRP paths. The goal might for example be to optimize the utilization of available resources. The best way to approach this problem is to use network optimization techniques, which is the main goal of this paper. In this regard, the main contributions of the article are as follows. For each mechanism we formulate an optimization problem as a mixed integer programming (MIP) and consider a combined approach where protection is assured either through ECMP paths, LFA next-hop addresses, or MHRP paths.

Thanks to the variety of the protection mechanisms used, IP fast reroute technique is able to provide protection for any single link failure. The associated optimization problem consisting of a simultaneous optimization of a weight system, LFA alternative next-hop addresses and MHRP paths, is difficult and is thus approached with a heuristic method. In our numerical experiments we evaluate effectiveness of this heuristic method.

The article is organized as follows. In Sections 2, 3, and 4 we discuss in more detail three considered recovery mechanisms and present corresponding MIP optimization problems. Due to the difficulty of the combined approach involving all these three mechanisms, Section 5 is devoted to the presentation of an efficient heuristic method that deals with this kind of problem. It is then evaluated by means of the numerical study presented in Section 6.

2. Equal-cost multiple paths

In this section we present an idea of using equal-cost multiple paths (ECMP) as recovery paths in the case of a link failure (cf. [1]). Typically, when a network operator chooses a link weight system to be used in a network, it may appear that there are multiple shortest paths connecting some routers.

Basic implementations of OSPF or IS-IS routing protocols do not clearly state which shortest path should be used in such a case. Therefore, the deployed implementations of the SPR routing arbitrarily choose routing paths among the shortest paths, e.g., the protocols can choose a next-hop address of the router with the smallest id among the neighboring routers associated with the shortest paths. Due to ambiguity in routing path selection, many works concentrate on selection of the weight systems generating single unique shortest paths. Such systems allow for exactly determining the routing paths without any ambiguity.

However, in the case of a failure it may happen that instead of one shortest path multiple shortest paths appear. Thus, in general, this approach does not overcome the ambiguity. Another approach to deal with multiple shortest paths is to use all available shortest paths to carry traffic. With the help of hashing functions, the flow is equally split in the router among each outgoing shortest-path link (this is the main principle of the equal-cost multiple path (ECMP) flow split). Ideally, the hashing function should allow for equal traffic splitting such that each outgoing shortest-path link carries flow of the same (aggregated) volume.

Notice that multiple shortest paths that do not share a common outgoing link in a particular router may be used as IFR backup paths, protecting one each other. The router detecting a failure immediately erases a next-hop address related to the unavailable neighbor, and distributes evenly the flow destined to a particular destination to all remaining next-hop addresses, related to the non-affected shortest paths. To protect as many links as possible the split must be done at each intermediate router. Unfortunately, a weight system satisfying this condition does not exist. This is because for a particular destination there is always at least one router that, due to the properties of the shortest paths, must not apply ECMP.

However, the protection based on ECMP paths is among the fastest restoration IP mechanisms (we have valid alternative routes in hand). Hence, the optimization problem that arises is how to choose a weight system that generates ECMP shortest-path flows such that the link capacities are not exceeded and a number of routers with multiple outgoing paths is maximal. Below we state the related flow optimization problem. Its solution is an ECMP flow pattern (and the corresponding weight system) such that the link loads do not exceed the link capacities and demands are realized on the shortest paths. Our objective is to maximize the number of routers in which the ECMP splits is done. This problem is a combinatorial optimization problem, and it may be formulated as a MIP.

A. Problem formulation

To formulate the problem as a MIP we use the formulation presented in [2] (see also [6-9]) that searches for shortest-path routing patterns (and associated link weight systems) optimizing simple traffic engineering criteria for the nominal state of the network. Below we re-write this formulation introducing a different objective function. We use the following variables:

- $x = (x_{et} \geq 0 : e \in E, t \in V)$:
  vector of continuous flow variables;
  x_{et} denotes the total flow destined to node t
  realized on link e
- $u = (u_{et} \in \{0,1\} : e \in E, t \in V)$:
  binary vector of routing variables;
  $u_{et} = 1$ if, and only if, link e is on a shortest path
  to destination t
- $y = (y_{vt} \geq 0 : v, t \in V)$:
  vector of continuous flow variables;
  y_{vt} denotes volume of total flow assigned to each link e
  used for sending the traffic outgoing from node v
  to destination t
- $w = (1 \leq w_e \leq W : e \in E)$:
  vector of continuous variables representing link weights
  (W – maximum weight)
- $r = (r_{vt} \geq 0 : v, t \in V)$:
  vector of variables representing lengths of the shortest paths;
  r_{vt} denotes the length of the shortest path from node v
  to destination t, calculated according to link weights w
- $q = (0 \leq q_{vt} \leq 1 : v, t \in V)$:
  vector of binary variables indicating whether router v
  applies ECMP split for the destination t;
  $q_{vt} = 1$ if, and only if, there are at least two links outgoing
  from v that belong to the shortest paths to destination t
  (ECMP is used); 0 otherwise.

According to the above definitions the variables are supposed to fulfill the following relations:

- $u_{et} = 0$ implies that $x_{et} = 0$
- $r_{vt} = \sum_{e \in P} w_e$, where P is a shortest path
  from v to t ($u_{et} = 1$ for all $e \in P$); $r_{tt} = 0$ for each $t \in V$.

