Paper

Minimizing Cost of Network Upgrade for Overlay Multicast – Heuristic Approach

Maciej Szostak and Krzysztof Walkowiak

Department of Systems and Computer Networks, Wrocław University of Technology, Wrocław, Poland

Abstract—A rapid increase of the Internet users and traffic at the rate of 31% in years 2011–2016 contributes to emerging of new approaches to the content distribution. Among other approaches, the overlay multicasting seems to be one of the most interesting concepts according to relatively low deployment costs and large scalability. In this paper, the authors formulate a new incremental multicast overlay design problem. In particular, authors assumed that the overlay network is to be upgraded due to an increase of the number of participating users and the need to improve the streaming quality. However, the existing multicast tree structure is assumed to remain fixed. The goal was to minimize the cost of the upgrade, represented in euro/month. To achieve it, for each peer participating in the transmission, a link type offered by one of the ISPs was selected and overlay trees were constructed, rooted at the source of the content. The authors also present a new heuristic algorithm to efficiently solve this problem. According to experiments, the biggest factor influencing the upgrade cost and determining possible streaming quality values that the system can be upgraded to is the initial tree structure.

Keywords*—*multicasting, network design, optimization, overlay network, streaming.

1. Introduction

A very important aspect of any kind of design which should be taken into consideration at the time of planning is expansion. This especially applies to the network planning, due to the pace of user's number increase, Internet traffic (from 20000 PB per month in year 2011 to over 80000 PB per month in year 2016 [1]), as well as the growing number of applications and services with the high bandwidth demand. Newly created systems should incorporate such criteria as low cost of deployment, transport efficiency and fault tolerance. In this paper, we focus on the first two factors. We take into consideration real systems and business rights governing the market. This lead us to the development of a new network upgrade scenario – capacity increment with additional nodes and no changes to the existing tree structure.

In our work, we focus on one of the content delivery approaches – multimedia streaming – which has nowadays a significant role in the Internet. Not only isn't it flouting the artist's copyright but it also has a definite advantage over the Internet's major sharing mechanism, in which a user can access a file only once it has been fully downloaded. The overlay multicast meets all the requirements for such transmission without a violation of the physical core [2]. We assume that the overlay multicast is applied for a relatively static applications with a low membership change rate, e.g., videoconferencing, personal video broadcast in small groups, distance learning, collaborated workgroup, delivery of important messages (stocks, weather forecast, emergency alerts) [3]. The stream can increase or decrease a bit rate, depending on the network infrastructure capabilities. This method is called Adaptive Bit Rate Streaming, however it is designed to use unicast or anycast connections. The main reason for the network upgrade is the growing need for the bandwidth, e.g., users wanting higher quality of the video stream, which in turn means a higher bit rate. To answer this demand, the existing network must be incremented.

In this work authors continue research from [4], where three Integer Linear Problems (ILP) were formulated of join optimization of overlay multicast flows and link capacity with the objective to minimize the cost of the network upgrade. Main contributions of this paper are as follows:

- Formulation of the ILP for a new incremental multicast overlay design problem.
- Development of a new heuristic algorithm solving the proposed problem.
- Extensive experiments evaluating the performance of the proposed algorithm against optimal results and showing the behavior of the system as a function of various scenarios including number of trees, initial and final network size and QoS constraint.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 introduces the formulation of the incremental overlay multicast design problem. Section 4 contains a description of the heuristic algorithm. In Section 5 the results of experiments are presented and discussed. Section 6 concludes this work.

2. Related Works

There is an extensive literature about the topology design well covered in [5]. In addition, many surveys on the application layer multicasting have been carried out in [6], as there is a growing need for applications that will both stream real time content and retrieve on-demand content. However, most of the approaches focus on the optimal overlay multicast topology creation ([7]–[11]). Only few studies concern the incremental approach to the network design ([5], [12]–[16]). The main aim of those studies is to propose algorithms for network design problems considering number of different constraints and objectives. Both topology design and capacity increment coupled with a routing changes approaches are presented.

