Ireneusz OLSZEWSKI

University of Technology and Life Sciences in Bydgoszcz, Faculty of Telecommunications and Electrical Engineering

An algorithm of choosing LSPs in the MPLS network with unreliable links

Streszczenie. *W pracy zaproponowano algorytm wyboru ścieżek LSPs w sieciach IP/MPLS o zawodnej strukturze. Liczba utraconych pakietów na uszkodzonej ścieżce LSP zależy od czasu odtwarzania uszkodzonej ścieżki na ścieżce zabezpieczającej. Aby ograniczyć czas odtwarzania, odległość pomiędzy węzłami jest ograniczona poprzez ograniczenie długości ścieżki aktywnej. Rozważany problem obejmuje ograniczenie nałożone na długość ścieżki, mierzonej liczbą łączy oraz ograniczenie prawdopodobieństwa uszkodzenia ścieżki. Algorytm rozwiązujący sformułowany problem optymalizacji przy zadanych ograniczeniach wyznacza rozwiązanie lokalne (**Algorytm wyboru ścieżek LSP w sieciach MPLS przy zawodnej strukturze sieci**).*

***Abstract**. In this paper an algorithm for choosing LSPs in the MPLS network with unreliable links is proposed. The number of lost packets on the failed LSP depends on the restoration time of this LSP on the global backup path. In turn, the restoration time depends on the distance between the node which detected failure and the node responsible for traffic redirection from failed to global backup LSP. To reduce restoration time the distance between these nodes is decreased by length limitation of active LSP. The formulated problem covers limitation of the path length determined by the number of links and limitation of LSP failure probability. The algorithm solving the formulated problem of optimization gives a local solution for given limitations.*

Słowa kluczowe: Routing, Wieloprotokłowa Komutacja Etykietowana, struktura sieci, Ścieżka komutowana etykietowo.
Keywords: Routing, Multi Protocol Label Switching (MPLS), network structure, Label Switching Path (LSP).

Introduction

A network can be designed only for assumed initial conditions but the load and traffic characteristics vary in time. The network resources also vary because of the network topology changes (nodes or links failures). An important element of the quality of service (QoS) is the network reliability. Although most of the problems considered in this paper concern the network of different technologies using logical paths conception, so the further considerations will be focused on IP network with Multi Protocol Label Switching (MPLS). Fault management mechanisms in MPLS networks are based on setting up the backup Label Switching Path (LSP) [1]. In case of a failure the traffic can be redirected to the backup path. The backup paths can be static or dynamic [2]. In the first case a backup LSP is pre-established for each active LSP. In the second case a backup LSP is established as a result of failure in the network.

Many algorithms of choosing LSPs in MPLS networks have been proposed so far [3,4,5,6,7,8,9]. All these algorithms minimize the number of rejected requests or the amount of consumed bandwidth in a network. For this purpose the interference of incoming request of LSP set up with requests of LSPs set up which will come in the future is minimized [4, 8, 9] or interference of LSPs already set up in a network with coming request of LSP set up is minimized [3, 6, 8]. The algorithms of LSPs choice presented in [3,4,5,6,7,8,9] are based solely on the state of links occupancy and the number of flows carried by them.

Most of these algorithms, however, do not take into account other aspects, such as link failure probability in the network, packet loss and recovery time of bandwidth on the backup path. The impact of these parameters on the number of links requiring local backup on LSP, number of paths for which failure probability exceeds pre-set threshold and number of rejected requests of setting up paths has been shown in [10]. The analysis has been done on the base of *k*-WSP algorithm which works on *k*-element set of possible paths between each pair of nodes. It should be noticed that fixed *k*-element set of possible paths for each request of setting up connection can not be able assure that obtained result will be near to the optimal solution. However, for the *on-line* routing algorithms (based on this set) the generation of a *k*-element set of possible paths for each request on unreliable topological structure is time-consuming. Therefore, there is a need of research on new routing algorithms taking into account residual bandwidth on each links and other aspects such as link failure probability, packets loss and recovery time of bandwidth.

In the paper, the algorithm of choosing LSPs in IP/MPLS network which considers link failure probability and bandwidth recovery time has been proposed. The paper is organized as follows. In the second part, the problem of optimization has been formulated. In the third part the heuristics algorithm solving this problem and algorithm which determines the optimal solution has been proposed. In the fourth part of the paper the simulation results have been given. In the final part the summary and conclusions has been drawn.

Formulation of optimization problem

Before formulation of the optimization problem in this paper, the limitations occurring in this problem and the objective function will be discussed. The first considered limitation is limitation of LSP length and the second the limitation of LSP failure probability.

Packets Loss (PL) depends on restoration time (RT) and amount of allocated bandwidth b (in Bits/s) on failure LSP.

