Metaheuristic Approach for Survivable P2P Multicasting Flow Assignment in Dual Homing Networks

Wojciech Kmiecik and Krzysztof Walkowiak

Department of Systems and Computer Networks, Wroclaw University of Technology, Wroclaw, Poland
{wojciech.kmiecik,krzysztof.walkowiak}@pwr.wroc.pl

Abstract. Due to growing demand for high definition music and video content, Peer to Peer (P2P) multicasting providing live streaming services has been gaining popularity in the last years. In this paper, we focus on applying the P2P multicasting for delivering of critical data that require to be transmitted safely, intact and with as little delay as possible, e.g., financial data, software security patches, antivirus signature database updates etc. To improve survivability of the P2P multicasting, we propose to use dual homing approach, i.e., each peer is connected to the overlay by two separate access links. The optimization problem is formulated in the form of Integer Linear Programming (ILP). We introduce a Simulated Annealing (SA) algorithm for the considered optimization problem and compare it with optimal results provided by CPLEX solver. Our studies demonstrate that the SA method yields results close to optimal and provides better scalability comparing to CPLEX, since it can solve in reasonable time much larger problem instances than CPLEX.

Keywords: P2P multicasting, survivability, dual homing, protection, Simulated Annealing, overlay network.

1 Introduction

Nowadays, we are observing a rapid growth in popularity of multimedia streaming in the Internet. To emphasize the growing popularity of various video streaming services, we need to quote [3] where the authors claim that Video on Demand traffic will triple and Internet TV will be increased 17 times by 2015. The total share of all forms of video (already mentioned) and P2P will grow continuously to be approximately 90 percent of all global consumer traffic in the next three years. Services like IPTV, internet radio, Video on Demand and high definition video or audio streaming are very useful for network users, but often require a lot of bandwidth, which can be costly [1]. The main advantages of using the P2P (Peer-to-Peer) approach are scalability, adaptability, low deployment cost and optimal content distribution [2], which are crucial to meet that demand. An overlay P2P multicasting technology is based on a multicast delivery tree consisting of peers (end hosts). Content transmitted by the P2P multicasting can be either data files or streaming content with additional requirements like

Á. Herrero et al. (Eds.): Int. JointConf. CISIS'12-ICEUTE'12-SOCO'12, AISC 189, pp. 215–224.
springerlink.com © Springer-Verlag Berlin Heidelberg 2013

bit rate etc. [8]. In many related works, the authors assume that users of a multicast network can leave the system. We address the situation where the P2P multicast system is static (peers stay connected to the system for a long time), like in:

- Content delivery network (CDN) - e.g., Akamai Technologies,
- Set-top box (STB) technology used in IPTV,
- Critical information streaming - e.g., hurricane warnings.

To improve network survivability, we apply the P2P multicasting in a dual homing architecture. The dual homing approach assumes that all hosts (nodes) have two disjoint links (homes) to the network. Those links provide network protection because of redundancy. The main contribution of the paper consists of: (i) Heuristic Simulated Annealing algorithm developed for the optimization problem of survivable P2P multicasting systems using dual homing architecture. (ii) Numerical experiments based on Integer Linear Programming (ILP) model and heuristic approach showing comparative results of both methods and other characteristics of the proposed concept.

The rest of the paper is organized in the following way. Section 2 introduces the concept of survivable P2P multicasting based on the dual homing technology. In Section 3, we formulate an ILP model for survivable dual homing P2P multicast. Section 4 presents our Simulated Annealing algorithm. In Section 5, we present results of our experiments. Finally, the last section concludes this work.

2 Survivability for the P2P Multicasting

In our previous works, we proposed to apply disjoint P2P multicast trees streaming the same content [6], [9]. Peers affected by a failure of one of the trees can use another tree to receive the required data in case of a failure. This procedure guarantees very low restoration time. In this paper, we study the network survivability problem for the dual homing architecture. In Fig. 1, we present a simple example to illustrate our concept. There are two disjoint multicast trees A, B that connect 8 nodes - a, b, c, d, e, f, g. In the case of tree A, nodes a, d and f are uploading nodes, while remaining ones are leafs. We use the term *level* to describe the location of nodes in the multicast tree. For example, node a is on level 1 of tree B, nodes b and e are on level 2 of tree B and rest of the nodes are on level 3.

