STUDIA INFORMATICA 2015
Volume 36 Number 4 (122)

Mariusz GOLA
Institute of Automatic Control and Informatics, Opole University of Technology
Adam CZUBAK
Institute of Mathematics and Informatics, Opole University

A NOTE ON CAPACITY PROVISIONING IN SDN SERVICE PROVIDER NETWORKS

Summary. In this paper we take a closer look at ISP networks that have a centralized SDN control plane. A heuristic algorithm for the CFA problem for such networks is proposed, that takes under account two factors: the speed of individual links and ISP network's given properties.

Keywords: WAN, routing, CFA, SDN

O PRZYDZIELANIU PASMA W SIECIACH ISP STOSUJĄCYCH TECHNOLOGIĘ SDN

Streszczenie. W niniejszej pracy rozważane są sieci ISP z wdrożoną warstwą sterowania typu SDN. Zaproponowano algorytm heurystyczny rozwiązujący problem CFA dla tego typu sieci, biorąc pod uwagę dwa aspekty: przepustowość łączy oraz cechy własne sieci ISP.

Słowa kluczowe: sieci WAN, trasowanie, CFA, SDN

1. Introduction

Software-Defined Networks [1, 10, 12] introduce to contemporary network management several ideas and concepts:

- There is a separation between the devices which forward the traffic (data plane) and the software that is used to configure the forwarding policies;
- The deployment of configuration to the devices is performed in a non-vendor specific manner with some open APIs;

- There is a centralized platform with an abstract overview of the whole network.

While in the field of Data Center networks Software-Defined Networking is gaining more and more market share and LAN networks show the same constant increase in number of SDN solution deployments, ISP networks remain more conventional and do not implement new technologies as rapidly as the two other. This is due to several aspects:

- the sheer scale of ISP networks, ie. the amount of specialized vendor-specific L2 and L3 switches in the infrastructure makes it simply impossible to introduce new technologies, like SDN, in a timely manner;
- the robustness of the currently deployed refined solution is usually very high and network architects are reluctant to make so significant changes to the perfectly working, well known infrastructure;
- although SDN is not a new concept anymore, the protocols and tools available at the moment are often work-in-progress solutions and are expected to change several times in the near future. So ISP network architects, for good reason, still prefer well known conventional solutions to a new exciting but still evolving new technology.

Nevertheless, SDN if introduced, would either improve, add or simplify the following ISP network functionalities:

- overall management: a view of the entire network from a centralized platform with the use of underlying open configuration protocol (like openFlow [11]) would consolidate management of devices from different vendors. An addition or removal of a single device and troubleshooting might be performed in a unified way regardless of the device manufacturer;
- implementation of specialized functionalities like QoS, VPN or Traffic Engineering might also be performed in a unified and centralized manner;
- currently, introduction of new features to an ISP network is extremely hard (like for example introduction of SDN itself). But SDN adds a notion of programmability to the network since we are able to view the whole networks from a central location and configure individual devices from there [21]. So the very deployment of new features would be fairly trivial. And this adds a great deal of flexibility for an ISP network architect, which is invaluable in solving real-life networking issues.

One of the main administrative tasks in an ISP network is the joint optimization of link capacities and flow assignment (routing) also known as the Capacity and Flow Assignment Problem (CFA). Several different formulations of this problem can be found in the literature. Generally, they correspond to different choices of performance measures, design variables, constraints and different types of flows [6, 14, 15]. Exact algorithms for the CFA problem are presented in the papers [5-7, 14]. In literature a heuristic approaches, which is less exact but significantly faster, for solving these problems could be found as well. Some approaches to CFA problem are presented in [8, 13, 17]. Some aspect of top down approach to TCFA (Topology Capacity Flows Assignment) problem was presented in [9].

Now, the ISP networks are designed to handle client traffic up to the amount noted in the service agreement. This forces an infrastructure that is designed to forward traffic of clients up to a certain level. For example if an ISP offers internet access of the speeds: 1,2,5,10 MB as quite affordable, the network is prepared to handle any of these speeds without the need of tampering with the equipment and a change in the clients agreement doesn't require an intervention in the network. It doesn't matter which exactly of the available bandwidth the client chooses, the network is designed to handle it so it is overprovisioned. But it is of importance if a group of clients requests a much higher bandwidth like 100 MB. Questions arise like: which link to use and why?

The links in an ISP network:

have a fixed throughput;
belong to that ISP;
do not belong to an ISP but are shared or leased from a different ISP.