3. Mathematical Formulation

In this section, a mathematical formulation of the overlay network design problem for the overlay multicasting is presented. Overlay multicast networks are built on top of a general Internet unicast infrastructure rather than point-to-point links, therefore the problem of overlay network design is somewhat different than in networks that do have their own links [17]. The objective is twofold: to determine how much capacity is needed for each user participating in the transmission, and how to economically distribute the streaming content among the participants. The former goal comes to selection of access link types offered by Internet Service Providers, whereas the latter is to construct the overlay multicast trees. Assumptions for the model are taken from our previous works and real systems, therefore continuing the analysis from [18], an approach to consider a new scenario of the system capacity increment with additional nodes is extended. In this manner, the streaming rate of the system has to be incremented and additional nodes are to be added to the system. However, the structure of the existing trees cannot be modified and the link types of existing nodes cannot be worse than the initial ones. For the business reasons this comes as no surprise, because changing the link type to the lower capacity means a contract violation and can end up in additional fees.

Used model is an overlay tree distribution graph with one source of the content, in which we assume a division to multiple substreams of the main stream. Multiple delivery trees are created, each tree carrying a different substream. This approach prevents establishment of a leaf nodes among participating peers, which do not contribute to the overall distribution, and assumes that each peer receives substreams through the different routes. In presented approach, we require each node to be connected to all the trees, and streaming rate of each substream to be equal in amount. However, this scenario can be easily modified and consider a model with nodes receiving the streaming rate of a different quality, i.e., nodes are connected to the different subsets of substream trees and streaming rates of substreams varies. To formulate the problem notation proposed in [15] is used.

We apply a binary decision variable y_{vk} equal to 1, if node v is connected to overlay network by a link of type k and 0 otherwise. Each access link type offered by a given ISP has a particular download capacity (denoted as d_{vk}), upload capacity (denoted as u_{vk}) and cost (denoted by ξ_{vk}).

To construct multicast trees, the following types of decision variables are required: x_{wvt} equal to 1, if there is a link from node (peer) w to node v (no other nodes in between) in the multicast tree t, 0 otherwise. Also x_{wvet} equal to 1, if there is a path from the root node to node e, and it traverses through the link between nodes w and v in the tree t, 0 otherwise.

We also introduce continuous decision variable s_v representing monthly cost of network upgrade of node v. Participants apart from downloading the streaming content in the overlay trees, also take part in the other network services and therefore consume upload and download resources. For this reason, each node v is given a download and upload traffic ratio, denoted by the constants a_v and b_v respectively.

Capacity Increment Model with Additional Nodes (CIMAN)

Indices

v, w, e $= 1, 2, W, W+1, W+2, \ldots, V$ overlay nodes, where nodes $1, \ldots, W$ are the existing nodes, and $W+1, W+2, \ldots, V$ are additional nodes,

t $= 1, 2, \ldots, T$ multicast trees,

k, a $= 1, 2, \ldots, K_v$ access link types for node v.

Constants

a_v	download background transfer of node v (kbit/s),
b_v	upload background transfer of node v (kbit/s),
ξ_{vk}	cost of link of type k for node v (euro/month),
d_{vk}	download capacity of link of type k for node v (kbit/s),
u_{vk}	upload capacity of link of type k for node v (kbit/s),
r_v	= 1 if node v is the root of all trees, 0 otherwise,
q_t	streaming rate of the tree t (kbit/s),
t_{va}	= 1 if node v was connected to the overlay network by a link of type a, 0 otherwise,
z_{wvt}	= 1 if there was a link between node w and v in multicast tree t, 0 otherwise,
M	large number,
H	maximal number of hops from the root node to every additional node in the tree.

Variables

y_{vk}	= 1, if the node v is connected to the overlay network by a link of type k, 0 otherwise (binary),
x_{wvet}	= 1, if there is a path from the root node to node e, and it traverses through the link between nodes w and node v in the multicast tree t, 0 otherwise (binary),
x_{wvt}	= 1, if the link from node w to node v (no other nodes in between) is in the multicast tree t, 0 otherwise (binary),
s_v	cost of upgrading node v (continuous, euro/month).

Objective

The Objective (1) is to minimize the cost of upgrading the network.

$$\text{minimize } F = \sum_{v} s_v \,. \tag{1}$$

Subject to

Constraint (2) guarantees that for each tree $t = 1, 2, \ldots, T$ each additional node $v = W+1,\ W+2, \ldots, V$ must have exactly one parent node:

$$\sum_{w \neq v} x_{wvt} = 1 \quad v = W{+}1, W{+}2, \ldots, V \quad t = 1, 2, \ldots, T\,. \tag{2}$$

Condition (3) assures that there is a path from the root node to additional node e traversing through the link between nodes w and v only if this link exists:

$$x_{wvet} \leq x_{wvt}$$

$$w{=}1, 2, \ldots, V \quad v, e{=}W{+}1, W{+}2, \ldots, V \quad t{=}1, 2, \ldots, T. \tag{3}$$