The P2P multicasting is done in the application layer, i.e., end hosts are connected using the overlay network. Connections between peers are established as unicast connections over the underlying physical layer. Each peer is connected to the overlay by an access link. We propose to utilize the dual homing approach to protect the system against a failure of the access link. The main idea is to create two P2P multicasting trees guaranteeing that each of access links carries traffic only of one of the trees. Since each node has two access links (dual homing), it receives the streaming data from both trees on two separate links. Thus, if one of access links is broken, the node is still connected to the stream and moreover, it can upload the stream to subsequent peers located in the tree. A proper configuration of the P2P multicasting with dual homing protects the network from two kinds of failures:

- Uploading node failure – a failure that impacts all successors of the failed peer in the tree,
- Overlay link failure – that one comprises a failure of both directed links between nodes.

3 ILP Model

What we describe below is an ILP model introduced in [6] that considers dual homing architecture. To formulate the problem we use the notation as in [7]. Let indices $v,w = 1,2,\ldots,V$ denote peers – nodes of the overlay network. There are K peers (clients) indexed $k = 1,2,\ldots,K$ that are not root nodes in any trees and want to receive the data stream. Index $t = 1,2,\ldots,T$ denotes streaming trees. We assume that $T = 2$, however the model is more general and values $T > 2$ may be used. In trees, nodes are located on levels $l = 1,2,\ldots,L$. That gives us possibility to set a limit on the maximum depth of the tree. The motivation behind this additional constraint is to improve the QoS (Quality of Service) parameters of the P2P multicasting, .e.g., network reliability and transmission delay. If node v is root of the tree t, then $r_{vt} = 1$, otherwise $r_{vt} = 0$. Constant c_{wv} denotes streaming cost on an overlay link (w,v).

To model survivable P2P multicasting, we modify formulations presented in previous papers [6], [9]. We introduce constant $\tau(v)$, which adds a virtual node associated with the node v. Together they form a primal node. Every primal node has in fact four capacity parameters – constants d_v and u_v are respectively download and upload capacity of the node v and constants $d_{\tau(v)}$ and $u_{\tau(v)}$ are parameters of the virtual node $\tau(v)$. The objective function is streaming cost of all multicast trees. This can be defined in many ways, e.g., as network delay or transmission cost.

Fig. 1 depicts an example of the dual homing modeling. Dual homes are marked with a pattern of sequential lines and dots.

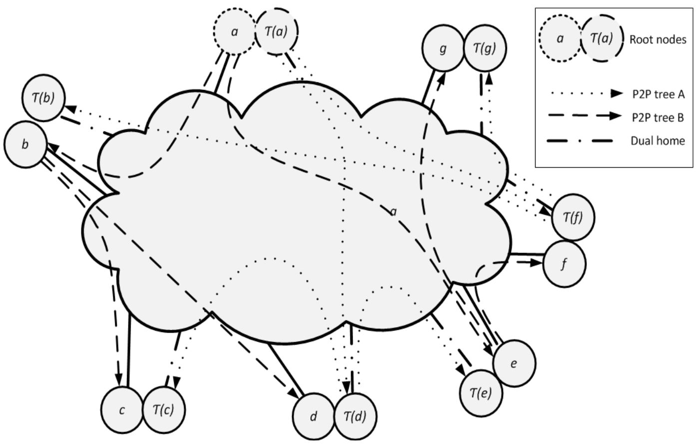

Fig. 1. Modeling dual homing

To model the P2P multicasting we use as in [6] binary variable x_{wvtk}, which is used to denote an individual flow for each streaming path from the root node to node k. This variable is set to 1, when streaming path to node k in tree t includes an overlay link between nodes w and v; 0 otherwise. Additional binary variable x_{wvt} is set to 1, if a link from node w to node v (no other peer nodes in between) is established in the multicast tree t; 0 otherwise. Auxiliary binary variable x_{vt} is set to 1, if an access link of node v is used to download or upload flow of the multicast tree t; 0, otherwise.

indices

$v,w,b = 1,2,\ldots,V$ overlay nodes (peers)
$k = 1,2,\ldots,K$ receiving nodes (peers)
$t = 1,2,\ldots,T$ streaming tree index

constants

d_v download capacity of node v (Kbps)
u_v upload capacity of node v (Kbps)
r_{vt} = 1, if node v is the root (streaming node) of tree t; 0, otherwise
q the streaming rate (Kbps)
c_{wv} streaming cost on overlay link from node w to node v
M large number
$\tau(v)$ index of node associated with node v (dual homing)
L The maximum number of levels of the tree

