Ireneusz OLSZEWSKI

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

An algorithm of choosing LSPs based on the Dijkstra's Algoritm in the MPLS networks with unreliable links

Streszczenie. *W pracy zaproponowano algorytm wyboru ścieżek LSPs w sieciach MPLS o zawodnej infrastrukturze. Rozważany problem dotyczy minimalizacji opóźnienia w sieci przy ograniczeniu niezawodnościowym ścieżki LSP. Zaproponowany algorytm wyznacza rozwiązanie lokalne przy zadanych ograniczeniach. W celu weryfikacji tego algorytmu zastosowano ten sam algorytm z pominięciem ograniczenia niezawodnościowego oraz dobrze znany algorytm LIOA. Uzyskane wyniki dowodzą, że proponowany algorytm w warunkach dynamicznych odrzuca trochę więcej żądań, jednakże wybrane LSPs spełniają nałożone ograniczenia (**Algorytm wyboru LSPs oparty na algorytmie Dijkstry w sieci MPLS przy zawodnej strukturze sieci**).*

Abstract. *In this paper an algorithm for choosing LSPs in the MPLS network with unreliable links is proposed. The considered problem is to minimize network delay under reliability constraint imposed on the LSPs. The proposed algorithm, which is based on Dijkstra's algorithm, determines the local solution with the given constraints. In order to verify this algorithm the same algorithm without reliability constraint and well-known LIOA have been used. The obtained results show that the proposed algorithm under dynamic conditions rejects a little bit more requests, but the chosen LSP satisfies the imposed constraints.*

Słowa kluczowe: routing, wieloprotokołowa komutacja etykietowana, struktura sieci, ścieżka komutowana etykietowo.
Keywords: Routing, Multi Protocol Label Switching (MPLS), network structure, Label Switching Path.

Introduction

The network can be designed only for the assumed initial conditions, but 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) offered by the network is the network reliability. Although most of the problems considered in this paper concern the networks of different technologies using the conception of logical paths, so the further consideration will be focused on the IP network with Multi Protocol Label Switching (MPLS). Fault management mechanisms in MPLS networks are based on setting up the backup LSP [1]. In the case of a failure the traffic can be directed to that backup path. The backup LSPs can be realized as a static or dynamic [2]. In the first case the backup path is pre-established for the active path. In the second case the backup path is established as a result of failure (node or link) in the network.

Many algorithms for choosing LSPs in the MPLS networks have been proposed so far [3, 4, 5, 6, 7, 8]. All these algorithms minimize the number of rejected requests or the amount of consumed bandwidth in a network. For this purpose, the interference of the LSP being chosen from LSPs have already been chosen is minimised [3,5,6,7] or the interference of the LSP being chosen with LSPs for which requests of LSPs set up will come in the future is minimized [4, 8]. The algorithms are based on residual bandwidth and number of flows on the links in the network. Most of these algorithms, however, not take into account other aspects such as delay on the LSPs, and the links failure in the network. Minimizing delays in the network is very important due to the significant increase in the number of the real-time applications in the network, which are characteristic for the circuit-switched networks. The optimization problem, minimizing the amount of delays on the LSP, assuming that the set up of this LSP with a specified bandwidth, will not exceed maximum end-to-end delay for already admitted flows (LSPs), was formulated in [9]. However, the proposed admission control mechanism for the traffic flows in the MPLS networks requires an (exact) solution of the formulated optimization problem, which contains a set of restrictions imposed on the maximum end-to-end delay for all flows in the network. For solving the problem the CPLEX Mixed Integer Optimizer 9.0 [9] was used. It should be noticed, that the optimization algorithm used by the CPLEX module is based on the branch and bound method. Taking into account the fact that algorithms based on this method generally are characterized by high computational complexity, it is necessary to search for a heuristic algorithm with polynomial function of computational complexity, for a network with a larger number of nodes. However, heuristic algorithms provide approximate solutions, and generally require resignation from certain restrictions in the formulated optimization problem.

In this paper, an algorithm of choosing LSP in IP/MPLS networks, which minimizes the amount of end-to-end delay on the LSP, limiting the probability of LSP failure, has been proposed.