Let us presume, that we wish to add a new functionality to an ISP SDN network. In an SDN network an application runnnig on top of an SDN controller may solve the CFA problem, since a controller has a view of the topology and an open protocol for device configuration. One of the factors taken into account will be the delay in the network and the other will be the preference that a network architect has regarding which link to use (owned by the ISP) and witch preferably not to use (leased from another ISP).

Since CFA is in NP-Complete class of problems [2] we turn to heuristic methods. As mentioned before, a criterion function we chose is a function that considers both: channel preference and network performance. Network performance is represented by average delay per packet. The routing problem is perceived as the multi-commodity flow problem [4,14]. To present the criterion function in the unified way, the following two distinct types of costs are considered:

- channel cost: $c = \frac{1}{pref}$ involves the preference of a particular link channel used in the network;
- delay cost: total average delay per packet times unit cost of the delay.

The introduced criterion is the sum of the channel cost and the delay cost and it is called the combined cost. Thus, the optimization problem with the combined cost criterion is formulated as follows:

given: *network topology, external traffic requirements, discrete cost-capacity function, maximal admissible delay per packet*

minimize: *combined cost*

over: *channel capacities, routing (i.e. multicommodity flow)*

subject to: *multicommodity flow constraints, channel capacities constraints, delay per packet constrains*

Channels' capacities constraints mean that capacity for channels could be chosen only from the set of discrete values of capacities. In this case the considered topology design problem is NP-complete [2].

2. Problem formulation

Consider a WAN with *n* nodes and *b* channels which may be used to build the network. For each channel *i* there is the set $ZC^i = \{c_1^i, c_2^i, ..., x_{e(i)}^i\}$ of alternative capacities from which exactly one must be chosen. e(*i*) is the number of capacities available for channel i. Let d_k^i be the value of cost of capacity c_k^i for channel i. Let capacities from each set ZC^i be ordered in such way that

$$c_1^i \geq c_2^i \geq ... \geq c_{e(i)}^i \tag{1}$$

Let x_k^i be the decision variable, which is equal to one if capacity c_k^i is assigned to channel i and x_k^i is equal to zero otherwise. Let $W^i = \{x_1^i, ..., x_{e(i)}^i\}$ be the set of all variables x_k^i which correspond to the channel *i*. Exactly one capacity from the set ZC^i must be chosen for channel i, then

$$\sum_{k=1}^{e(i)} x_k^i = 1 \quad for \quad i = 1,...,b. \tag{2}$$

Let $W = \bigcup_{i=1}^{b} W^i$ be the permutation of values of all variables $x_k^i, k = 1,...,e(i), i = 1,...,b$ for which the condition (2) is satisfied, and let X_r be set of variables which are equal to one in W. The set X_r is called a selection. Then, each selection determines the unique values of channel capacities in the wide area computer network. Let $\Re$ be the family of all selections X_r.

Let $T(X_r)$ be the minimal average delay per packet [s/packet] in the wide area computer network in which values of channel capacities are given by the selection X_r. The expression on an average delay per packet is given by Kleinrock's formula [4, 14]. $T(X_r)$ could be obtained by solving a multicommodity flow problem in the network with the channel capacities given by X_r [4]. If this problem has no solution for a given requirement matrix and for the given selection X_r, then $T(X_r) = \infty$.

Let $d(X_r)$ be the sum of cost of capacities given by selection X_r, i.e.

$$d(X_r) = \sum_{x_k^i \in X_r} x_k^i d_k^i \tag{3}$$

Then, the objective function consisting of two distinct types of costs (delay cost and capacity cost) is following:

$$Q(X_r) = \alpha T(X_r) + d(X_r) \tag{4}$$

where α is the unit cost of delay

The cost capacity assignment problem with the combined cost criterion function can be formulated as follows:

$$\min_{X_r} Q(X_r) \tag{5}$$

subject to

$$X_r \in \Re \tag{6}$$

$$T(X_r) \le T_{\max} \tag{7}$$

Where $T_{\max}$ is maximal admissible average delay per packet. In other words $T_{\max}$ is responsible for performance condition, We couldn’t accept solution for which value of average delay per packet exceeds $T_{\max}$. In the paper [7] exact algorithm for solving above problem was presented.