variables

x_{wvtk} = 1 if in multicast tree t the streaming path from the root to node k includes an overlay link from node w to node v (no other peer nodes in between); 0, otherwise (binary)
x_{wvt} = 1 if link from node w to node v (no other peer nodes in between) is in multicast tree t; 0, otherwise (binary)
x_{vt} = 1 if access link of node v is used to download or upload flow of multicast tree t; 0, otherwise (binary)

objective

$$\text{minimize} \quad \sum_w \sum_v \sum_t x_{wvt} c_{wv} \tag{1}$$

constraints

$$\sum_t x_{vvt} = 0 \quad v = 1,2,\ldots,V \tag{2}$$

$$\sum_w \sum_t x_{wvt} q \le d_v \quad v = 1,2,\ldots,V \tag{3}$$

$$\sum_v \sum_t x_{wvt} q \le u_w \quad w = 1,2,\ldots,V \tag{4}$$

$$\sum_w x_{wvt} + \sum_w x_{vwt} \le M x_{vt} \quad v = 1,2,\ldots,V \quad t = 1,2,\ldots,T \tag{5}$$

$$\sum_t x_{vt} = 1 \quad v = 1,2,\ldots,V \tag{6}$$

$$x_{vt} + x_{\tau(v)t} = 1 \quad v = 1,2,\ldots,V \quad t = 1,2,\ldots,T \tag{7}$$

$$\sum_t x_{v\tau(v)t} = 0 \quad v = 1,2,\ldots,V \tag{8}$$

$$\sum_w x_{wvkt} - \sum_w x_{vwkt} = x_{kt} \quad v = k \quad v = 1,2,\ldots,V \quad k = 1,2,\ldots,V \quad t = 1,2,\ldots,T \tag{9}$$

$$\sum_w x_{wvkt} - \sum_w x_{vwkt} = -x_{kt} \quad r_{vt} = 1 \quad v = 1,2,\ldots,V \quad k = 1,2,\ldots,V \quad t = 1,2,\ldots,T \tag{10}$$

$$\sum_w x_{wvkt} - \sum_w x_{vwkt} = 0 \quad v \neq k \quad r_{vt} \neq 1 \quad v = 1,2,\ldots,V \quad k = 1,2,\ldots,V \quad t = 1,2,\ldots,T \tag{11}$$

$$x_{wvkt} \leq x_{wvt} \quad v = 1,2,\ldots,V \quad w = 1,2,\ldots,V \quad k = 1,2,\ldots,V \quad t = 1,2,\ldots,T \tag{12}$$

$$\sum_w \sum_v x_{wvkt} \leq L \quad k = 1,2,\ldots,V \quad t = 1,2,\ldots,T \tag{13}$$

Condition (2) assures that the node internal flow is zero. This constraint guarantees that in each tree there is at most one transmission per overlay link (w,v). (3) and (4) are respectively the download and upload capacity constraints. Condition (5) specifies definition of the x_{vt} variable. The survivability constraint (6) assures, that multicast trees are separate – each node v (access link) can only be used in one tree t. Constraints (7), (8) state that only one node from the primal node can belong to tree t and there cannot be any connection within the primal node. Conditions (9)-(11) are called flow conservation constraints and define flows in P2P multicasting trees. Connection between (w,v) exists in tree t, when there is at least one transmission between nodes (w,v) in tree t to receiving node k (12). Finally, constraint (13) sets the levels upper limit in the path to each receiving node. For more details on modeling of survivable P2P multicasting, refer to [9].

The considered problem is complex and NP-hard, since it can be reduced to the hop-constrained minimum spanning tree problem, which is known to be NP-hard [4]. Due to that fact, branch-and-bound or branch-and-cut methods have to be used to obtain optimal solution.

4 Simulated Annealing (SA) Algorithm

Solving ILP models including a large number of variables and constraints can take a huge amount of processing time and computing power. In order to be able to solve such models in a relatively short period of time and obtain solution close to optimum, we developed a heuristic algorithm based on Simulated Annealing approach.

The starting solution of the algorithm is prepared by taking the following steps [6]:

1. Pick a node that is root for tree t.
2. Randomly pick any remaining node and connect it to the tree t by making it a child of the node connected before ; if node v is connected to tree t then node $\tau(v)$ has to be connected to the other tree.
3. If all nodes are connected – terminate.

This procedure guarantees that there are no loops in the network. At this point, the created structure does not have to meet all the constraints.

We use an insert method to create a random solution from the neighborhood of the current solution. We randomly pick node v with $r_{vt} = 0$ for $t = 1,2...T$ and choose its new parent w. All the children of node v become children of its former parent. If new parent w is in the same tree as node $\tau(v)$, then that node is reconnected to a different tree.