The paper is organised as follows: in the first part the optimization problem is formulated. In the second part the heuristics algorithm solving this problem is proposed. In the third part of the paper, the obtained results are given. In the final part the summary and conclusions are presented.

Formulation of optimization problem

Let *G(N, E, C)* be the network, where *N* is the set of nodes and *E* is the set of unidirectional links (arcs). *C* is *m* – vector of bandwidth of the links and *p* is the matrix, where element p_{ij} denotes probability of link *(i, j)* failure. Moreover, let *n* denote the number of nodes and *m* denote the number of links in the network. Let *R* be *m*-vector of residual capacity on the links. Furthermore, let the current request of setting up of LSP between a pair of nodes *s- t* be for *b* units of bandwidth. To simplify the notation, the link *(i, j)* and *l* will be used interchangeably.

Minimizing delay on the LSP requires determination of delays on particular links *(i, j)*, where $(i, j)\in E$. The total delay on link *(i, j)* includes the processing and queuing of packet delay at node *i* and the transmission and propagation delay on link *(i, j)*. To determine the queuing and transmission time the classical M/D/1 model has been used, where the input stream of packets of fixed length equal *l*, is Poissonian.

n [10] it was empirically shown that the packet delay for model $\sum D_i^{X_i}/D/1$ is upper bounded by the delay of packets for model M/D/1 (Figure 4 in [10]). Input process for model $\sum D_i^{X_i}/D/1$ is treated as the superposition of *K* independent constant rate streams, where both period and packet size can be different for each stream [10]. It should be noticed, that there is no analytical dependence describing the size of packet delay in this model. In turn, the total delay on link *(i, j)*, obtained on M/D/1 model can be defined as follows:

$$d_{i,j}\left(f_{i,j}\right)=\frac{1}{\lambda_{i,j}}\left(\frac{\varsigma}{1-\varsigma}-\frac{\varsigma^2}{2(1-\varsigma)}\right)+\alpha_{i,j} \tag{1}$$

$$\lambda_{i,j}=\frac{f_{i,j}}{l}$$

$$\varsigma=\frac{f_{i,j}}{c_{i,j}}$$

where: $\lambda_{i,j}$ is the mean arrival rate of packets on link (i, j), f_{ij} is the traffic flow (in bps) on link *(i, j)* with *b* units of bandwidth for the current request, ς is the average utilization of the link (system) and $\alpha_{i,j}$ is the sum of the processing delay at node *i* and the propagation delay on link *(i, j)*.

A probability of LSP failure which is determined on the basis of probabilities of corresponding links failure in the network, is the second factor that will be taken into account in this paper. The values of these probabilities can be calculated on the basis of any analysis of different statistics or the network operator experience [2]. Assuming that the probability of each link failure in the network is known, the probability of LSP failure can be determined as the probability of a complement event. The probability that LSP with length *L(LSP)* is in order can be determined as follows:

$$P(LSP\ is\ in\ order\ )=$$
$$\prod_{e_i\in LSP}(1-p_i)=1-\sum_{j=1}^{L(LSP)}(-1)^{j-1}\sum_{e_{i_1},e_{i_2},\ldots,e_{i_j}\in LSP}p_{i_1}p_{i_2},\ldots,p_{i_j} \tag{4}$$

where: p_i is the probability of failure of link e_i.

The probability that LSP is a failure can be written as follows:

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

for adequate small p_i. Routing algorithm will choose the LSP with minimal delay, for which the probability of LSP failure is not greater than threshold p_0, which was assumed for all LSPs. Lower limit of probability of LSP failure generally causes limitation of the number of links requiring protection on LSP. Used in [9], end-to-end delay constraints for all traffic flows in the network have been omitted in the formulated optimization problem, due to the lack of possibility of inclusion of these constraints in the proposed algorithm. It should be noticed that the omission of these restrictions may lead to changes of delay for 8% ÷ 12% of the flows, depending on the number of nodes in the network [9]. After discussion of the delay on the links and at the nodes of the network and assumed reliability constraints, the optimization problem can be formulated as follows:

$$\min\sum_{(i,j)\in E}d_{i,j}x_{i,j} \tag{6}$$

$$\sum_j x_{i,j}-\sum_j x_{j,i}=0 \qquad \forall_{i,\,j\in N} \tag{7}$$

$$\sum_j x_{s,j}-\sum_j x_{j,s}=1 \qquad \forall_{s,\,j\in N} \tag{8}$$

$$\sum_j x_{t,j}-\sum_j x_{j,t}=-1 \qquad \forall_{t,\,j\in N} \tag{9}$$

$$\sum_{(i,j)\in E}x_{i,j}p_{i,j}\le p_0 \tag{10}$$

$$R_{i,j}x_{i,j}\ge b \tag{11}$$

$$x_{i,j}\in\{0,1\} \qquad \forall_{(i,\,j)\in N} \tag{12}$$

Vector *x* represents the flow on the path between the pair of nodes *(s, t)*, where $x_{i,j}$ is set to 1, if link *(i,j)* is used on the path. Formula (6) defines the optimized objective function with weights, which are defined on the bases of (1). Equations (7)-(9) give the flow balance for LSP. Inequality (10) is a constraint of probability of LSP failure, whereas inequality (11) is a constraint imposed on the amount of residual bandwidth. Formulated LSP path selection problem with minimum delay is the integer linear programming problem. Because the algorithm solving this problem must work *on-line* for the networks with large numbers of nodes, a heuristic approach can be considered only.

An heuristic algorithm–sub-optimal solution

The algorithm solving the optimization problem formulated for residual capacities should optimize the objective function given in (1) while maintaining the constraint on the probability of LSP failure. For a given weight matrix *D*, whose individual elements d_{ij} determine the delay on links *(i, j)*, the shortest path, that is the path with minimum delay, is determined on the basis of a modified version of Dijkstra's algorithm. Modification of this algorithm, also outlined in [11], is based on introducing the reliability constraint to the chosen LSP. The general idea of Dijkstra's algorithm [12] is based on the movement on the network arcs, in the subsequent iterations, from source node *s* to terminate node *t* and marking the intermediate nodes by their current distances from node *s.* The feature of node *u* is fixed when it is equal to the length of the shortest path from node *s* to *u*. During the initialization of the algorithm the source node *s* receives a fixed feature. Then, in the first iteration, a temporary feature of each successor *v* of node *s* is changed from infinity to the feature equal weight of arc d_{sv}. The node with the smallest feature of a temporary node, for example node *u*, is replaced by a fixed feature, which does not change until the end of the work of the algorithm. In the next iteration the successors of node *u* are featured. Then, as before, the node with the smallest temporary feature of all, receives a fixed feature. The algorithm terminates when the final node *t* receives a fixed feature. In each iteration of the algorithm the value of the temporary features is reduced. Let *dist* be an *n*-element vector, where element *dist(v)* is the distance from the source node *s* to node *v* and *pred* is a vector of the predecessor on the shortest path from node *s* to node *t.* Furthermore, let the variable ***newlabel*** be the value of feature of temporary node *v*, determined from node *u* for which the feature has recently been established, i.e. *newlabel←dist(u)+*$d_{u,v}$. If the value of feature of node *v*, i.e. the distance from node *s* to node *v* through node *u* is reduced, then *dist(v) ← newlabel* and *pred(v) ←u*. At this point, the algorithm should be modified. Let variable *newprob*, analogically to *newlabel,* be the probability of path failure from node s to node v passing through node u and variable *prob* be n-element vector, whose element *prob(v)* is the probability of a path failure from node s to node v. Let the value of variable *newprob* be determined, i.e. $newprob \leftarrow prob(u)+p_{u,v}$. If feature node v is reduced, i.e. the path from node s to node v through node u is reduced, then the probability of path failure should not be greater than the maximum probability of LSP failure p_0 from node s to t. The modified Dijkstra's algorithm, denoted as M_Dijkstra(), for choosing the LSP with the minimum delay and satisfying reliability constraint (10) is shown below. Let *final* be an n-element vector of boolean, where *final(i)* changes its state from *false* to *true* when feature node i is changed from temporary to fixed.