3. Calculation scheme

Let $X_1 = \{x_1^1, x_1^2, ..., x_1^b\}$ be the initial selection. According to expression (1) the representation X_1 consists of channels set to maximal possible capacities. The variable x_k^i is called normal if $x_k^i \in X_r$. The remaining variables are called reverse. A replacement of any normal variable x_k^i by reverse variable from W^i is called complementing. Each selection

X_{r+1} is obtained from a selection X_r by complementing one variable x_k^i by another variable x_j^i from the same set W^i. In [7, 8] is proven that we should choose for complementing such variables $x_k^i \in X_r$ and $x_j^i \in X_s$ for which the value of Δ_{kj}^{ir} is maximal and condition (6) is satisfied. Where

$$\Delta_{kj}^{ir} = \begin{cases} \dfrac{\alpha}{\gamma}\left(\dfrac{f_{ir}}{c_k^i - f_{ir}} - \dfrac{f_{ir}}{c_j^i - f_{ir}}\right) + \left(d_k^i - d_j^i\right) if\ c_j^i \geq f_{ir} \\ \dfrac{\alpha}{\gamma}\left(\dfrac{f_{ir}}{c_k^i - f_{ir}} - \dfrac{c_{\max}^{ir}}{\varepsilon}\right) + \left(d_k^i - d_j^i\right)\ otherwise \end{cases} \tag{8}$$

and f_{ir} is equal to the optimum value of f_i which is obtained by solving the multicommodity flow problem for values of channel capacities given by X_r. ε is the minimal feasible difference between capacity and flow in each channel $(0 < \varepsilon \leq 1)$ and $c_{\max}^{ir}$ is maximal available capacity for channel *i*.

The algorithm starts with representation X_1 with a maximal channel cost and with minimal average delay per packet. If X_1 fulfills constraint (7), in next step we try to find a 'better' solution with lower value of criterion function. If for X_1 condition (7) is not satisfied it follows that no solution to (5-7) exists.

For complementation we take variables x_j^i which correspond to opposite arcs (with lower capacities and thus lower costs). Every succeeding representation generated by the algorithm has a lower cost $d(X_r)$ and could have greater value of average delay per packet $T(X_r)$.

4. Heuristic algorithm

Now we introduce an approximate algorithm used to solve the optimization problem (5-7). *M* is the set of reverse variable. If $M = \varnothing$ then we have no variables for complementing. *K* is the set of fixed normal variables; *F* is the set of fixed reverse variables. Fixed variables cannot be used for complementing for each successor X_r. *W* is the permutation set of values of all variables and was introduced in problem formulation section.

Step 1

Perform $X_1 = \{x_1^1, x_1^2, \ldots, x_1^b\}$,

If $T(X_r) > T_{\max}$ then no solution for optimization problem (5-7) exists, stop algorithm

Otherwise perform $Q^* = Q(X_1)$, $F = \varnothing$, $K = \varnothing$, r=1, go to step 2.

Step 2

Perform $M = W - X_r - F$; If $M = \varnothing$ then go to step 5 otherwise go to step 3.

Step 3

Choose the variable $x_k^i \in X_r$ and variable $w_j^i \in M$ for which the value Δ_{kj}^{ir} is maximal.

Next generate the selection X_{r+1} by complementing x_k^i by x_j^i by letting $X_{r+1} = (X_r - \{x_k^i\}) \cup \{x_j^i\}$,

If $T(X_r) \le T_{\max}$ compute $Q(X_{r+1})$ then go to step 4

Otherwise $F = F \cup \bigcup_{t=j}^{t \le e(i)} \{x_t^i\}$ then go to 2

Step 4

If $Q(X_{r+1}) < Q^*$ then perform r=r+1, $Q^* = Q(X_r)$, $F = F \cup \bigcup_{t=k}^{t<j} \{x_t^i\}$ go to step

Step 5

X_r is approximate solution of problem (5-7).

5. Computational results

The presented approximate algorithm was implemented in C++ code. Extensive numerical experiments have been performed with this algorithm for many different network topologies, different set of possible capacities for each of the channels and different workload. The experiments were conducted to examine the distance between an approximate and exact solution. For example a network with 12 channels, each channel with 9 possible capacities required an execution time by an exact algorithm of a few hours or more. Presented approximate algorithm for the same network gave an acceptable solution within 35 milliseconds.

Let us introduce definitions of certain magnitude, which will be indispensable to presenting properties of optimization problem. Let:

$T_{\min}$ be the minimal average delay per packet [s/packet] in the wide area computer network in which values of channel capacities are given by the selection X_1. $T_{\min}$ is the lowest possible value of average delay per packet at given requirement matrix R.