Original Simulated Annealing algorithm was created to solve unconstrained problem. In order to use it for our problem that includes numerous constraints, we propose a penalty method (14) similar to the Tabu Search algorithm described in [6]:

$$\mathrm{f}(cp,lp,up)=x_1cp+x_2cplp+x_3cpup \tag{14}$$

cp cost penalty, which is a difference between the cost of generated solution and the cost of current solution,

lp level penalty, which is set to 1 when there are more levels in the multicast trees than L value; 0 otherwise,

up upload penalty, which is a difference between how much the new parent w uploads and its upload capacity,

x_1,x_2,x_3 weights for each module of penalty method.

SA algorithm has following input parameters:

T_s starting temperature

T_e ending temperature

SA uses geometric progression of temperature. We developed additional optimization method, which is invoked after all iteration of the algorithm. The function finds the most expensive connections in the multicast trees and tries to find cheaper alternatives. The new solution is accepted if the obtained overall network cost is lower than the current one.

5 Research

5.1 Comparing ILP Model and SA Algorithm - Experiment Design

We randomly generated networks with 10 (small network) and 20 (big network) primal nodes and two disjoint trees. Our goal was to test how the level limit L and the streaming rate q would affect the overall network cost achieved by CPLEX and SA algorithm. To solve ILP model in an optimal way, we use newest CPLEX solver v12.4 [5]. We created 2 different networks with link costs in range 1-20, 1-50 and with either symmetric (100Mbps/100Mbps - 10% of all nodes) or asymmetric nodes (1Mbps/256Kbps, 2Mbps/512Kbps, 6Mbps/512Kbps, 10/1Mbps, 20/1Mbps, 50/2Mbps, 100/4Mbps). We assume that the first node is the streaming node for tree $t = 1$ and node $\tau(1)$ is the streaming node for tree $t = 2$. We set $time_{max}$ value to 3600 seconds, limiting the execution time of CPLEX. That gave us the possibility to compare ILP model and heuristic algorithm in terms of quality of obtained solution and time of processing. We tested how the SA algorithm would perform for different values of starting and ending temperature (from 1 to 1000 in both cases) and we found that the best parameters were:

- Starting temperature T_s: 500,
- Ending temperature T_e: 1.

We set those values for the experiments described below. In another preliminary experiment we discovered that the best values for the weights of penalty method were 1, 0.25 and 0.15 respectively. Overall, we conducted three experiments:

- Checking how the number of iterations and execution time would affect results obtained by the *Simulated Annealing* algorithm for either size of the network,
- Comparing CPLEX and SA algorithm for small size of the network and different streaming rate q values in terms of the overall network cost,
- Comparing CPLEX and SA algorithm for big size of the network and different number of levels L in terms of the overall network cost.

5.2 Comparing ILP Model and SA Algorithm – Results

The purpose of the first experiment was to check how the number of iterations would affect the results of the SA algorithm. In Fig. 2 and 3 we show results of SA for small and big size of the network respectively – we present both average overall network cost and best obtained network cost for 100 repetitions of algorithm execution. We can easily notice that, with an increase in the number of iterations, the SA algorithm performs better. Main drawback of increasing the number of iterations is growth of the execution time. Our conclusion is that for every experiment we have to choose the number of iterations that allows us to achieve a quality solution in a reasonable time (i.e., seconds instead of hours etc.).

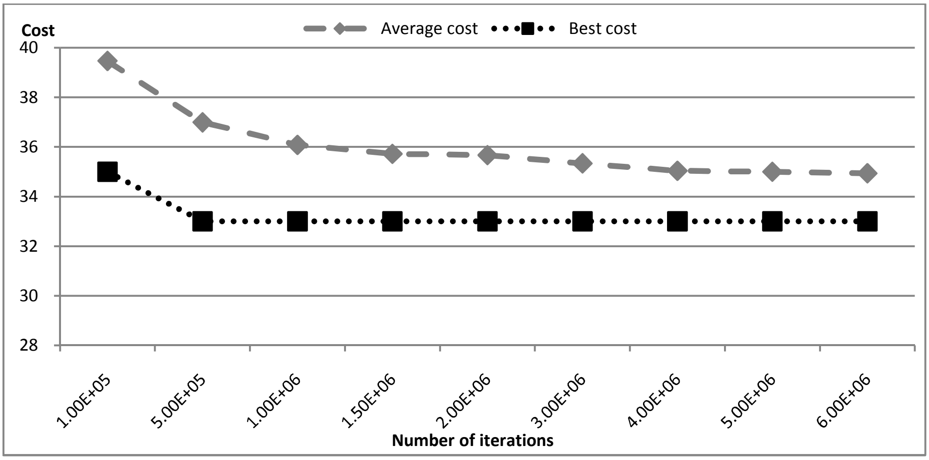

Fig. 2. Overall network cost as a function of the number of iterations for a small-sized network