M_Dijkstra(p_0);
begin {begin M_Dijkstra()}
  for $v \in V$ **do begin** $dist(v) \leftarrow \infty$; $final(v) \leftarrow false$; **end**;
  $dist(s) \leftarrow 0$; $final(s) \leftarrow true$; $u \leftarrow s$;
  for $(i,j) \in E$ **do**
   if $R_{ij} < b$ **then** $d_{i,j} \leftarrow \infty$ **else** $d_{i,j} \leftarrow$ delay on the base (1);
  while *final(t) = false* **do**
   for each direct successor v of node u if not *final(v)***do**
    begin
     $newlabel \leftarrow dist(u)+d_{u,v}$;
     $newprob \leftarrow prob(u)+p_{u,v}$
     if ($newlabel < dist(v)$) **and** ($newprob \le p_0$) **then**
      begin {**c**hange the shortest path to node v}
       $dist(v) \leftarrow newlabel$;
       $prob(v) \leftarrow newprob$;
       pred(v):=u;
      end;
    end;
find a node y with the smallest temporary feature, different from ∞;
   final(y) $\leftarrow$true; {y receives a fixed feature}
   $u \leftarrow y$;
  end
end; {end M_Dijkstra()}

The delay on LSP is in variable *dist (t)*, (if $dist\ (t) = \infty$ then the request of LSP set up is rejected) and the course of the path can be obtained on the basis of vector *pred*. The complexity function of the algorithm is $O(n^2)$.

Obtained results

Verification of the algorithm has been made for two networks containing respectively 15 and 23 nodes. The first network whose topological structure is shown in figure 1a. [4] contains 15 nodes (routers), connected by links with the capacity of 155 Mbps (thin lines) and 620 Mbps (thick lines). The second network [3], whose topological structure shown in figure 1.b contains 23 nodes connected by links with the same values: 155 (thin lines) and 620 Mbps (thick lines). Capacities of the links in both networks has been multiplied by an appropriate scaling factor ω. Each link is unidirectional. In this paper it has been assumed that each node can be input and output node. Therefore, in the first network, 210 pairs of nodes can be distinguished, while in the second one 506 pairs of nodes. The values of link failure probability p_{ij} for each link (i, j) are shown on arcs of the graph, while the acceptable probability of the LSP failure p_0 varies from 0.0006 to 0.0012.

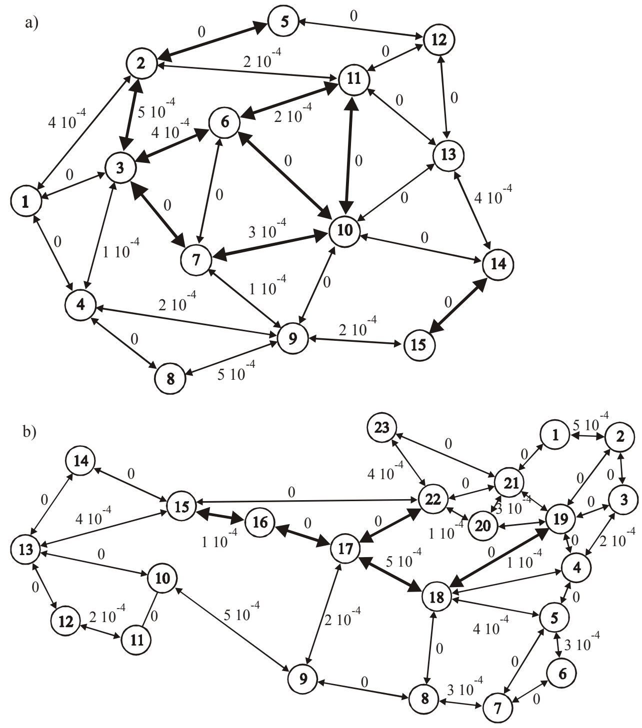

Fig. 1. Topology structure of the considered networks: a) the MIRA network b) the LIOA network