(1) $$PL = RT\,b + LP$$

where LP is the lost packets in the failure link. Time of restoration is defined as the time between failure detection in the network (node or link) and traffic redirection (stream of packets) from active to backup LSP [10]. This time consists of following four elements: Detection Time (DT), the time of notification (NT) the node responsible for switchover from active LSP to backup LSP by node which detected the failure; Time for Backup LSP Setup (TB) and time of switching over packets stream from active path to the backup path (ST) . So:

(2) $$RT=DT+NT+TB+ST$$

It should be noticed, that if the backup LSP is pre-established then TB can be omitted. The most important element of restoration time is notification time because it is the most responsible for packets loss [2]. This time can be determined as follows:

(3) $$NT=D(i,a)PT$$

where: $D(i,a)$ is the distance determined by the number of links from *a* node which detected failure to node i responsible for switchover from an active path to a backup path. PT is propagation time of the Fault Indication Signal (FIS) through each link. PT consists of the node Processing Delay, Buffer Processing Delay and Link Delay. In [10] it was shown that in the global backup path protection RT is directly proportional to distance $D(i,a)$. For each active LSP protected by global backup path $D(i,a)$ can be changed from 0 to $L(LSP)$ i.e. $L(LSP)> D(i,a)\geq 0$,where: $L(LSP)$ is the length of failure LSP. For $D(i,a)=0$ ingress node i of LSP is responsible both for failure detection and switchover on the backup path. From (1) it results that to minimize Packets Loss, the RT needs to be shortened. In turn, the RT depends on NT (see (2)), whereas NT is proportional to $D(i,a)$. Therefore, by limiting the range of changing of $D(i,a)$ by restriction $L(LSP)$, the PL after failure can be minimized.

The second parameter which should be taken into consideration by the routing algorithm is the probability of LSP failure, which is determined on the basis of probability of links failure in the network. These probabilities can be determined on the basis of an analysis of different statistics or the network operator experience [10]. As the probabilities of each links failure in the network are known, so the probability of LSP failure can be determined as the probability of a complement event. Probability that LSP is in order can be determined as follows:

(4) $$P(LSP\ is\ in\ order)=\prod_{e_i\in LSP}(1-p_i)=1-\sum_{j=1}^{L(LSP)}(-1)^{j-1}\sum_{e_{i_2}e_{i_3},\dots,e_{i_j}\in LSP}p_{i_2}p_{i_3},\dots,p_{i_j}$$

where: p_i is the probability of failure of link e_i. The probability that LSP is failed can be written as follows:

(5) $$P(LSP\ is\ out\ of\ order)=1-\prod_{e_i\in LSP}(1-p_i)\approx\sum_{e_i\in LSP}p_i$$

for adequate small p_i. Routing algorithm will choose such LSP for which probability of failure is not greater than threshold of p_0, assumed for all LSPs in the network. Lower limit of probability of LSP failure causes the limitation of number of links requiring protection on LSP. This eliminates the difference between possible protection methods (Local Backup, Global Backup) from the point of view of the consumed bandwidth. In general, for the case of a great number of links requiring protection, the global protection method is better than the local protection method. However, with a decreasing number of links requiring protection, the local backup method becomes more competitive to the method of global protection [2].

Both, the limitation of path length $D(LSP)$ and limitation of value of probability LSP failure, become constraints of formulated optimization problem. After defining constraints for the optimization, the objective function still needs to be defined. In MIRA the LSP between pair of nodes (s,d) with bandwidth of b units is calculated by using Dijkstra algorithm with the weights of arcs determined as follows [4]:

(6) $$w(l)=\sum_{\substack{(a,c)\in P\setminus(s,d)\\ l\in C_{ac}}}\alpha_{ac}$$

where C_{ac}- is the set of critical arcs for the ingress-egress pair (a,c). An arc is critical for given ingress-egress pair (a,c) if it belongs to any min-cut for that ingress-egress pair. P-is a set of distinguished node pairs (a,c), whereas α_{ac} is inversely proportional to θ_{ac} i.e. $\alpha_{ac}=1/\theta_{ac}$, were θ_{ac} is the max-flow value between ingress-egress pair (a,c). From (6) it results that the weight of link $w(l)$ is the sum of reverses of max-flows for these pairs of nodes $(a,c)\in P\setminus(s,d)$ for which $l\in C_{ac}$. In Constraint Shortest Path First algorithm (CSPF) [3], [7], an LSP between pair of nodes (s,d) is calculated by using Dijkstra algorithm in reduced network on the base weights of arcs in the following way:

(7) $$w(l)=\frac{1}{R(l)}$$

where: $R(l)$ is the residual bandwidth of link l. The reduced network is formed through elimination of all links from the graph, for which the residual bandwidth is less than b units of the required bandwidth LSP path. Weights of links calculated in this way allow to minimize the number of rejected request of setting LSPs because CSPF takes into account the least loaded links during choosing LSP. Moreover, another advantage of CSPF, for so determined weights, is balancing of the network load. CSPF avoids most loaded links. A drawback of CSPF is greater consumed bandwidth [7] due to longer selected paths in comparison with MIRA. Taking into account that admissible lengths of LSPs which was limited in the formulated problem of optimization, it was assumed that the weight of each link will be inversely proportional to residual capacities of these links.