$D_{\max}$ be the maximal cost of the network, its sum of channel cost for representation X_1.

$\overline{T}_{\max} = \frac{T_{\max}}{T_{\min}}$ be the normalized maximal admissible average delay per packet and

$\overline{\alpha} = \alpha \cdot \frac{T_{\min}}{D_{\max}} \cdot 100\%$ be the normalized coefficient α .

Let $\Delta_Q = \frac{Q_{app} - Q_{opt}}{Q_{opt}} 100\%$ be the distance between approximate solution and exact solution, where: Q_{app} is value of combined cost criterion received as approximate solution of optimization problem (5-7). Q_{opt} is the value of combined cost criterion received as exact solution of optimization problem (5-7).

Let $\Phi_{(\overline{\alpha},\overline{B})} = \sum_{i=1}^{P} \frac{\Delta_Q^i}{i}$ be the arithmetic mean of the distance between an approximate and exact solution calculated for all considered networks and for normalized coefficient $\overline{\alpha}$ and normalized maximal admissible average delay per packet $\overline{T}_{\max}$. P is the number of considered network topologies. The dependence of Φ on normalized budget $\overline{T}_{\max}$ has been examined. In Table 1 values of Φ for different values of $\overline{T}_{\max}$ and $\overline{\alpha}$ are presented.

Table 1

Distances between exact and approximate solutions

$\overline{T}_{\max}$	Normalized $\overline{\alpha}$					
	1%	3%	5%	10%	20%	$\overline{\alpha} > 30\%$
1,2	14,9%	11,0%	7,3%	2,6%	0,8%	For normalize $\overline{\alpha}$ greater than 30% value of Φ is lower than 0,5 %
1,4	12,4%	9,0%	6,9%	2,5%	0,5%	
1,6	12,2%	9,0%	6,8%	1,6%	0,5%	
1,8	12,3%	9,0%	7,2%	1,3%	0,5%	
2,0	12,3%	9,0%	6,7%	1,2%	0,5%	
2,2	12,0%	9,0%	7,0%	1,2%	0,5%	
2,4	11,8%	9,0%	7,1%	1,2%	0,5%	
2,6	11,7%	9,0%	6,5%	1,2%	0,5%	
2,8	11,8%	9,0%	6,8%	1,2%	0,5%	
3,0	11,6%	9,0%	6,5%	1,2%	0,4%	

First we notice that approximate algorithm works very well as it is very close to optimal solutions for normalized $\overline{\alpha}$ greater than 20% and it is almost independent from value of $\overline{T}_{\max}$. For $\overline{\alpha}$ lower than 5% distance between approximate and exact solution is greater than 5%.

On Figure 1 the dependence of Φ on $\overline{T}_{\max}$ for different value of normalized $\overline{\alpha}$ is presented.