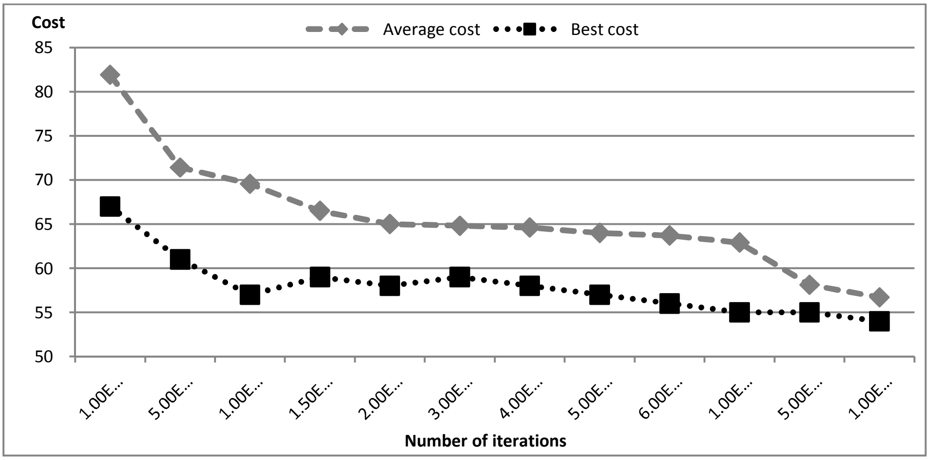

Fig. 3. Overall network cost as a function of the number of iterations for a big-sized network

The goal of the next experiment was to compare CPLEX and SA algorithm in terms of the overall network cost and time of processing for small size of network and different values of the streaming rate *q*. Tables 1 and 2 present obtained results for both CPLEX and SA and for different sets of cost range. Best and average results of SA were calculated based on 100 executions of the algorithm. For all types of networks, CPLEX achieved optimum in a short period of time (4 to 24 seconds). The SA algorithm was able to find optimum in 4 of 8 cases. Average overall network cost achieved by SA was 4-8% larger than CPLEX result, but processing time was only ~5 seconds. The general conclusion is that both algorithms proved to be good tools for finding solution for small size of the networks in terms of the overall network cost and processing time.

Table 1. Comparison of CPLEX and SA for small size of the network and cost range 1-20

cost 1-20							
	CPLEX		SA				
q [Kbps]	Cost	Time[s]	Best Cost	Diff. to opt.[%]	Avg Cost	Diff. to opt.[%]	Time[s]
64	33	18	33	0.0%	35.4	6.8%	4.7
128	33	16	35	5.7%	35.7	7.6%	4.8
192	36	24	37	2.7%	38.1	5.5%	4.8
256	36	22	37	2.7%	37.6	4.3%	4.7

Table 2. Comparison of CPLEX and SA for small size of the network and cost range 1-50

cost 1-50							
	CPLEX		SA				
q [Kbps]	Cost	Time[s]	Best Cost	Diff. to opt.[%]	Avg Cost	Diff. to opt.[%]	Time[s]
64	69	17	69	0.0%	73.0	5.5%	4.7
128	69	14	69	0.0%	73.3	5.8%	4.8
192	72	4	73	1.4%	78.2	7.9%	4.8
256	72	9	72	0.0%	77.4	7.0%	4.7

Tables 3 and 4 depict results of SA and CPLEX for big size of network (20 hosts). Optimal values were obtained by CPLEX without $time_{max}$ constraint that resulted in very long execution time (from 2 to almost 9 hours). In one hour time, CPLEX managed to find optimum in only one case of 14 and in 3 cases it was not able to find even a feasible solution. Most of the results of CPLEX with 1 hour time limit was 80-90% worse than optimum. The SA algorithm was able to find feasible solution every time and found optimum for 2 of 14 cases. The average cost obtained by SA was 3.5-7.5% worse than optimum. Another conclusion is that with a lower *L* limit the overall network cost is higher.