Let $G(N, E, C)$ be the network, where N is the set of nodes (routers) and E is the set of unidirectional links (arcs). C is m–vector of bandwidth of the links. Let n denote the number of the nodes and m the number of links in the network. Let R be an m-vector of residual capacity. Entry j in R vector corresponds to the residual capacity of arc j. Moreover, let current request of setting LSP between pair of nodes (s,d) requires b units capacity. To simplify the notation, we will often refer to a link by (i,j) instead l. Formulated problem of optimization can be shown as follows:

(8) $$Min\left(\sum\frac{1}{R_{i,j}}x_{i,j}\right)$$

(9) $$\sum_j x_{i,j}-\sum_j x_{j,i}=0$$

(10) $$\sum_j x_{s,j}-\sum_j x_{j,s}=1$$

(11) $$\sum_j x_{d,j}-\sum_j x_{j,d}=-1$$

(12) $$\sum_{(i,j)\in E}x_{i,j}p_{i,j}\leq p_0$$

(13) $$\sum_{(i,j)\in E}x_{i,j}\leq L(LSP)$$

(14) $$R_{i,j}x_{i,j}\geq b\quad\forall(i,j)\in E$$

(15) $$x_{i,j}\in(0,1)$$

The vector x represents the flow on the path between pair of nodes (s,d), where $x_{i,j}$ is set to 1 if link (i,j) is used in the path. Formula (8) defines the optimized objective function with weights, which are inversely proportional to residual capacities of the links in the network. Equations (9) to (11) give the flow balance for a path. Inequality (12) is a constraint of probability of LSP failure, whereas, inequality (13) is a constraint of LSP length. Equation (14) states that b units of bandwidth have to be sent between nodes s and d. In general, LSP path selection problem, which minimizes the number of rejected requests of LSP set up is *NP*-hard problem [4]. However, the considered problem, in which the weights of links are defined, is the integer linear programming problem. Because the algorithm solving this problem must work *on-line*, a heuristic approach can be considered only.

An heuristics algorithm – sub-optimal solution

For a given vector of weights whose components $w(l)$, $l=1,2,...,m$ are inversely proportional to residual capacities $w(l)=1/R(l)$, $R(l)\geq b$, the shortest path is determined by Dijkstra's algorithm. If probability of path failure is less or equal to p_0 and length of path (number of links) is less or equal to $L(LSP)$ then the obtained result is the optimal solution. Otherwise, if the obtained path does not satisfy the assumed limitation then the sub-optimal solution is calculated as below. Let's assume that reliability limitation is not satisfied (probability of failure of obtained LSP is grater than p_0). In this case all links (i,j) belonging to LSP are analysed. For each link $(i,j)\in LSP$, for which probability of link failure is equal to $p_{i,j}$ the cost of bypassing this link is calculated as follows:

(16)
$$\kappa_{ij}(k) = \min_{\substack{k\in N\\ k\neq i,j,s,d}} (w_{ik} + w_{kj} - w_{ij})$$

The link (i,j) with the least cost $\kappa_{i,j}(k)$, for which $p_{ik}+p_{kj}<p_{ij}$, is replaced with a pair of links: (i,k) and (k,j). It means modification of primary LSP: $(s\text{-},...,\text{-}i\text{-}j\text{-},...,\text{-}d)$ to LSP: $(s\text{-},...,\text{-}i\text{-}k\text{-}j\text{-},...,\text{-}d)$. This modification allows to increase reliability of the path under minimum-increase of total weight of path (objective function) provided that length of path is not greater than $L(LSP)$. If so obtained path satisfies constraints (12) and (13) then it is accepted as a sub-optimal solution. Otherwise, if constraint (12) is not satisfied and constraint (13) is satisfied, then next analogous iteration for this new path is realized. When the length of path (measure number of links) is greater than $L(LSP)$ then the request is rejected.

The shown algorithm is limited enough. To increase reliability of LSP any node k for which $p_{ik}+p_{kj}<p_{ij}$ must exist at least for one link $(i,j)\in LSP$. The generalization of this algorithm, which eliminates this limitation, will be shown below.