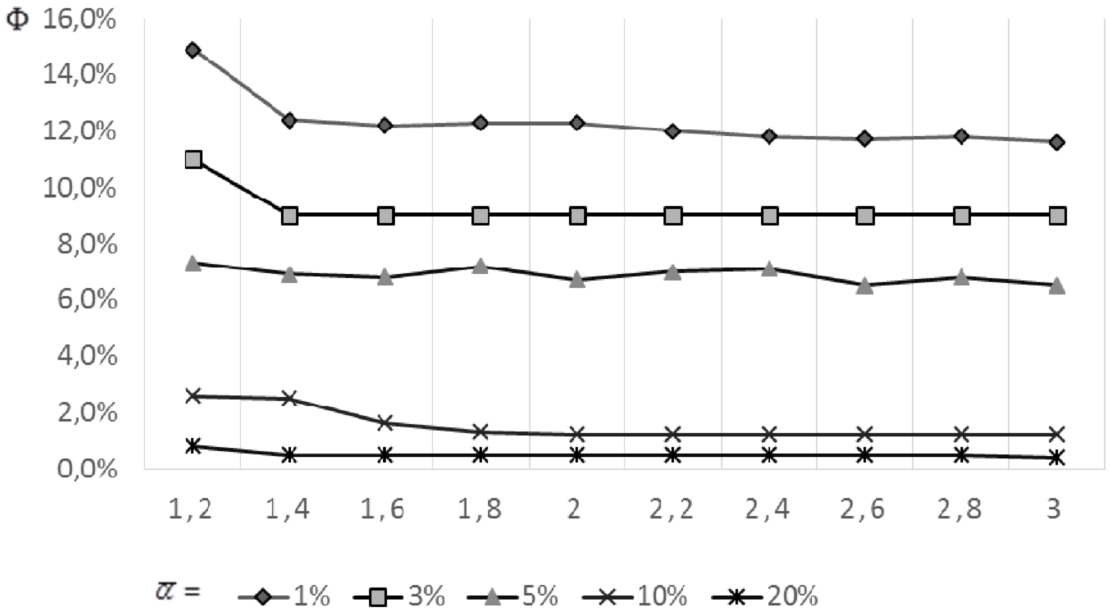

Fig. 1. The dependence of Φ on $\overline{T}_{\max}$ for various values of $\overline{\alpha}$

Rys. 1. Zależność Φ od $\overline{T}_{\max}$ dla różnych wartości $\overline{\alpha}$

We performed many experiments with different network topologies, different set of possible capacities and various values of workload. For a network consisting of 20 nodes and 34 channels (9 alternative capacities) the execution time of approximate algorithm is presented in Table 2.

Table 2

Executing time of approximate algorithm for a sample ISP network in seconds

$\overline{T}_{\max}$	Normalized $\overline{\alpha}$							
	1%	3%	5%	10%	20%	30%	40%	50%
1,2%	0,201	0,133	0,124	0,078	0,046	0,046	0,046	0,031
1,4%	0,202	0,140	0,125	0,078	0,046	0,062	0,031	0,046
1,6%	0,202	0,156	0,128	0,080	0,046	0,046	0,031	0,047
1,8%	0,206	0,158	0,130	0,084	0,046	0,031	0,031	0,046
2,0%	0,207	0,160	0,131	0,085	0,046	0,046	0,046	0,047
2,2%	0,210	0,164	0,135	0,088	0,062	0,062	0,046	0,051
2,4%	0,210	0,165	0,139	0,090	0,078	0,062	0,078	0,051
2,6%	0,215	0,167	0,140	0,092	0,124	0,109	0,109	0,051
2,8%	0,216	0,169	0,144	0,095	0,124	0,140	0,140	0,051
3,0%	0,218	0,171	0,145	0,099	0,420	0,156	0,156	0,051

The above results are for a sample ISP topology corresponding to a polish network backbone of fiber interconnection infrastructure. We notice that an approximate algorithm is especially effective regarding the running time for normalized $\overline{\alpha}$ greater than 5%. For this range executing time of algorithm is lower than 100ms. Otherwise, for the rest of the cases executing time is quite acceptable from a practical point of view.

6. Conclusion

Capacity and Flow Assignment Problem is of great importance in contemporary computer networks. Recently emerging technologies and concepts like Software Defined Networks with its centralization and distinction between data and control planes provide a single place to provision bandwidth across the network. The complexity of an exact algorithm for solving CFA makes it infeasible to use, for that reason a heuristic algorithm was proposed in this paper.

The proposed algorithm itself can be used for any application where a CFA problem arises, for example its variation can be used for resource allocation problems. But only as long as a centralized point of execution is available, it is not known whether a distributed version of the heuristic approach is possible.

The experiments involved using an off-the-shelf desktop computer and a real-life network topology. The amount of time required by the heuristic algorithm for the analysed network topology was below 0.3 second. Such running time makes it a viable solution for an SDN controller used in an ISP network to provision bandwidth across a network with regard to link preference.

BIBLIOGRAPHY

1. Agarwal S., Kodialam M., Lakshman T. V.: Traffic engineering in software defined networks. In INFOCOM Proceedings IEEE, 2013, p. 2211÷2219.
2. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C.: Introduction to algorithms. MIT press 2001.
3. Ferreira R. P. M., Luna H. P. L., Mahey P., Souza M. C. D.: Global optimization of capacity expansion and flow assignment in multicommodity networks. Pesquisa Operacional, 33(2), 2013, p. 217÷234.
4. Fratta L., Gerla M., Kleinrock L.: Flow Deviation: 40 years of incremental flows for packets, waves, cars and tunnels. Computer Networks, 66, 2014, p. 18÷31.