The formulation is as follows:

$$\textbf{maximize} \sum_{v \in \mathcal{V}} \sum_{t \in \mathcal{V}} q_{vt} \tag{1a}$$

subject to

$$\sum_{e \in \delta+(v)} x_{et} - \sum_{e \in \delta-(v)} x_{et} = h_{vt} \; v, t \in \mathcal{V} \tag{1b}$$

$$\sum_{e\in\delta^-(t)} x_{et} = \sum_{v\in\mathcal{V}\setminus\{t\}} h_{vt} \qquad t\in\mathcal{V} \tag{1c}$$
$$\sum_{t\in\mathcal{V}} x_{et} \leq c_e \qquad e\in\mathcal{E} \tag{1d}$$
$$r_{b(e)t} + w_e - r_{a(e)t} \geq 1 - u_{et} \qquad t\in\mathcal{V},\ e\in\mathcal{E} \tag{1e}$$
$$r_{b(e)t} + w_e - r_{a(e)t} \leq M(1 - u_{et}) \qquad t\in\mathcal{V},\ e\in\mathcal{E} \tag{1f}$$
$$x_{et} - y_{vt} \geq 0 \qquad v,t\in\mathcal{V},\ e\in\delta^+(v) \tag{1g}$$
$$x_{et} - y_{vt} \leq M(1 - u_{et}) \qquad v,t\in\mathcal{V},\ e\in\delta^+(v) \tag{1h}$$
$$x_{et} \leq M u_{et} \qquad t\in\mathcal{V},\ e\in\mathcal{E} \tag{1i}$$
$$\sum_{e\in\delta+(v)} u_{et} \geq 1 + q_{vt} \qquad v,t\in\mathcal{V},\ v\neq t \tag{1j}$$
$$r_{tt} = 0 \qquad t\in\mathcal{V} \tag{1k}$$
$$q_{vt} \in [0,1] \qquad t\in\mathcal{V},\ e\in\mathcal{E} \tag{1l}$$
$$u_{et} \in \{0,1\} \qquad t\in\mathcal{V},\ e\in\mathcal{E}. \tag{1m}$$

Formulation (1) specifies a multi-commodity flow optimization problem in the aggregated node-link notation (cf. [2]). Constraints (1b)–(1c) express the aggregated flow conservation conditions for the flow variables *x*. Constant h_{vt} denotes the volume of the requested demand from node *v* to node *t*.

Constraint (1d) does not allow the total link flow to exceed link capacity c_e.

The quantity $r_{b(e)t} + w_e - r_{a(e)t}$ in constraints (1e)–(1f) measures the difference between the length of the shortest path from *a*(*e*) to *t* (given by $r_{a(e)t}$) and the length of the shortest path from *a*(*e*) to *t* necessarily traversing link *e* (for the latter, the length of the sub-path from *b*(*e*) to *t* is determined by $r_{b(e)t}$). Note that link *e* is on a shortest path to node *t* if, and only if, $r_{b(e)t} + w_e = r_{a(e)t}$ – this condition is enforced by constraints (1e)–(1f). Hence, the routing vector *u* determines the shortest paths according to the weight vector *w*.

Constraints (1g)–(1h) enforce that traffic outgoing from each node is distributed evenly on all shortest paths to a destination, and constraint (1i) assures that traffic is not routed on the links that do not belong to the shortest paths, i.e., constraint (1j) enforces traffic destined to node *t* to use only the links $e \in E$ allowed by the routing configuration specified by vector *u* (i.e., the links with $u_{et} = 1$). Constraint (1j) allows maximized variables *q* to be ones if only the shortest paths split in the considered node.

Formulations analogous to (1) were published in the literature (see [10-13]). In [14], a non-linear formulation of the considered problem is described. The basic formulation presented in [2] is able to provide good quality lower bounds calculated using its linear relaxation and uses less constraints as compared to other models. However, a direct use of standard MIP solvers to IFR/ECMP can fail already for rather small networks with, say, 10 nodes.

Notice that if a router uses just one outgoing link to transit traffic to a specific destination and this particular link fails, the router is unable to transfer packets until the standard SPR re-convergence is performed. Since restoration based on the ECMP paths cannot be used in such a case, another fast restoration mechanism should be used instead. Authors of [15] (see also [16]) proposed one such mechanism, called loop-free alternatives described in the next section.

3. Loop-free alternates

The idea of using loop-free alternates (LFA) is to define, for each destination, an alternative next-hop address used in the case of a failure of any adjacent link. Once such a failure occurs, the detecting router transmits all packets originally sent through the failed link to the predefined alternative next-hop address of a neighboring router. Because the described mechanism concerns only the local protection, the router associated with the new next-hop address is not aware of the failure. Hence, selection of the alternate next-hop address must be done carefully enough to avoid traffic cycling, i.e., the router associated with the new next-hop address cannot send back the redirected packets to the router performing the recovery action. Inappropriate application of the IPR/LFA recovery mechanism may result in flow loops. Therefore, we can only choose the next-hop addresses of the routers that can transmit traffic to the destinations omitting the failing link. Clearly, to meet this requirement, the link weight system must satisfy the triangle condition [15].