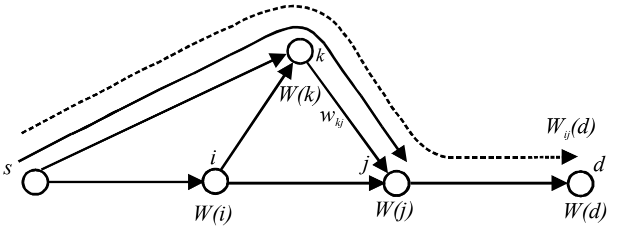

Fig. 1. The bypassing path for link (i,j) and the path between pair of nodes (s,d)

Let $\mathfrak{I}$ be the shortest path tree rooted at s determined by Dijkstra's algorithm for weight of arcs $w(l)=1/R(l)$, $R(l)\geq b$. Let $W(i)$ be the weight of the shortest path from node s to node i. Moreover, let $L(i)$ denote the length of path from node s to node i (determined by number of links) on the basis of obtained tree $\mathfrak{I}$. Let $P(i)$ denote the probability of failure of the path between node s and node i, determined on the same tree $\mathfrak{I}$. If determined LSP between the pair of nodes s and d with weight $W(d)$ satisfies reliability limitation ($P(d)\leq p_0$) and path length limitation ($L(d)\leq L(LSP)$) then this LSP is an optimal solution. Otherwise, for each link (i,j) of LSP the bypassing path is determined. Figure 1. shows a bypassing path (denoted as continuous line) for link (i,j) of LSP $(s\text{-}i\text{-}j\text{-}d)$ and the path between pair of nodes (s,d) (denoted as dashed line), which contains this bypassing path.

The estimation of weight of path $W_{ij}(d)$ from node s to node d, taking into consideration bypassing cost for link (i,j), can be written as follows:

(17)
$$W_{ij}(d) = \min_{\substack{k\in N\\ k\neq i,s,d}} (W(k) + w_{kj} + (W(d) - W(j)))$$

Similarly, the probability of path failure $P_{ij}(d)$ and length path $L_{ij}(d)$, taking into account the bypassing cost link (i,j), for the same path can be written as follows:

(18) $$P_{ij}(d)=P(k)+p_{kj}+P(d)-P(j)$$

(19) $$L_{ij}(d)=L(k)+1+L(d)-P(j)$$

Since the main goal is minimization of summarized weight of path, link (i,j) assuring the least value of estimation of path $W_{ij}(d)$ is removed ($w_{ij}:=\infty$). In the case when for several links $(i,j)\in LSP$ estimations of costs $W_{ij}(d)$ are equal, then the estimation of path failure probability $P_{ij}(d)$ is considered and the link (i,j) with the least $P_{ij}(d)$ is rejected. Finally, in the case when for two or more links estimations of costs $W_{ij}(d)$ are equal and estimations of failure probability $P_{ij}(d)$ are equal then the link (i,j) assuring the least length of path $L_{ij}(d)$ is removed. After removing so determined link, the next shortest path of tree $\mathfrak{I}'$ rooted at node s with weight of arc w_{ij} on link (i,j) is determined. If determined LSP between the pair of node s and d with weight $W(d)$ satisfies constraints (12) i (13) ($P(d)\leq p_0$, $L(d)\leq L(LSP)$) then it is a sub-optimal solution. Otherwise, analogous iteration for this new path is realized

Below, an algorithm solving the formulated problem of optimization is shown. This algorithm is denoted as CSPF_U(p_0; $L(LSP)$), (Constraint Shortest Path First with Unreliable links).

```
Algorithm CSPF_U(p0; L(LSP));
begin {begin CSPF_U()}
  for (i,j)∈E do
     if b≤Rij then wij:=1/Rij else wij:=∞;
condition:=true;
while condition do
   begin
      ℑ:=GetSPTree(s); { Shortest Path Tree roted at s}
      LSP:=GetPath(ℑ, d);   {Get path to d from s in ℑ}
      W^old(d):= ∞; L^old(d):= ∞;
      P^old(d):=P(d);
      if (P(d)≤p0) and (L(d) ≤L(LSP)) then
        begin
           condition:=false;     {LSP satisfies (12) and (13)}
           UpdateMatrix(R);
        end
      else for (i,j)∈LSP do
             begin
               if Wij(d)<W^old(d) then
                 begin
                   (u,v):=(i,j)
                   W^old(d):= Wij(d);
                 end;
```

23: **if** ($W_{ij}(d)$=$W^{old}(d)$) **and** ($P_{ij}(d)$< $P^{old}(d)$) **then**
24: **begin**
25: *(u,v):=(i,j);*
26: $P^{old}(d)$:= $P_{ij}(d)$;
27: **end;**
28: **if** ($W_{ij}(d)$=$W^{old}(d)$) **and** ($P_{ij}(d)$= $P^{old}(d)$) **and** ($L_{ij}(d)$< $L^{old}(d)$) **then**
29: **begin**
30: *(u,v):=(i,j);*
31: $L^{old}(d)$:= $L_{ij}(d)$;
32: **end;**
33: **end;**
34: **if** $W^{old}(d)<\infty$ **then** *{Next iteration}*
35: w_{uv}:= ∞;
36: **else** *condition:*=false; *{Request is rejected}*
37: **end;**
38: **end;** *{ begin CSPF_U()}*