5. Girgis M., Mahmoud T., El-Hameed H. A., El-Saghier Z.: Routing and capacity assignment problem in computer networks using genetic algorithm. Inf. Sci. Lett 2, 2013, p. 13÷25.
6. Gladysz J., Walkowiak K.: Combinatorial optimization of capacity and flow assignment problem for unicast and anycast connections with linear and convex objective functions – exact and heuristic algorithms. Electronic Notes in Discrete Mathematics, 2010, p. 1041÷1048.
7. Gola M., Kasprzak A: The two-criteria topological design problem in WAN with delay constraint, Computational Science and Its Applications – ICCSA 2003.
8. Gola M., Czubak A.: Approximate Algorithm for Fast Capacity Provisioning in WANs with Trade-Off between Performance and Cost under Budget Constraint. In Proceedings of the Ninth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX. June 30÷July 4, 2014, Springer International Publishing, Brunów, Poland, 2014, p. 211÷221.
9. Goscien R., Pozniak-Koszalka I., Koszalka L., Kasprzak A.: A Top-Down Heuristic for TCFA Problem in WAN. In ICWMC 2013, The Ninth International Conference on Wireless and Mobile Communications, 2013, p. 89÷94.
10. Gude N., Koponen T., Pettit J., Pfaff B., Casado M., McKeown N., Shenker S.: Nox: towards an operating system for networks. SIGCOMM Comput. Commun. Rev. 38, July 2008, p. 105÷110.
11. Koldehofe B., Dürr F., Tariq M. A., Rothermel K.: The power of software-defined networking: line-rate content-based routing using OpenFlow. In Proceedings of the 7th Workshop on Middleware for Next Generation Internet Computing, p. 3, ACM 2013.
12. Koponen T., Casado M., Gude N., Stribling J., Poutievski L., Zhu M., Ramanathan R., Iwata Y., Inoue H., Hama T., Shenker S.: Onix: a distributed control platform for large-scale production networks. In: Proceedings of the 9th USENIX conference on Operating systems design and implementation. OSDI'10, USENIX Association, Berkeley, CA, USA 2010, p. 1÷6.
13. Lin G., Huang C., Zhan S., Lu X., Lu Y.: Ranking Based Selection Genetic Algorithm for Capacity Flow Assignments. In Computational Intelligence and Intelligent Systems, Springer, Berlin Heidelberg 2010, p. 97÷107.
14. Pioro M., Medhi D.: Routing, flow, and capacity design in communication and computer networks. Morgan Kaufmann, 2004.
15. Walkowiak K.: A flow deviation algorithm for joint optimization of unicast and anycast flows in connection-oriented networks. In: Computational Science and its Applications–ICCSA 2008, Springer, 2008, p. 797÷807.

16. Van der Merwe J., Kalmanek C.: Network Programmability is the answer. In Workshop on Programmable Routers for the Extensible Services of Tomorrow (PRESTO 2007), Princeton, NJ, May 2007.
17. Zhan Y., Lu J., Li S.: A Hybrid GA-TS Algorithm for Optimizing Networked Manufacturing Resources Configuration. Appl. Math, 7(5), 2013, p. 2045÷2053.

Omówienie

Sieci SDN są technologią, która daje nowe możliwości w warstwie sterowania ruchem w sięciach komputerowych. Sieci dostawców Internetu (ISP) są jednym z możliwych zastosowań tej technologii, która zapewnia m.in. scentralizowany punkt zarządzania siecią. Również i w tych topologiach należy rozwiązać problem przydzielenia pasma dla danego ruchu sieciowego na wybranym łączu (CFA). Ze względu na złożoność obliczeniową dokładnego rozwiązania tego problemu w pracy proponujemy algorytm heurystyczny. Algorytm ten z jednej strony bierze pod uwagę osiąganą przepustowość przez minimalizowanie opóźnienia, z drugiej uwzględnia indywidualne preferencje dostawcy usług internetowych (ISP), który z pewnych przyczyn, często ekonomicznych, decyduje się przesłać ruch sieciowy innym niż najszybsze łącze. W tabeli 1 umieszczono stratę związaną z wykorzystaniem rozwiązania heurystycznego zamiast dokładnego, natomiast czas wykonania algorytmu został przedstawiony w tabeli 2.

Addresses

Mariusz GOLA: Opole University of Technology, Institute of Automatic Control and Informatics, ul. Sosnkowskiego 31,45-272 Opole, Poland, mariusz.gola@po.opole.pl

Adam CZUBAK: Opole University, Institute of Mathematics and Informatics, ul. Oleska 48, 45-052 Opole, Poland, adam.czubak@math.uni.opole.pl