Assuming destination *t*, the condition determining if link *e*, starting in node *v*, can be used as an alternate next-hop is as follows:

$$r_{b(e)t} \leq r_{b(e)v} + r_{vt} : v,t \in V, e \in \delta^+(v). \tag{2}$$

As already mentioned, protection on the ECMP paths cannot guarantee 100% protection against single link failures. For this reason, a combined IFR/ECMP+LFA protection mechanism can be considered. Below, we study an optimization problem related to selection of LFA router addresses used to redirect packets from the links not protected through ECMP paths for the given weight system w^o. Considering a fixed vector w^o we can easily calculate the corresponding solution of formulation (1). Let us denote this solution by (u^o,r^o,x^o). Vector w^o induces a routing pattern. Thus, at this stage we know which links can be protected using ECMP and which can not. The latter one corresponds to the links that are the sole shortest path links outgoing from some router to a given destination. Clearly all paths using such links are affected by the failures and therefore for all such links we validate which of them can be protected using the LFA mechanism.

To select the next-hop address for a specific destination we determine all links starting at the same router as the failing link *e* and choose the addresses of the routers satisfying the triangle condition for the considered destination. Then, for each selected addresses we choose another valid next-hop address such that the resulting flow distribution does not exceed given link capacities. Although this problem is likely NP-hard, due to a small number of admissible (with respect to condition (2)) links outgoing from one router (typically several links) it seems reasonable to formulate the problem as a MIP and resolve it with the use of a standard MIP solver.

For a given link *e* to be protected the MIP formulation is as follows:

$$\textbf{minimize } z \tag{3a}$$

subject to

$$\textstyle\sum_{t\in\mathcal{V}}\sum_{f'\in\delta^+(a(e))\setminus\{e\}} l^o_{f'tf}\, g_{f't} \le c_f z \qquad f\in\mathcal{E} \tag{3b}$$

$$\textstyle\sum_{f\in\delta^+(a(e))\setminus\{e\}} g_{ft} = G^o_t \qquad t\in\mathcal{V} \tag{3c}$$

$$g_{ft} \le r^o_{a(e)t} + r^o_{b(e)a(e)} - r^o_{b(e)t} \qquad t\in\mathcal{V},\ e\in\mathcal{E} \tag{3d}$$

$$g_{et} = 0 \qquad t\in\mathcal{V} \tag{3e}$$

$$g_{ft} \in \{0,1\} \qquad t\in\mathcal{V},\ f\in\mathcal{E}, \tag{3f}$$

where $l^o_{f'tf}$ is a constant representing the load of link f resulting from the selection of link f' as a LFA address to redirect traffic originally flowing through e with destination in t. This is indicated by a binary variable $g_{f't}$ equal to 1 if link f' is chosen as an LFA next hop address for a destination t; and 0 otherwise. Clearly, the weight system w^o exactly determines the vector of link loads $l^o = (l^o_{f'tf} : f', f\in E, t\in V)$.

G^o_t is a binary constant determining if traffic to destination t can be protected in the considered router by the LFA mechanism, i.e., if there exists at least one neighboring router satisfying the triangle condition.

The formulation (3) is an instance of multi-knapsack problem with constrained selection of items (which in this case are redirected flows), where capacity constraint (3b) is mentioned knapsack constraint. Constraint (3c) is used to enforce selection of either 0 or 1 alternate next-hop address for the considered destination, accordingly to value of the constant G^o_t. Admissibility of a specific next-hop address (to protect link e) depends on obeying the triangle condition by the candidate protection links f', that is assured by constraint (3d).

As an objective function we consider minimization of the maximal link utilization or overload (when protection requires more capacity than available).

Similarly, as in the case of the ECMP protection, the delay related to the IPR/LFA restoration is typically much smaller than the delay of the standard SPR re-convergence. However, applicability of the ECMP or LFA mechanism can be still questionable from the operator's point of view. It is so because these mechanisms may require link overprovisioning, but as in the case of any local protection there is a tradeoff between short restoration time and large capacity requirement. Authors of [1] anticipate that the two discussed IFR mechanisms, would be able to provide protection in around 80% cases of single link failures. To assure 100% protection, another mechanism is needed. One such mechanism, called multi-hop repair paths (MHRP), is discussed in the next section.

4. Multi-hop repair paths

MHRP is a protection mechanism applied when a disrupted flow cannot be rerouted on a protection path using ECMP or LFA, i.e., in a case when there is exactly one outgoing link, say e, (used by the disrupted flow) belonging to the shortest path and none of the links different than e and outgoing from the originating node of link e obey the triangle condition (2). Clearly, it may happen that none of both mechanisms can be applied. In order to assure 100% reliability, we need to consider multi-hop reroute paths that do not necessarily terminate in the routers adjacent to the originating node of the failing link. In general, following [1], in most of the cases it can be achieved when using two-hops reroute paths only. Although, the ECMP and LFA mechanisms can be adopted in IP quite easily, e.g., as simple extensions to the router's software, the third protection mechanism requires more sophisticated solutions, like MPLS tunnels (cf. [17]), multiple FIBs (cf. [18]), or the mechanism of the alternative shortest paths (cf. [19]).

Below, we investigate an application of MHRP as a mechanism complementary to the ECMP+LFA protection mechanism. Namely, a router detecting a failure performs the following restoration actions for each destination:

1) The router verifies if there exists at least one surviving ECMP path to the destination.
   If this is the case, the flow from the broken link is evenly distributed over all the surviving ECMP path links.
2) If no ECMP path exists, the router checks if an alternative next-hop is defined for this destination.
   If this is the case, the flow is redirected to the defined next-hop address.
3) If neither ECMP nor LFA can be used, the router establishes a tunnel to a remote router used as a multi-hop repair path.

Clearly, using such a three-fold protection mechanism 100% protection can be achieved.