Simulation results obtained on the basis of the presented algorithm M_Dijkstra() were compared with results obtained on the basis of two algorithms. First of them is the same algorithm without a constraint imposed on the probability of failure, i.e. Dijkstra's algorithm with weights of the links assigned on the basis of the M/D/1 model; while the second is the well known Least Interference Optimization Algorithm (LIOA). In LIOA, weights of links are proportional to the number of flows (the number of LSPs) realized on these links and inversely proportional to their residual capacity [3]. Network simulation was made using the Monte Carlo method. It was assumed that the stream of requests of LSPs set up between each pair of nodes *(i, j)* is Poissonian, with intensity λ and the holding time of the LSP is exponentially distributed with mean value $1/\mu = 1$. Bandwidth of LSPs is uniformly distributed from 3 to 5 Mbps. The network simulation has been done in static and dynamic conditions. In static conditions LSPs are only set up (long lived connections), but in dynamic conditions LSPs are set up and disconnected (short lived connections). In static conditions, for a given simulation trial, all algorithms are verified for the same stream of requests of LSPs choice. In dynamic conditions the results are recorded after obtaining an equilibrium state of the system. Both in static and dynamic conditions, tests have been done for T=10 trials. The number of requests of LSP set up in the network is a condition for the end of the simulation. Below, figure 2 to figure 8 show the results obtained after using these three algorithms for networks containing 15 and 23 nodes. In both networks a different coefficient for scaling capacity was used: for the network with 15 nodes, operating in static conditions (long lived connections) ω=8, while in dynamic conditions (short lived connections) ω=1. In turn, for a network consisting of 23 nodes operating in static conditions ω=10, while in dynamic conditions ω=3. During the simulation of any network, for each algorithm, T = 10 trials were done. In the dynamic conditions, for a network consisting of 23 nodes (LIOA network), each trial included

50000 requests, while for the network containing 15 nodes (MIRA network) each trial included 20000 requests. However, in static conditions each trial included 10000 requests regardless of the network size. Moreover, it was assumed that in the LIOA coefficient α =0.5 [3].

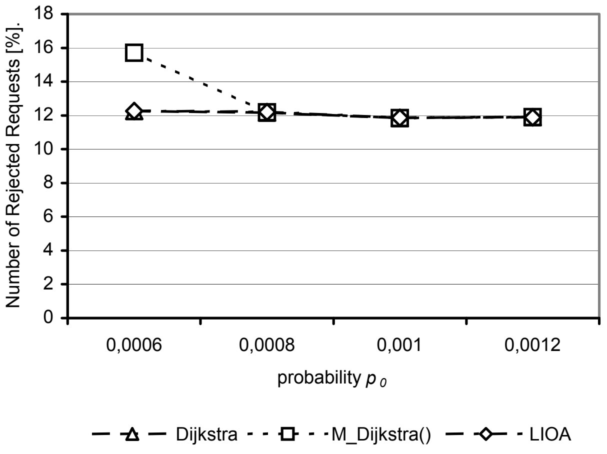

Fig.2. Number of rejected requests vs. probability p_0 for the network with 15 nodes in static conditions

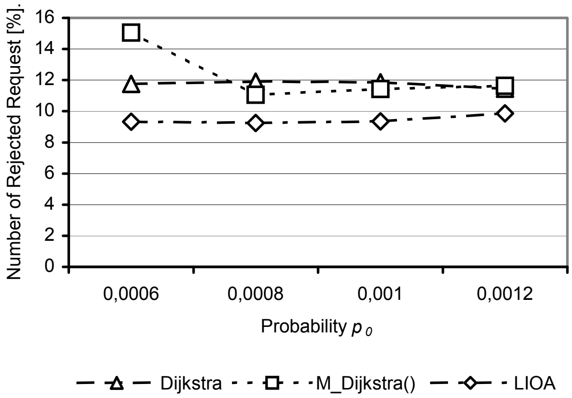

Fig.3. Number of rejected requests vs. probability p_0 for the network with 15 nodes in dynamic conditions

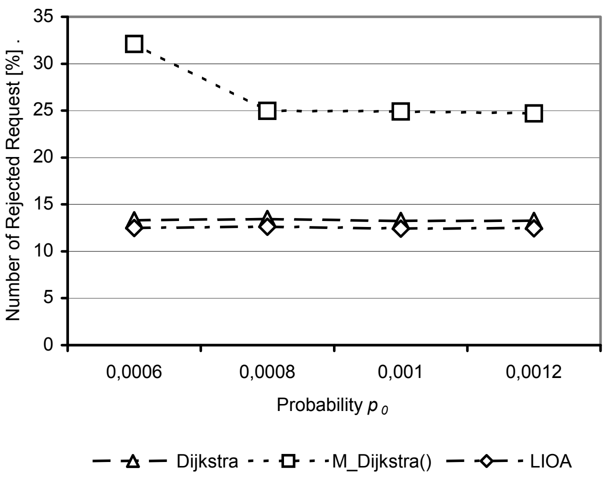

Fig.4. Number of rejected requests vs. probability p_0 for the network with 23 nodes in static conditions