The conditions from line 18 and line 23 can be substituted by condition **if** $(W_{ij}(d)\leq W^{old}(d))$ **and** $(P_{ij}(d)<P^{old}(d))$ **then**. However, the request will be removed if there is not link *(i,j)*, $(i,j)\in LSP$ for which the bypassing path causes growth of LSP reliability. Therefore, in this case link *(i,j)*, $(i,j)\in LSP$, which will assure possibly the least growth of the objective function for the path in the next iteration (line 18), is removed from the LSP. The last condition (line 28) causes choosing the path with the shortest length.

Complexity of the computing function of CSPF_U() for single iteration is $O(n^2)$; The time of choosing the shortest LSP which satisfies constraints (12) i (13) depends on the number of iteration.

An exact algorithm – optimal solution

An algorithm determining the optimal solution, denoted as LM(p_0; *L(LSP)*) for a given request between the pair of nodes *s* and *d* generates all possible paths with length less or equal to *L(LSP)* for reduced network with weight $w(l)=1/R(l)$. The shortest LSP which satisfies reliability constraint (12) is chosen by LM() from all generated paths. To generate all possible paths with length constraint (13) Latin Multiplication [11] was applied. Because of the number of paths between a given pair of nodes is very huge (grows exponentially), therefore, LM() giving the optimal solution can serve to verify other algorithms only.

Obtained results

A study of the considered algorithms has been performed for the mesh structure network with different number of nodes (and links). In this paper the obtained results for network with 15 nodes and 56 unidirectional links are shown. The topology structure of the examined network is shown in figure 2 [4]. Each edge in figure 2 represents a pair of arcs of opposite direction. The bandwidth of 38 links is 1200 units (thin line) and other 18 links is 4800 units (thick line). The values of link failure probability p_{ij} for each link *(i,j)* are shown on arcs of the graph. In this paper it was assumed, similarly as in [4] and [10], that there exists a set of pairs of nodes, for which probability of LSP path rejection is smaller than probability of LSP path rejection for remaining pairs of nodes. Therefore, even in the general case, when every node can be an ingress-egress router, a certain subset of node pairs will be more essential then the remaining node pairs in the network. The distinguished pair of nodes S_i-D_i are: 1-13, 5-9, 4-2 and 5-15. A request of choosing LSPs between pair of ingress-egress routers arrive randomly and each request is uniformly distributed between 1 and 4 units. It was assumed that 5000 requests arrive between the distinguished pair of nodes. Each pair is chosen randomly. In this paper, it is assumed that p_0=0,0008 and *L(LSP)*=6,7,8 depends on experiment.

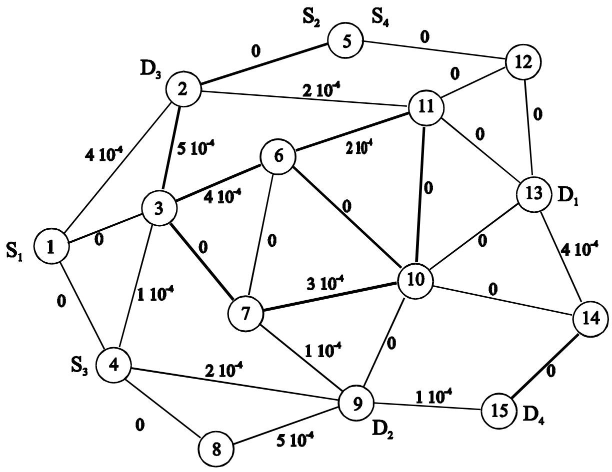

Fig. 2. Topology structure of the considered network

Each experiment for all considered algorithms was performed for the same stream of request. It should be noticed that the length of shortest paths between the distinguished pair of nodes are: Length(1-13)=3, Length(5-9)=4, Length(4-2)=2, Length(5-15)=4. Figure 3. shows dependence of the network bandwidth consumption on the number of requests of LSP set up. The figure shows that the consumed bandwidth for MIRA is less than the consumed bandwidth for other algorithms (CSPF, CSPF_U(0.0008,8) and LM(0.0008,8)). It results from the fact that for MIRA, weights of links which belong to the set of critical links, are calculated on basis of (6), whereas, the weights of other links for which $R_{ij}\geq b$ assume small positive values. It assures minimization of the path length determined by the number of hops and minimization of the consumed bandwidth. Whereas, in the case of all other algorithm weights of arcs *(i,j)*, for which $R_{ij}\geq b$ are determined as $w_{ij}=1/R_{ij}$. So determined weights cause a slight increase of length of paths and an increase of the consumed bandwidth in the network.