Now, we consider an optimization problem computing the multi-hop repair paths for all links that are protected neither through ECMP nor LFA for the given link weight system w^o. Again, the considered problem may be formulated as a MIP.

Assuming that link e is failing, the formulation is as follows:

$$\textbf{minimize } (a+b)\alpha + \textstyle\sum_{f\in\mathcal{E}}\sum_{t\in\mathcal{V}} s^e_{ft} + z \tag{4a}$$

subject to

$$\textstyle\sum_{f\in\delta^+(v)} s^e_{ft} - \sum_{f\in\delta^-(v)} s^e_{ft} = -S_{vt} \qquad v,t\in\mathcal{V} \tag{4b}$$

$$\textstyle\sum_{f\in\delta^+(a(e))\setminus\{e\}} s^e_{ft} = 1 \qquad t\in\mathcal{V},\ f\in\mathcal{E} \tag{4c}$$

$$\textstyle\sum_{t\in\mathcal{V}} x^o_{et} s^e_{ft} + \sum_{t\in\mathcal{V}}\sum_{v\in\mathcal{V}} n^o_{vtf} S_{vt} \le c_f z \qquad f\in\mathcal{E} \tag{4d}$$

$$s^e_{et} = 0 \qquad t\in\mathcal{V} \tag{4e}$$

$$S^e_{a(e)t} = 0 \qquad t\in\mathcal{V} \tag{4f}$$

$$z \le 1 + aM \tag{4g}$$

$$z \le b + 1 \tag{4h}$$

$$b \ge 0 \tag{4i}$$

$$s^e_{ft} \in \{0,1\} \qquad t\in\mathcal{V},\ f\in\mathcal{E} \tag{4j}$$

$$S^e_{vt} \in \{0,1\} \qquad v,t\in\mathcal{V}. \tag{4k}$$

$$a \in \{0,1\} \tag{4l}$$

The above problem is solved for each link e that for a current weight system w^o cannot be protected either by ECMP or LFA mechanisms. As a result of solving Problem (4) we obtain multi-hop routes that can be used to protect these links.

The resulting tunnels are described by means of binary variables $s^e = (s^e_{ft} : f \in E, t \in V)$ specifying, for a given link *e* to be protected, whenever link *f* is a part of a multi-hop repair path to destination router *t*. Using these variables we write down the flow conservation constraints (4b)–(4c) assuring the continuity of the flow carried through the tunnel. Since the end-nodes of the multi-hop repair paths are not predetermined (they are optimized), they are expressed by the values of the binary variables S^e_{vt}. A positive value of variable S^e_{vt} states that router *v* is the terminating node of the protection tunnel used for destination *t*. Calculating the link loads one must take into account two elements: the load resulting from the selected multi-hop repair paths, and the load resulting from the ECMP paths from the selected end-nodes of the multi-hop repair paths to the traffic destinations. Constraint (4d) is a capacity constraint taking into account link loads resulting from the multi-hop repair paths, determined by the first term in the sum, where x^o_{et} is a constant representing an amount of traffic sent in a normal state to destination *t* on link *e* (link *e* is the link to be protected). The second term of the sum (4d) represents the link load components induced by the flows leaving MHRP paths and sent to the destination using shortest paths. This is achieved using constant n^o_{vtf} which is equal to the amount of traffic sent through link *f* to destination *t* if we decide to use an MHRP path that terminates in *v*. Constraints (4e) and (4f) assure that protected link *e* cannot be used in any MHRP path and that MHRP path cannot terminate in the originating node of link *e* respectively.

When solving Problem (4) our primary goal is to find a tunnel configuration that does not cause link overloads, or when it is unavoidable, tries to minimize the overload given as *z*. This is reflected in the objective function by the component $(a+b)\alpha$. α is a big-M constant equal to $|E| \cdot |V|$, and *a* is a binary variable equal to 1 if at least one link becomes overloaded $z>1$ (cf. constraint (4g)). *b* was used to linearize (cf. constraint (4h)) the original version of $(a+b)\alpha$ given as a non-linear expression αaz focusing optimization on the minimization of *z* whenever $z>1$. This is our primary goal. Whenever *z* goes below 1 (there are no overloaded links) the optimization switches to minimization of the total length of MHRP paths (due to $(a+b)\alpha = 0$. The second goal is expressed in the objective function using component $\sum_{f \in E} \sum_{t \in V} s_{ft}$. If it happens that for two different solutions their resulting total lengths of MHRP paths are equivalent, the solution with smaller value of *z* if preferable (thanks to sole *z* that is also minimized in the objective function).

5. Resolving methods

In this section we present an efficient heuristic method for resolving the general SPR routing optimization problem with combined ECMP+LFA+MHRP protection. The problem consists in determining a weight system (inducing the ECMP protection), valid LFA next-hop addresses, and the MHRP paths, such that all links in the network are protected through one of the considered IFR mechanisms. We assume that the link capacities and the traffic demands are given. To resolve the problem we adopt a heuristic approach that consists in decomposing the resolution process into three phases.

Algorithm 1 Resolving method

P1: Find a weight system *w* (e.g., resolving the formulation (1)). Denote the resulting solution by w^o. For w^o calculate SPR flow distribution pattern in the normal state. Notice that the given weight system already determines where the ECMP protection can be used.

P2: Denote by *S* the set of all unprotected links (associated with single shortest paths outgoing from a specific router). For each unprotected link $e \in S$ resolve formulation (3) and remove links that can be protected by the LFA alternate next-hop addresses from the set *S*.