Figure 2 shows the number of rejected requests (in%) depending on p_0, for the network with 15 nodes, in static conditions. From figure it results that for $p_0 \geq 0.8$ the number of rejected requests is convergent for all these algorithms. It should be noticed that the probability of link failure ranges from 0.0001 to 0.0005 for 24 links.. So, for p_0 =0.006, two or three links for which $p_i \neq 0$ can eliminate the possibility of LSP choice. Figure 3 also shows the number of rejected requests (in%) depending on p_0, for the same network, in dynamic conditions. From this figure it results that convergence of the number of the rejected requests is for $p_0 \geq 0.8$. In this case, the LIOA marks its slight advantage (less than 2%) over the two other algorithms. Figure 4 and figure 5 show the same relationship for the network with 23 nodes. It should be noticed, that between about 40% of pairs of nodes *s-t*, the LSPs can be set up and these LSPs must pass through a cut containing 3 - 4 links (eg. cat {(15-22) (15-16) and (10 - 9)}), for which $p_i \neq 0$ (see figure 1.b). In addition, the setting up of LSPs consists of several links (more then 10).

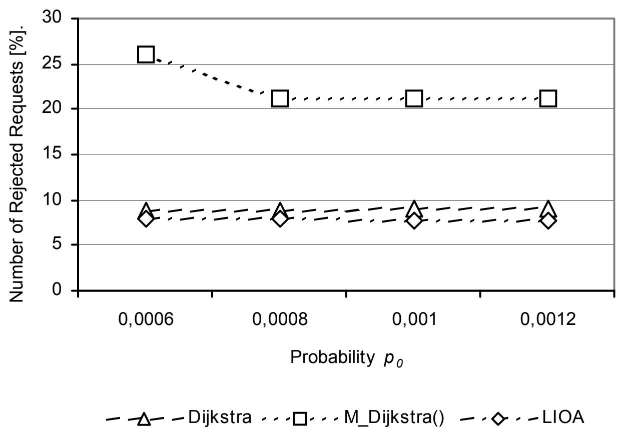

Fig.5. Number of rejected requests vs. probability p_0 for the network with 23 nodes in dynamic conditions

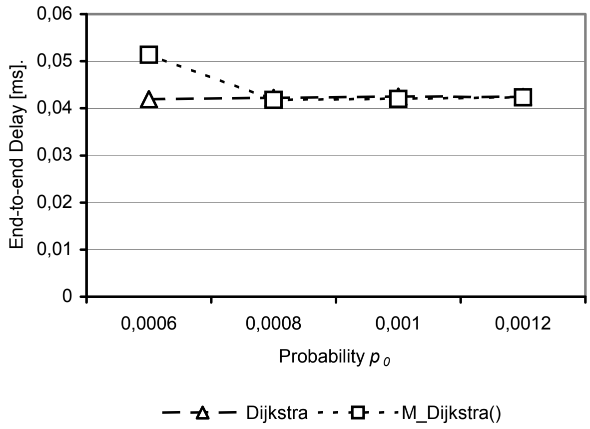

Fig.6. Average delay on the LSP vs. probability p_0 for the network with 15 nodes in static conditions

Therefore, a request to choose such a long path, consisting of two or more links for which $p_i \neq 0$, is rejected. Therefore, the proposed algorithm M_Dijkstra(), which includes reliability constraint (10), rejects more requests in both the static and dynamic conditions for this type of network. Figure 6 shows the average delay on LSPs in the network with 15 nodes in static conditions. From this figure it results that for $p_0 \geq 0.8$ the average delay on the paths chosen by M_Dijkstra() coincides with the delay on the paths chosen by the algorithm that does not include reliability constraint. The decrease in the delay stems from the fact that with increasing p_0 M_Dijkstra() algorithm chooses a shorter paths, minimizing the value of the objective function. Figure 7 shows the average delay on the LSPs depending on p_0 for the network consisting of 23 nodes. From this figure it results that the value of delay on the paths chosen by the M_Dijkstra () for $p_0 \geq 0.8$ decreases below the delay in the paths chosen by the Dijkstra's algorithm. The decrease in this delay results from the increased number of rejected paths (see figure 4) which have larger length, measured in number of links, causing a decrease in delay on shorter paths. A similar dependence of the delay on LSPs versus p_0 has been obtained for the network in dynamic conditions (figure 8).