Condition (4) represents the flow conservation constraint for the nodes being destination node.

$$\sum_{w \neq v} x_{wvet} - \sum_{w} x_{vwet} = 1$$

$$v = e \quad e = W{+}1, W{+}2, \ldots, V \quad t = 1, 2, \ldots, T. \tag{4}$$

Formula (5) is the flow conservation constraint for the nodes being traversing node.

$$\sum_{w \neq v} x_{wvet} - \sum_{w} x_{vwet} = 0$$

$$v \neq e \quad r_v = 0 \quad e = W{+}1, W{+}2, \ldots, V \quad t = 1, 2 \ldots, T. \tag{5}$$

Equation (6) represents the flow conservation constraint for the node being root node.

$$\sum_{w \neq v} x_{wvet} - \sum_{w} x_{vwet} = -1$$

$$r_v = 1 \quad e = W+1, W+2, \ldots, V \quad t = 1, 2, \ldots, T. \tag{6}$$

Condition (7) is in the model to assure that each node $v = 1, 2, \ldots, V$ can have only one access link type.

$$\sum_{k} y_{vk} = 1 \quad v = 1, 2, \ldots, V. \tag{7}$$

Formula (8) is a download capacity constraint and satisfies the requirement of the download capacity of existing nodes $v = 1, 2, \ldots, W$ being greater or equal to the background traffic of a node v and the sum of streaming rates of all the multicast trees the node is connected to.

$$a_v + \sum_{w \neq v} \sum_{t} z_{wvt} q_t \leq \sum_{k} y_{vk}\, d_{vk} \quad v = 1, 2, \ldots, W. \tag{8}$$

Condition (9) is a download capacity constraint and satisfies the requirement of the download capacity of additional nodes $v = W+1, W+2, \ldots, V$ being greater or equal to the background traffic of a node v and the sum of streaming rates of all the multicast trees the node is connected to.

$$a_v + \sum_{w \neq v} \sum_{t} x_{wvt} q_t \leq \sum_{k} y_{vk} d_{vk} \quad v = W{+}1, W{+}2, \ldots, V. \tag{9}$$

Analogously, condition (10) is the upload capacity constraint of existing nodes $w = 1, 2, \ldots, W$, and is equal to the summary upload transfer of w which follows from the number of children nodes, the streaming rate and the background traffic of the node w.

$$b_w + \sum_{v \neq w} \sum_{t} z_{wvt} q_t \ \leq \ \sum_{k} y_{wk} u_{wk} \quad w = 1, 2, \ldots, W. \tag{10}$$

Constraint (11) is the upload capacity constraint of additional nodes $w = W+1, W+2, \ldots, V$, and is equal to the summary upload transfer of w which follows from the number of children nodes, the streaming rate and the background traffic of the node w.

$$b_w + \sum_{v \neq w} \sum_{t} z_{wvt} q_t \leq \sum_{k} y_{wk} u_{wk} \quad w{=}W{+}1, W{+}2, \ldots, V. \tag{11}$$

Formula (12) guarantees that there is no downgrade of the link type for existing nodes.

$$\sum_{k} t_{vk}\, \xi_{vk} \ \leq \ \sum_{k} y_{vk}\, \xi_{vk} \quad v = 1, 2, \ldots, W. \tag{12}$$

We introduce to the model conditions (13) and (14) which represent the cost of upgrading the access link types in the case of existing nodes and cost of building the network for additional nodes, respectively.

$$\sum_{k} \sum_{a} y_{vk} t_{va} (\xi_{vk} - \xi_{va}) \ \leq \ s_v \quad v = 1, 2, \ldots, W. \tag{13}$$

$$\sum_{k} y_{vk} \xi_{vk} \ \leq \ s_v \quad v = W{+}1, W{+}2, \ldots, V. \tag{14}$$

Formula (15) is introduced to meet the QoS requirement of the total length of hops from the root node to every additional node e in the multicast tree t.

$$\sum_{w} \sum_{v \neq w} x_{wvet} \leq H$$

$$e = W{+}1, W{+}2, \ldots, V \quad t = 1, 2, \ldots, T. \tag{15}$$

4. Heuristic Algorithm

In this section a new heuristic algorithm for CIMAN given by Eqs. (1)–(15) is presented. To formulate the algorithm, several additional terms and operators are introduced. All functions presented below are executed using the current state of the problem, i.e., the current values of decision variables, which in effect yield current network flows and access links' capacity. To formulate the algorithm the following definitions are introduced.