For larger networks CPLEX was unable to find a satisfying solution in a reasonable time (1 hour). SA achieved quality solutions in terms of overall network cost and proved itself useful for bigger types of network.

Table 3. Comparison of CPLEX and SA for big size of the network and cost range 1-20

cost 1-20										
	CPLEX		**CPLEX with time constraint**			**SA**				
LL	**Optimum**	**Time[s]**	**Best Cost**	**Diff. to opt.[%]**	**Time[s]**	**Best Cost**	**Diff. to opt.[%]**	**Avg Cost**	**Diff. to opt.[%]**	**Time[s]**
5	53	31525	Unknown	X	3600	55	3.6%	55.5	4.5%	3600
6	53	14423	Unknown	X	3600	54	1.9%	55.1	3.8%	3600
7	52	16563	791	93%	3600	54	3.7%	55.3	6.0%	3600
8	52	13963	430	88%	3600	54	3.7%	56.2	7.5%	3600
9	52	19231	520	90%	3600	54	3.7%	55.5	6.3%	3600
10	52	27082	349	85%	3600	53	1.9%	54.9	5.3%	3600
11	52	5411	52	0%	3560	54	3.7%	55.2	5.8%	3600

Table 4. Comparison of CPLEX and SA for big size of the network and cost range 1-50

cost 1-50										
	CPLEX		**CPLEX with time constraint**			**SA**				
L	**Optimum**	**Time[s]**	**Best Cost**	**Diff. to opt.[%]**	**Time[s]**	**Best Cost**	**Diff. to opt.[%]**	**Avg Cost**	**Diff. to opt.[%]**	**Time[s]**
5	93	10501	1655	94%	3600	94	1.1%	96.1	3.2%	3600
6	90	12323	Unknown	X	3600	91	1.1%	93.2	3.4%	3600
7	89	9482	465	81%	3600	90	1.1%	92.1	3.4%	3600
8	88	7942	89	1%	3600	90	2.2%	92.1	4.5%	3600
9	87	8892	93	6%	3600	90	3.3%	92	5.4%	3600
10	85	15076	354	76%	3600	90	5.6%	91.9	7.5%	3600
11	85	7383	549	85%	3600	88	3.4%	92	7.6%	3600

6 Conclusion

In this paper, we address the problem of survivable P2P multicasting in dual homing networks. Experiments testing the impact of streaming rate and number of levels on the overall network cost were conducted along with experiments focusing on comparison of the introduced SA algorithm and the ILP model. The results of experiments indicate that proper selection of parameters has big influence on the network cost. Both the streaming rate and the level limit impact the overall network cost. Moreover, the SA proved to be an useful algorithm for finding cost efficient solutions for all sizes of networks and achieved quality solutions close to optimum.

In future work, we plan to introduce new constraints that will provide more survivability, like node and ISP disjoint trees, and conduct more experiments evaluating these solutions.

Acknowledgements. This work is supported in part by the National Science Centre (NCN), Poland.

References

1. Aoyama, T.: A New Generation Network: Beyond the Internet and NGN. IEEE Comm. Magazine 47(5), 82–87 (2009)
2. Christakidis, A., Efthymiopoulos, N., Fiedler, J., Dempsey, S., Koutsopoulos, K., Denazis, S., Tombros, S., Garvey, S., Koufopavlou, O.: VITAL++, a new communication paradigm: embedding P2P technology in next generation networks. IEEE Comm. Magazine 49(1), 84–91 (2011)
3. Cisco Visual Networking Index Forecast 2010–2015 (2011), `http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf`
4. Dahl, G., Gouveia, L., Requejo, C.: On formulations and methods for the hop-constrained minimum spanning tree problem. In: Resende, M.G.C., Pardalos, P.M. (eds.) Handbook of Optimization in Telecommunications, pp. 493–515. Springer (2006)
5. ILOG CPLEX 12.4 User's Manual, USA (2009)
6. Kmiecik, W., Walkowiak, K.: Survivable P2P multicasting flow assignment in dual homing networks. In: 3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Budapest (2011)
7. Pióro, M., Medhi, D.: Routing, Flow, and Capacity Design in Communication and Computer Networks. Morgan Kaufman Publishers (July 2004) ISBN:978-0-12-557189-0
8. Shen, X., Yu, H., Buford, J., Akon, M.: Handbook of Peer-to-Peer Networking, 1st edn. Springer (2009)
9. Walkowiak, K., Przewoźniczek, M.: Modeling and optimization of survivable P2P multicasting. Computer Communications 34(12), 1410–1424 (2011)