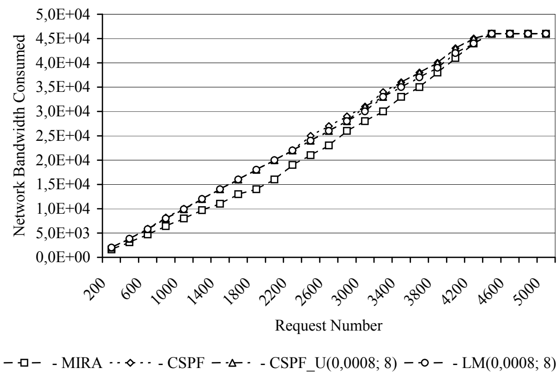

Fig. 3. Network bandwidth consumed vs. request number

Figure 4 shows the number of rejected requests in the dependence on the number of incoming requests for all considered algorithms. It can be noticed that MIRA, CSPF and algorithm LM(), based on Latin Multiplication, rejects almost the same number of requests. CSPF_U() rejects a slightly greater number of requests for p_0=8•10^{-4} and *L(LSP)*=8. Figure 5 shows the number of links in the dependence on load of link (in %). For CSPF and LM(), based on weights of link determined by (7), the number of loaded links from 80% to100% is smaller than for MIRA. For CSPF_U(), which chooses paths satisfying limitations ($L(LSP)$=8 and p_0=8•10^{-4}) the number of loaded links from 80% to100% is the same as for MIRA. Figure 6. shows the consumed bandwidth in the network obtained after using the considered algorithms for 10 different experiments. The stream of requests for each experiment is the same for all the considered algorithms. It results from the fact that the consumed bandwidth in the network for MIRA is comparable with the consumed bandwidth for LM(0,0008, 8). The consumed bandwidth for two other algorithms (CSPF and CSPF_U(0,0008, 8) is comparable, too (except 4th experiment), however, greater than the optimal solution.

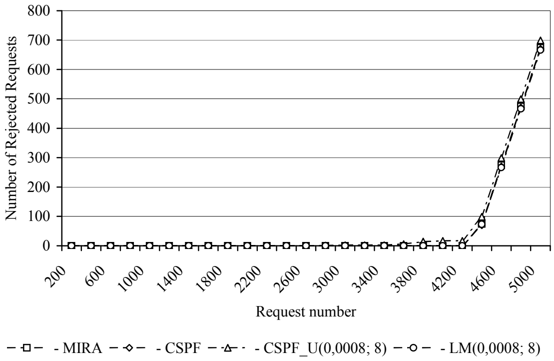

Fig. 4. Number of rejected requests vs. request number

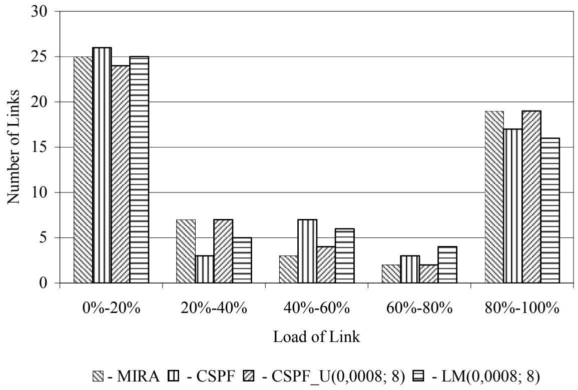

Fig. 5. Number of Links vs. load of link

Exactly, the consumed bandwidth for CSPF_U() is from 0.9 % to 1.4 % greater (except 4th experiment) than the consumed bandwidth for LM(). Figure 7. shows the number of rejected requests by the considered algorithms for the same 10 experiments. It should be noticed, that MIRA rejects more requests than CSPF. An CSPF_U(), which satisfies reliable limitation and length of path limitation rejects from 1.9% to 12.9% requests (for 5th experiment) greater than LM(). Figure 8. shows the running time of algorithms for the same 10 experiments. It should be noticed, that the time of choosing paths is critical for each *on-line* routing algorithm. The obtained results prove that CSPF is much faster than MIRA. The complexity function for CSPF is $O(n^2)$, whereas for MIRA is $O\left(n^4\sqrt{m}\right)$ [4]. CSPF_U(), which satisfies constraints (12) i (13) is also much faster then MIRA. The obtained results prove that the weighs of link in the proposed optimization algorithm (CSPF_U()), which takes into account reliable limitation and the length of path limitation are defined properly. It should be noted that in the proposed algorithm other weights are also considered, such as those used in the ILIOA [8].