Let x_{wvtl} be equal to 1, if in the multicast tree t there is a link from the node w to the node v, and w is located on the level l of the multicast tree t, 0 otherwise.

Transfer between any node w and additional node v is *possible* in the tree t on the level l, if node w is located in the tree t on the level l; the node v is not yet connected to the tree t, and node w has sufficient residual upload capacity to stream the rate of the tree t.

Tree t is *feasible* on the level l, if there's at least one possible transfer from any node w located on the level l, to one of the additional nodes v.

Function $ftree(l)$ returns an index of a feasible tree on the level l. If there is more than one feasible tree, the tree with the lowest number of nodes connected to it is selected.

Let $isfeasible(v,t,l)$ return 1 if the node v is a *feasible parent* node on the level l of the tree t, which means, if at least one transfer in the tree t between the node v on the level l and any other additional node is possible.

Function $fpnode(t,l)$ returns an index of a feasible parent node located on the level l of the tree t. If there's more than one feasible parent node, the node with the largest value of residual upload capacity is selected. Notice that if $l = 1$, $fpnode(t,l)$ always returns an index of the root node.

Let $fcnode(v,t,l)$ return an index of a feasible child node of the node v located on the level l of the tree t. If there is more than one feasible child node, again the additional criterion is the residual upload capacity.

Function $istransfer(l)$ returns 1 if there is at least one possible transfer on the level l of any tree. Otherwise it returns 0.

Let $istree()$ return 1 if each additional node $v = W+1, W+2,\dots,V$ is connected to each tree $t = 1,2,\dots,T$ (all required transfers are completed), 0 otherwise.

Function $isupdate()$ returns 1 if incrementing the upload capacity of the access link is possible for at least one node. Otherwise it returns 0.

Let $istreetransfer()$ return 1 if there is a node v with sufficient upload capacity to provide at least one transfer in any tree, 0 otherwise.

Function $updatenode()$ returns an index of a node v, for which the upload capacity can be augmented. If there is more than one such a node, an additional criterion is applied, i.e., in the algorithm, several combinations of three values are used: the access link price, the node level and the relative cost of the upload capacity increase given by the formula $(u_{v(k+1)} - u_{vk})\,/\,(\xi_{v(k+1)} - \xi_{vk})$.

Set E denotes nodes updated after every iteration of the main loop of the algorithm.

4.1. Minimizing Cost of Upgrading the Network Heuristic Algorithm

Step 0. Load the existing tree structure and set $x_{wvtl} = 1$ for such nodes w, v, tree t and level l, that there is a link from existing node w to existing node v, and w is located on the level l of the multicast tree t.

Step 1. Create table E.

Step 2. Set $x_{wvtl} = 0$ for each $v = W+1,\ W+2,\dots,V$, $w = 1,2,\dots,V$, $t = 1,2,\dots,T$, $l = 1,2,\dots,L$. Set y_{vk} as the minimal values that guarantee sufficient download capacity for each node $v{=}W, W{+}1,\dots,V$ (i.e., $d_{vk} \geq a_v{+}Tq_t$) except for the root node as well as nodes from the table E, and the sufficient upload capacity for the root node ($r_v = 1$), nodes from table E and existing nodes $v = 1,2,\dots,W$ (i.e., $u_{vj} \geq b_v{+}Tq_t$).

Step 3. Set $l = 1$.

(a) Let $t{=}ftree(l)$. If $isfeasible(r,t,l){=}0$ set $l = l{+}1$ and go to Step 4. Otherwise, go to Step 3b.

(b) Calculate $w = fcnode(r,t,l)$ and set $x_{rwtl} = 1$. Go to Step 3a.

Step 4. If $istreetransfer(){=}0$ and $istree(){=}0$ go to Step 5. If $istree(){=}1$, go to Step 7, otherwise:

(a) If $istransfer(l) = 0$ set $l = l+1$ and go to Step 4. Otherwise go to Step 4b.

(b) Set $t = ftree(l)$, $w = fpnode(t,l)$, $v = fcnode(w,t,l)$ and $x_{wvtl} = 1$. Go to Step 4a.

Step 5. If $isupdate() = 1$, go to Step 6. Otherwise stop the algorithm, there is no feasible solution.