P3: For each of the remaining links $e \in S$ resolve formulation (4). Evaluate the obtained nominal and protection routing patterns with respect to objective function (5). If the obtained value is unsatisfactory go back to phase **P1** in order to find another candidate link weight system different from the previously found; otherwise, stop the procedure.

The resulting procedure (cf. Algorithm 1) searches for a weight system that minimizes the value of the penalty function given below:

$$\alpha az + \beta \sum_{e \in E} \sum_{t \in V} m^e_t + \sum_{e \in E} \sum_{f \in E} \sum_{t \in V} s^e_{ft} + z, \quad (5)$$

The role of the first component is to minimize the value of *z* regardless to the number of MHRP paths used and their total length if $z>1$. Again, *a* is a binary variable reflecting the existence of link overloads. More precisely $a=1$ if $z>1$ – overloaded links are identified, and $a=0$ otherwise. Since the coefficient of the first term α is set to $|E| \cdot |V| \cdot (\beta+1)$, i.e., the value bigger than any possible value of the other remaining components, it dominates them all. Therefore, the resolution procedure in its early stage tries to find the solutions that do not contain overloaded links with a little regard to the number of MHRP paths used.

When the solution with $z \leq 1$ is found it is evaluated with respect to the number of used MHRP paths in the first place. For this purpose we introduce a binary variable $m^e_t, e \in E, t \in V$ equal to 1 if traffic sent on link *e* to the destination *t* is protected by the MHRP path, and 0 otherwise. Constant $\beta = |E|^2 \cdot |V| + 1$ is set big enough in order to dominate the remaining components.

The third term in (5) is responsible for minimization of the total length of MHRP paths used. As a result, a weight system with the smallest total length of MHRP paths is evaluated as most profitable. Assuming that the value of *z* does not exceed 1 in the currently considered weight system (no overloaded links are present), the total length of the MHRP paths is expected to dominate the last term, that minimizes the maximal link utilization.

Notice that the general form of Algorithm 1 admits to use any method for searching the global space of feasible link weight systems. For this purpose any metaheuristic or exact branch-and-bound-like procedure can be applied. In the second phase the method recalculates a flow distribution pattern with the ECMP protection applied in all routers where at least two shortest paths exist to a given destination. Next, the procedure verifies if non-protected links may utilize the LFA next-hop mechanism. If it is the case they are selected so to minimize the value of *z*.

The method we used for the selection of w^o used in **P1** is based on the Local Search meta-heuristic. It starts from a base weight system with all link weights equal to 5. In each iteration we modify the link weights of 30% randomly chosen links. Each time a better weight system is found it replaces the base weight system. The weights are modified by adding random numbers chosen uniformly from a range $[-R;R]$.

More precisely, $w_e = \max\{1, w_e + rand(-R,R)\}$. Initially *R* is set to 5 and every 200 iterations is decreased by 1. When *R* reaches the value of 0 the process is stopped and the whole method is restarted. Additionally, before each iteration weights that correspond to most heavily loaded links are increased by $\frac{R}{2}$, if $z > 1$ or the number of MHRP connections in the current best solution is equal to zero. The same changes apply to weights which correspond to links that are not protected either by ECMP or by LFA, when $z \leq 1$ and the current best solution requires the MHRP connections to be established in order to provide the full protection of affected flows.

6. Numerical results

In our numerical experiments we tested the efficiency of the method presented in Section 5. In the implementation the subproblems of phases **P2** and **P3** were solved using the branch-and-cut algorithm of CPLEX 10.1 (cf. [20]). We tested the method using 17 different network examples. More details about the instances used together with numerical results are presented in *Table 1* (down).

For each considered example network the table contains the information about the number of nodes and links in the network. Moreover, it contains results obtained for three different methods for the weight system selection. The first strategy (called "Heuristic") refers to the heuristic method described in Section 5. The second and the third method are simple approaches to the weight setting problem, i.e., all link weights are uniformly set to 1 ($w_e = 1, e \in E$) or proportionally to the inverse of link capacity ($w_e = \frac{K}{c_e}, e \in E, K = \max_e c_e$).

For each of those strategies three different numerical values were collected: *z* – fraction of the capacity used on the most heavily loaded link, MHRP – number of the MHRP connections that have to be established in order to assure a full protection of all demands, MHRP length – sum of the lengths of the established MHRP connections.

The table also contains the information about the number of iterations the heuristic method managed to perform for each problem instance during a 3-hour period of time.

Table 1. Network data and results for the three methods of choosing weight system W