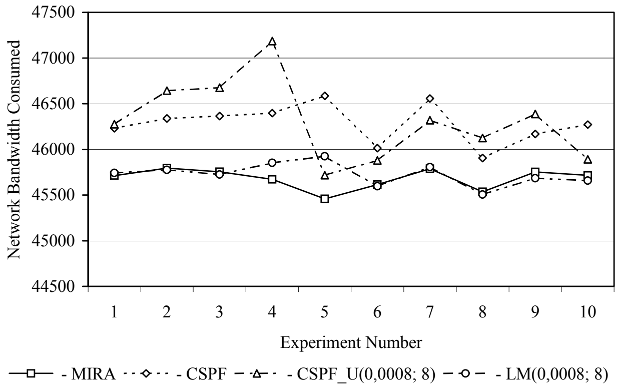

Fig. 6. Network bandwidth consumed for 10 experiments

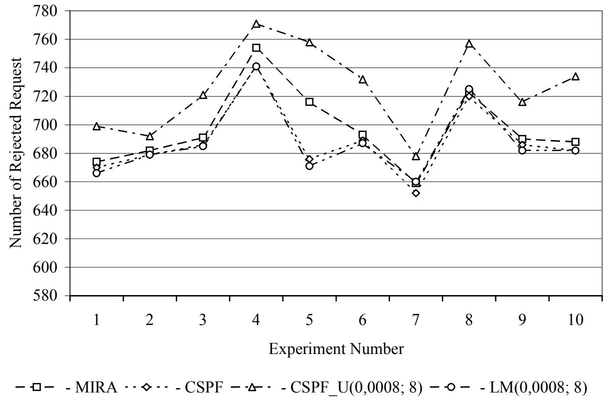

Fig. 7. Number of rejected requests for 10 experiments

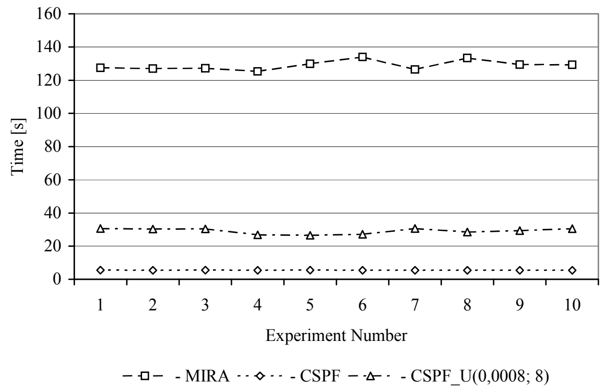

Fig. 8. The running time of algorithms for 10 experiments.

However, both, the number of rejected requests and the volume of the occupied bandwidth, was greater than in the case of link weights considered for the same value limitations. The shown results were obtained (by CSPF_U()) for p_0=8•10^{-4} and $L(LSP)$=8. Figure 9 shows the number of rejected requests for CSPF_U() and LM() for different lengths of paths: $L(LSP)$=6,7,8. It can be noticed that for $L(LSP)$=7,8 the number of requests rejected by CSPF_U() is comparable to the number of requests rejected by LM(). However for $L(LSP)$=6 CSPF_U() rejects from 63% to 127% more request than LM(). Such a great number of rejected requests results from the fact that CSPF_U() in next iterations, blocks the links *(i,j)*, which for a minimal increase of the path weight assure an increase in the path reliability. Therefore, with large limitation of the path length comparable to the shortest length of the path between a given pair of nodes (see Fig. 2.), CSPF_U() do not choose LSP, which satisfies constraints (12) i (13). Figure 10. shows the consumed bandwidth for CSPF_U() and LM() for the same constraint length of paths *L(LSP)*=6,7,8. Together with the increase in the number of requests rejected by CSPF_U() the amount of bandwidth consumed by LSPs decreases.