Step 6. Set $e = updatenode()$. Find k, for which $y_{ek} = 1$. Set $y_{ek} = 0$, $k = k+1$, $y_{ek} = 1$, $l = 1$, update table E, and go to Step 2.

Step 7. Find all nodes $v = 1,2,\dots,W,W{+}1,\dots,V$ with the unused upload capacity and decrease it if possible. Set $y_{vk} = 0$, $k = k-1$, $y_{vk} = 1$. Go to Step 8.

Step 8. Calculate the cost of upgrading the network denoted as C, as the sum of link type upgrade cost for existing nodes $v = 1,2,\dots,W$ ($y_{vk}\xi_{vk} - t_{va}\xi_{va}$), and used access link type prices for each additional node $v = W{+}1, W{+}2,\dots,V$ ($y_{vk}\xi_{vk}$). Go to Step 9.

Step 9. Stop the algorithm. The cost of network upgrade is equal to C.

The main idea of the Minimizing Cost of Upgrading the Network Heuristic Algorithm is as follows. The algorithm starts with loading the existing network structure and setting initial connections between existing nodes. Variable x_{wvtl} is set to 1 for such nodes w, v, tree t and level l, that there is a link from the existing node w to the existing node v, and w is located on the level l of the multicast tree t (Step 0). Then, we move on to the creation of a table to store updated nodes' indices (Step 1), which is updated after every access link type increase (Step 6).

In Step 2, an initialization of all of the remaining variables x_{wvtl} and variables y_{vk} is proceeded. The idea behind the selection of the latter is to find for each node a link that has a sufficient download capacity to transmit the background traffic and the overall streaming rate of multicast trees. For the root node, existing nodes and updated nodes, an additional procedure is run to ensure the satisfactory upload capacity to fulfill the transmission. Next, in Step 3, we check if there are any possible connections from the source of the content to additional nodes $v = W{+}1, W{+}2,\dots,V$ in each tree $t = 1, 2, \ldots, T$. If the root node has enough residual upload capacity the connection is established. Step 4 creates multicast trees denoted by variables x_{wvtl}. The main loop of Step 4 is repeated for the subsequent tree levels to allocate the resources of the upload capacity proportionally to all of the trees. If after Step 4, all nodes are connected to each tree, the algorithm tries to decrease the access links of every node (Step 7), and calculates the cost of upgrading the network (Step 8). Otherwise, there is an attempt to increase the capacity of the access link of the selected node (Step 6) and the network is rebuilt. Once all of the additional nodes $v = W+1, W+2, \ldots, V$ are connected, the algorithm stops.

5. Results

We implemented the presented heuristic algorithm in C++. The goal of numerical experiments was to examine the performance of presented approach against the traditional approach, and the heuristic approach against the optimal results. In all the experiments, we use DSL price lists of three ISPs operating in Poland (TP, Dialog and UPC) with prices in euro/month. To each node we randomly assign one of the ISPs, so that access link can be chosen from the pool of available options. The values of the download background traffic were selected at random in the range from 1024 to 2048 kbit/s. Analogously, the values of the upload background traffic were selected at random in the range from 256 to 512 kbit/s.

In order to obtain optimal results we solved the CIMAN using CPLEX 12.5 solver [19]. Due to the complexity of the problem, we decided to test the networks consisting of up to 25 initial and up to 40 final overlay nodes (peers), in order to obtain close to optimal results in a reasonable time. The streaming rate in the examined system was divided proportionally to 1–3 substreams. We introduced additional constraint following from the real systems, which assumed limitation of the number of hops from the source of the content to any additional node, in the range of 2–7, and is Quality of Service type of constraint. Tests were run for a fixed root node location and selection of ISP. Number of the final nodes was set to 15–40, depending on the size of initial system. Initial networks consisted of 15–25 nodes, 1–3 trees. For our investigation, we considered four different streaming bit rates corresponding to four different qualities of the video stream: 320p, 480p, 576p and 720p (HD). To compute the streaming rate to be distributed we used On2 VP6 video codec, NTSC frame rate, average motion, 16:9 aspect ratio, mp3 audio codec, stereo channels, medium audio quality and 44.1 kHz sampling rate. Due to the limitations of maximal upload capacities available from ISPs, we couldn't achieve Full HD quality stream using our approach. Note, that since the structure of the initial trees couldn't be modified, for some scenarios where the low quality stream (320p) was to be upgraded to the high quality stream (576p or 720p), the transmission was impossible. In total, 972 different scenarios were considered. We introduced a computation time limit of one hour for CPLEX solver, therefore in some cases no solution or only a feasible solution instead of the optimal one was found.