	Number of		**Heuristic**				$w_e = 1, e \in E$			$w_e = \frac{K}{c_e}, e \in E$		
	nodes	links	z	MHRP	MHRP length	iterations	z	MHRP	MHRP length	z	MHRP	MHRP length
network_6	6	18	2.26	0	0	69900	3.01	2	5	3.18	4	10
network_8	8	24	2.61	3	6	21400	4.02	12	24	3.23	9	26
network_10	10	30	2.13	7	14	22600	2.83	10	21	4.51	12	26
network_12	12	36	2.05	27	72	12000	2.23	29	64	3.60	32	79
network_14	14	42	2.48	54	120	7900	3.52	69	144	3.13	38	91
network_16	16	48	3.04	68	164	5800	3.90	66	154	5.43	43	127
artificial_6n	6	28	1.25	0	0	6700	1.25	0	0	2.18	1	2
polska_12n	12	36	2.49	14	43	18400	2.74	26	80	3.37	29	81
cost239 defaultTM	11	52	0.71	0	0	23300	1.08	2	4	1.08	2	4
cost239 ecmpScaledTM	11	52	0.83	0	0	10500	1.14	2	4	1.14	2	4
cost239 sprScaledTM	11	52	0.50	0	0	14200	0.68	2	4	0.68	2	4
geant defaultTM	19	60	0.93	46	98	13700	0.82	124	248	0.92	75	152
labnet ecmpScaledTM	21	106	0.99	1	2	5700	0.95	10	20	0.95	10	20
labnet sprScaledTM	21	106	0.69	1	2	3200	0.65	10	20	0.65	10	20
nobel ecmpScaledTM	27	82	0.99	216	453	2700	0.97	293	590	0.97	293	590
nobel sprScaledTM	27	82	0.99	183	381	6400	0.98	293	590	0.98	293	590

The experiments were performed on a machine running Windows XP on an Intel Pentium 4 (3.0 GHz) processor with 1.96 GB of RAM. The CPU usage limit was set to 50% and the execution time of the procedure was limited to 3 hours for each particular run.

Analyzing the results presented in *Table 1* obtained for simple weight system selections with the results provided by our method we can conclude that optimization of weights is highly recommended. Neither the equal weights nor the weights proportional to the inverses of link capacities provide results comparable to those achieved using the system calculated by our method.

The greatest differences, from an operator's point of view, can be seen as far as *cost239_defaultTM, cost239*