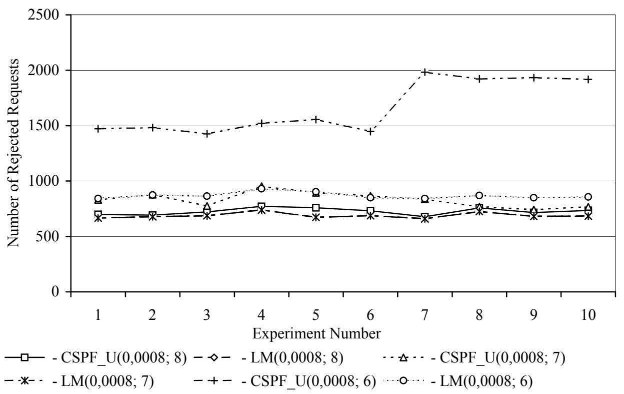

Fig. 9. Number of rejected requests for 10 experiments

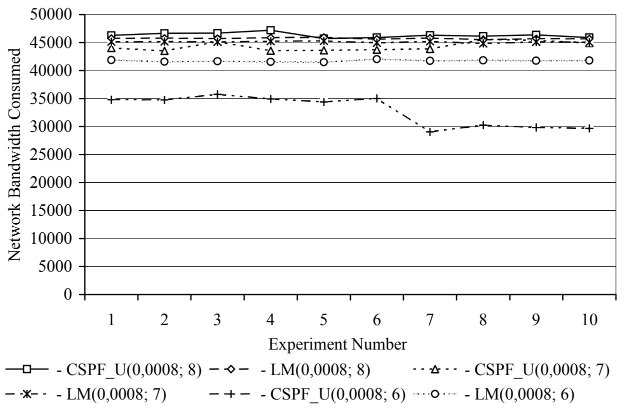

Fig. 10. Network bandwidth consumed for 10 experiments

Summary and Conclusions

In this paper an algorithm of choosing LSPs in the IP/MPLS network with unreliable network structure was proposed. This algorithm has been compared to algorithm giving optimal solution. The considered problem concerns reliability of paths limitation and length of paths limitation. The constrained length of path causes a decrease of lost packets in the network through shortening time of restoring of failure path on the backup path. In turn, the reliability limitation decreases the probability of failure of chosen paths. The accepted weights of links in the objective function are inversely proportional to residual bandwidth of links in the network.

CSPF_U() solving the formulated problem of optimization for given limitations determines a sub-optimal solution. To verify this algorithm, an algorithm based on Latin Multiplication was applied. The obtained results proved that the proposed CSPF_U() balances the load of links as well as MIRA. For this algorithm the number of rejected requests is slightly greater and chosen paths consume a slight greater bandwidth than for the exact algorithm. However, paths determined by the proposed algorithm satisfy the assumed limitations. CSPF_U() works well at reasonable values of limitations. For path length limitation comparable to the length of the shortest path, this algorithm rejects much more requests. Therefore, further works should determine the maximum length of paths in a dependence on the graph diameter and the number of the network links.

REFERENCES

[1] Autenrieth A., Kirstädter A., Engineering End-to-End IP Resilience Using Resilience-Differentiated QoS, *IEEE Communications Magazine*, January (2002).
[2] Calle E., Marzo Jose L., Urra A.: Protection Performance Components in MPLS Networks, *Computer Communications Journal*, 27 (2004), 12, 1220-1228.
[3] Bagula A.B., Botha M., Krzesiński A. E.: Online Traffic Engineering: The Least Interference Optimization Algorithm, *Communications, 2004 IEEE International Conference on,* 2 (2004), 1232-1236.
[4] Kodialam M., Lakshman T.V.: Minimum Interference Routing with Applications to MPLS Traffic Engineering, *Proc. IEEE INFOCOM 2000*, Mar. (2000), 884-893.
[5] Kotti A., Hamza R., Bouleimen K., Bendwidth Constrained Routing Algorithm for MPLS Traffic Engineering. *Third International Conference on Networing and Services*, ICNS, (2007).
[6] Krachodnok P., Constraint –Based Routing with Maximize Residual Bandwidth and Link Capacity-Minimize Total Flows Routing Algorithm for MPLS Networks. *Fifth International Conference on Information, Communications and Signal Processing,* (2005), 1509-1514
[7] Olszewski I., The Algorithms of Choice of the LSP path in the MPLS networks, *Kwartalnik Elektroniki i Telekomunikacji,* 50 (2004), n.1, 7-23.
[8] Olszewski I., The Improved Least Interference Routing Algorithm. *Paper accepted for the 2nd International Conference on Image Processing & Communications, (2010),* Bydgoszcz, Poland.
[9] Zhu M., Ye W., Feng S., A new dynamic routing algorithm based on minimum interference in MPLS Networks. 4th *International Conference on Wireless Communications, Networking and Mobile Computing. WiCOM '08*, (2008).
[10] Calle E., Marzo Jose L., A. Urra A., Villa P.: Enhancing MPLS QoS routing algorithms by using Network Protection Degree paradigm, *Proceedings of IEEE Global Communications Conference, GLOBELCOM* (2003), San Francisco.
[11]Kaufmann A.: Graphs, Dynamic Programming and Finite Games, *Academic Press*, (1967), 270-280.

Author: *dr inż. Ireneusz Olszewski, UniwersytetTechnologiczno-Przyrodniczy, Wydział Telekomunikacji i Elektrotechniki, Al. Prof. S. Kaliskiego 7, 85-796 Bydgoszcz, E-mail: Ireneusz.Olszewski@utp.edu.pl*