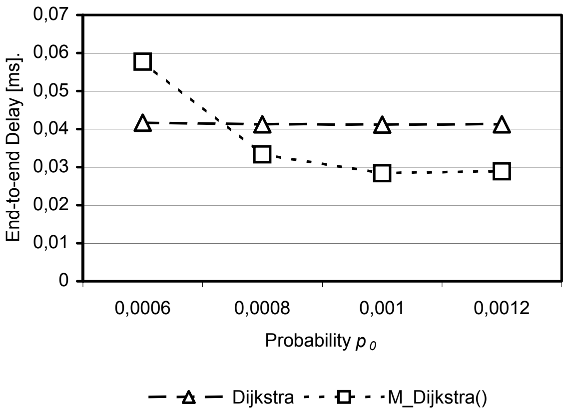

Fig.7. Average delay on the LSPs versus probability p_0 for a network consisting of 23 nodes in static conditions

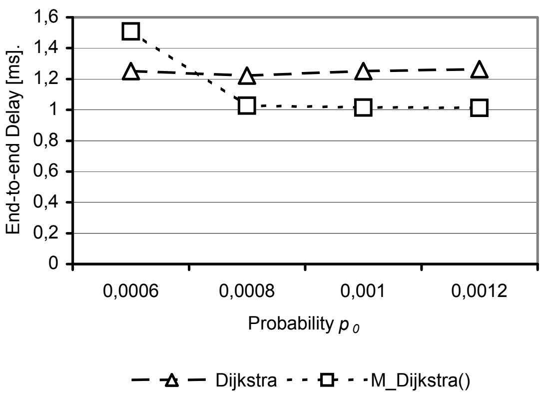

Fig.8. Average delay on the LSPs versus probability p_0 for a network consisting of 23 nodes in dynamic conditions

Summary and conclusions

In this paper an algorithm of choosing LSPs in the IP/MPLS network with unreliable network structure has been proposed and a comparison of this algorithm with an algorithm that does not include reliability constraint and LIOA has been made. The considered problem involves minimizing delays on the LSPs under reliability constraint which reduces the probability of failure of chosen LSPs. The accepted weights of links in the objective function representing delay at the node, and delay on the link of the network are determined basing on the M/D/1 model. The obtained numbers of rejected requests show that the algorithm based on weights of links assigned on the basis of M/D/1 model provides the same solutions under static conditions and comparable solutions under dynamic conditions as the solutions obtained on the basis of a sophisticated routing algorithm.

For the proposed algorithm M_Dijkstra(), which minimizes the value of delay, the number of rejected requests is slightly higher but the chosen LSPs satisfy the restrictions imposed on reliability.

Further works should include a larger number of restrictions, such as length of the paths in a dependence on the graph diameter and the number of network nodes.

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 Network, *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. In: *Proceeding of IEEE ICC* (2004), 1232–1236
[4] Kodialam M., Lakshman T.V., Minimum Interference Routing with Applications to MPLS Traffic Engineering. *Proc. INFOCOM*, Mar. (2000)
[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 Improved Least Interference Routing Algorithm. *2 nd International Conference on Image Processing & Communications*, (2010), Bydgoszcz.
[8] 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)
[9] Oulai D., Chamberland S., Pierre S., A New Routing-based Admission Control for MPLS Networks, *IEEE Communications Letters*, 11,(2007), n.2, 216-218
[10] Bonald T., Proutière A. and Roberts J.W.: Statistical Performance Guarantees for Streaming Flows using Expedited Forwarding, *INFOCOM* (2001)
[11] Salcedo Parra J., Manta C., Lopez Rubio G., Dijkstra's Algorithm Model over MPLS/GMPLS, *7 th International Conference on Wireless Communications, Networking and Mobile Computing WiCOM* (2011)
[12] Sysło M.M.,N. Deo, J.S.Kowalik, Algorytmy optymalizacji Dyskretnej, *PWN*, Warszawa (1995)

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