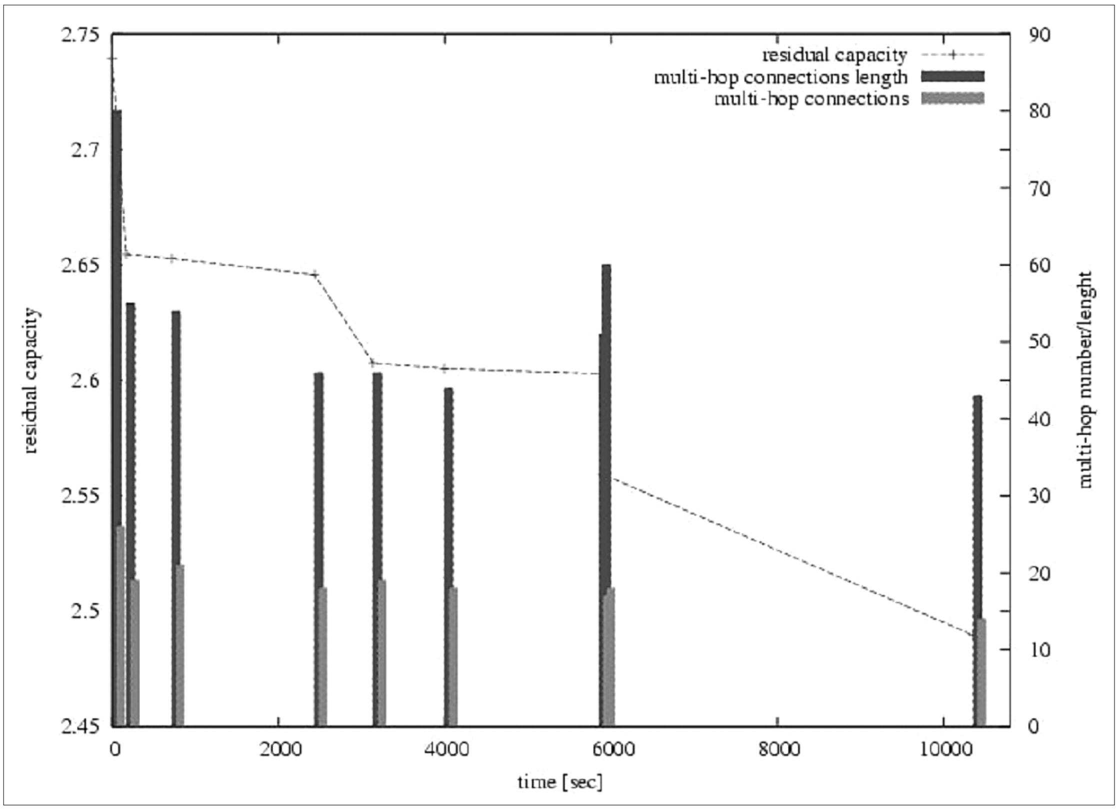


Figure 1. Procedure performance for network12

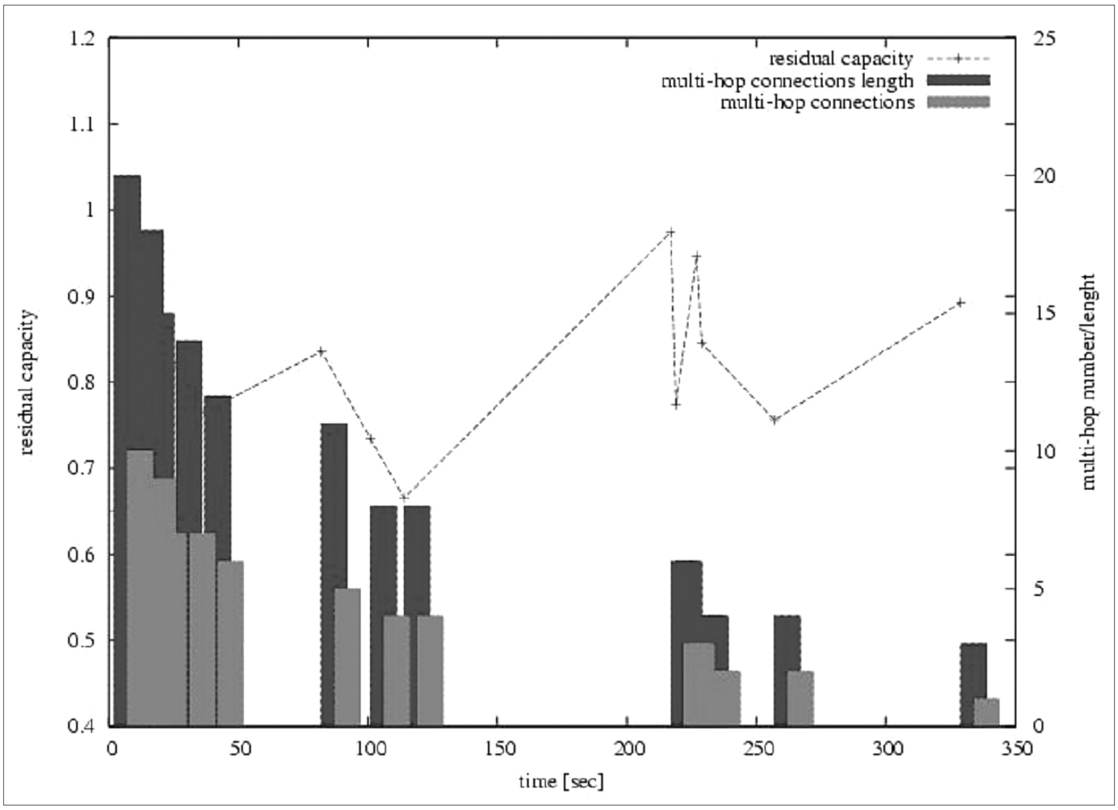


Figure 2. Procedure performance for labnet_sprscaled

_ecmpScaledTM, and cost239_sprScaledTM networks are concerned. In the first two cases the proposed heuristic approach provides a solution that does not cause network overloads. Note that, this is not the case with the naive approaches where provided solutions are characterized by $z>1$. In the third network, the proposed method finds a solution that does not require the MHRP connections to be implemented, and this fact significantly limits the financial burden imposed on the operator.

Concerning the behavior of our algorithm, the conducted experiments allow for concluding that the method works well for both overloaded ($z>1$) and non-overloaded ($z\leq1$) networks. This is illustrated in two figures (on the previous page) that present the behavior of the algorithm in the function of time.

Figure 1 contains the performance data of the method while solving *network12* problem instance, while *Figure 2* deals with problem *labnet_sprScaledTM*. Each column in the diagrams corresponds to a moment when a better solution was found by the heuristic method. Each solution is characterized by the number of the required MHRP connections (grey column), sum of their lengths (black column) and the value of z (solid line). In Figure 1, when $z>1$, the algorithm optimizes z regardless of the number and lengths of the MHRP connections. On the other hand, Figure 2 shows that the procedure focuses on minimization of the number of the MHRP connections when $z\leq1$. Note that, the latter figure does not present the whole optimization process. The reason is because the following improvements were much less frequent, and extending the x-axis to 3 hours would blur the presented best solutions found in the early iterations of the algorithm.

The final results for the problem instance can be found in *Table 1*.

7. Conclusion

In this article we have considered three different protection/restoration IP mechanisms that aim at improving the reactivity of IP routing in the case of failures. These are Equal Cost Multiple Path (ECMP) based flow restoration, Loop-Free Alternates (LFA) and Multi-Hop Repair Paths (MHRP). The considered IP fast reroute (IFR) mechanisms compensate the negative impact of relatively slow convergence of the routing protocol after a failure is detected. In fact, IFR framework, if deployed in the network, eliminates the need for the post-failure routing protocol re-convergence. The detecting routers simply locally reroute affected flows with the guarantee that the rerouted flows will reach the destination and will not be lost (e.g., due to transient flow loops). This is in general very important for different types of streaming services requiring low packet loss ratio, low jitter, etc.

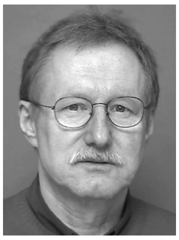
The considered restoration mechanisms, when deployed in the network, could be taken into account while designing the network flows distribution in order to optimize the utilization of available resources. In this context the article focuses on optimization techniques that can be used to solve related network design problems. First, each presented mechanism is considered separately and exact resolution approach based on solving the MIP problem formulations using a Branch-and-Cut algorithm is discussed. Then all IFR mechanisms are assumed to coexist in order to assure 100% reliability. Due to the complexity of the resulting problem, exact optimization methods may not be applicable. Therefore, an efficient heuristic method based on a three phase problem decomposition is proposed. It focuses on determination of a link weight system for which the network flows do not cause link overloads, and such that the number of flows protected either by ECMP or LFA is maximal.

Analyzing the results of the numerical experiments we conclude that our heuristic algorithm allows to obtain solutions that are much better as compared to the solutions with non-optimized systems of link weights, both in terms of network resources utilization and the number of flows protected using ECMP and LFA mechanisms.

Authors

MATEUSZ DZIDA received his MSc and PhD, both from Warsaw University of Technology, in year 2003 and 2009, respectively. His scientific interests focus on designing telecommunication networks and architectures of the post-IP Internet networks. He is an author and co-author of over 30 conference and journal publications in these domains.

MATEUSZ ZOTKIEWICZ is a PhD. student at the Computer Networks and Switching Division at the Institute of Telecommunications, Warsaw University of Technology, Poland, and at Institut TELECOM, TELECOM SudParis, France. He received the M.Sc. degree in telecommunications in 2007 from Warsaw University of Technology. His research interests concentrate on modeling and design of telecommunications networks.



MICHAL PIÓRO is a professor and head of the Computer Networks and Switching Division at the Institute of Telecommunications, Warsaw University of Technology, Poland, and a professor at Lund University, Sweden. He received the PhD degree in telecommunications in 1979 and the DSc degree in 1990, both from the Warsaw University of Technology. In 2002 he received a Polish State Professorship. His research interests concentrate on modeling, design and performance evaluation of telecommunication systems. He is an author of four books and more than 150 technical papers presented in the telecommunication journals and conference proceedings. He has lead many research projects for telecom industry in the field of network modeling, design, and performance analysis. He is deeply involved in international research projects including FP7, Celtic and COST projects.

References

[1] M. Shand, S. Bryant,
"IP Fast Reroute framework,"
Tech. Rep.,
Internet Draft draft-ietf-rtgwg-ipfrr-framework-13, 2009.

[2] M. Pióro, D. Medhi,
Routing, Flow, and Capacity Design
in Communication and Computer Networks.
Morgan Kaufman, 2004.

[3] P. Francois, C. Filsfils, J. Evans, O. Bonaventure,
"Achieving subsecond IGP convergence
in large IP networks,"
In: ACM SIGCOMM, 2005.

[4] G. Iannaccone, C.-N. Chuah, S. Bhattacharrya, C. Diot,
"Feasibility of IP restoration in a tier-1 backbone,"
Tech. Rep., sprint ATL Research Report,
Nr. RR03-ATL-030666, Sprint ATL, March 2003.

[5] A. Wielosz, K. Islam,
"Achieving fast restoration times in IP networks
for IPTV video transport,"
In: NAB, 2007.

[6] M. Dzida,
"Optimization models for designing resilient
intra-domain IP routing,"
Ph.D. dissertation, 2008.

[7] A. Tomaszwski, M. Pióro, M. Dzida,
M. Mycek, M. Zagozdzon,
"Valid inequalities for a shortest-path routing
optimization problem,"
In: Int. Network Optimization Conf. (INOC),
Spa, Belgium, 2007.

[8] A. Tomaszewski, M. Pióro, M. Dzida, M. Zagozdzon,
"Optimization of administrative weights in
IP networks using the branch-and-cut approach,"
In: The 2nd Int. Network Optimization Conf. (INOC),
Lisbon, Portugal, 2005.

[9] M. Dzida, M. Zagozdzon, M. Pióro,
"Optimization of resilient IP networks with
shortest path routing,"
In: 6th Int. Workshop on Design and
Reliable Communication Networks (DRCN),
La Rochelle, France, 2007.

[10] K. Holmberg, D. Yuan,
"Optimization of Internet protocol network design
and routing,"
Networks, Vol. 43, No. 1, pp.39–53., 2004.

[11] L.D. Giovanni, B. Fortz, M. Labbe,
"A lower bound for
the internet protocol network design,"
In: Proc. of the Int. Network Optimization Conf. (INOC),
Lisbon, Portugal, 2005.
pp. B.2. 401–408.

[12] A. Bley, M. Grötschel, R. Wessäly,
"Design of broadband virtual private networks:
Model and heuristics for the b-win,"
In: Proc. DIMACS Workshop on Robust Communication Networks and Survivability,
AMSDIMACS Series, 53, 1998, pp.1–16.

[13] P. Broström, K. Holmberg,
"Multiobjective design of survivable IP networks,"
In: P. Broström – PhD Dissertation,
Optimization Models and Methods for
Telecommunication Networks Using OSPF.
Linkoeping University, 2006.

[14] N. Bourquia, W. Ben-Ameur, E. Gourdin, P. Tolla,
"Optimal shortest path routing for Internet networks,"
In: Proc. of the 1st Int. Network Optimiz. Conf. (INOC),
Evry-Paris, France, pp.119–125., 2003.

[15] A. Atlas, A. Zinin,
"Basic specification for IP Fast-Reroute:
Loop-free Alternates,"
Tech. Rep., Proposed standard, 2008.

[16] S. Nelakuditi, S. Lee, Y. Yu, Z. Zhang, C. Chuah,
"Fast local rerouting for
handling transient link failures,"
IEEE/ACM Trans. Netw., Vol. 15, No. 2,
pp.359–372., 2007.

[17] S. Bryant, C. Filsfils, S. Previdi, M. Shand,
"IP fast reroute using tunnels,"
Tech. Rep., 2004, Internet Draft (work in progrees).

[18] A. Kvalbein, A.F. Hansen, T. Cicic,
S. Gjessing, O. Lysne,
"Fast IP network recovery using multiple routing
configurations,"
In: 25th IEEE International Conference on Computer
Communications (INFOCOM), 2006.

[19] A. Tian, N. Shen,
"Fast Reroute using alternative shortest paths,"
Tech. Rep., 2004, Internet Draft (work in progrees).

[20] CPLEX, CPLEX User's Manual.
ILOG, 1999.