Table 1 shows the comparison of the CPLEX results and ones delivered by the CIMAN heuristic algorithm for the scenario with 10 initial nodes and 576p initial streaming quality. Due to the fixed structure of initial trees, achieving HD streaming quality was impossible for this scenario. Column 1 represents the number of multicast trees (T), column 2 is the number of final nodes (V), column 3 is a hop limit constraint and describes the maximal number of hops from the source of the content to the additional node (H), column 4 is the end quality of the stream (EQ). Columns 5–6 are related to the increment cost in euro/month for optimal (OPT) and heuristic ($HEUR$) approach, respectively. Column 7 is related to comparison of those two approaches (GAP), whereas columns 8–9 give computation time of CPLEX solver and the heuristic algorithm, respectively.

Table 1
Heuristic algorithm versus CPLEX results for initial streaming quality of 576p and 10 initial nodes

T	*V*	*H*	*EQ*	*CPLEX*	*HEUR*	*GAP*	*CPLEX*	*HEUR*
				[euro/month]		[%]	Time [s]	
1	15	2	567p	73	73	0.00	0.03	0
1	15	3	576p	63	63	0.00	0.21	0
1	15	4	576p	60	60	0.00	0.14	0
1	15	5	576p	60	60	0.00	0.09	0
1	15	6	576p	60	60	0.00	0.14	0
1	15	7	576p	60	60	0.00	0.14	0
2	40	3	567p	367	382	–4.09	1635	0.03
2	40	4	576p	*INF*	382	*INF*	3600	0.05
2	40	5	576p	*INF*	367	*INF*	3600	0.01
2	40	6	576p	*INF*	367	*INF*	3600	0.01
2	40	7	576p	498	367	26.31	3600	0.03
3	35	3	567p	*INF*	307	*INF*	3600	0.03
3	35	4	576p	*INF*	302	*INF*	3600	0.01
3	35	5	576p	*INF*	302	*INF*	3600	0.02
3	35	6	576p	297	302	–1.68	3328	0.01
3	35	7	576p	370	302	18.38	3600	0.02

On average, for experiments that feasible solution was delivered by CPLEX, the proposed heuristic approach delivers solutions 0.9% worse than optimal ones. For networks consisting of one tree, two trees and three trees, CIMAN heuristic algorithm delivers solutions 1.3%, 0.6% and 0.8% worse, respectively. In 582 cases, the heuristic approach yields the results equal to the ones delivered by CPLEX, and in 79 scenarios CPLEX doesn't deliver any solution after one hour of computation. In 44 cases. the proposed algorithm outperforms feasible results obtained by CPLEX solver (with 3600 seconds execution time limit), also for 27 scenarios CPLEX yields "out of memory" error. We can easily notice that the heuristic approach significantly outperforms CPLEX when it comes to the computation time. When the complexity of the problem increases (more nodes and trees), the heuristic approach is even 30000 times faster. Also, the computation of the proposed algorithm is finished within the fraction of a second for most of the experiments. Tests with more complex networks lead to expanding computation time, which was still relatively short.

Table 2
Upgrade cost for CIMAN

T	*W*	*V*	*H*	*IQ*	*IP* [euro/month]	Upgrade cost [euro/month] 360p	480p	576p
1	10	35	2	360p	110	275	–	–
1	10	35	3	360p	110	267	294	–
1	10	35	4	360p	110	265	287	–
1	10	35	5	360p	110	265	287	–
1	10	35	6	360p	110	265	287	–
1	15	30	3	480p	170	n/a	178	220
1	15	30	4	480p	170	n/a	178	211
1	15	30	5	480p	170	n/a	174	204
1	15	30	6	480p	170	n/a	171	204
1	15	30	7	480p	170	n/a	166	204
2	10	40	3	360p	109	320	374	–
2	10	40	4	360p	109	314	372	–
2	10	40	5	360p	109	309	372	–
2	10	40	6	360p	109	307	372	–
2	10	40	7	360p	109	307	372	–
2	10	40	8	360p	109	307	372	–
2	25	40	2	480p	282	n/a	178	–
2	25	40	3	480p	282	n/a	178	208
2	25	40	4	480p	282	n/a	173	208
2	25	40	5	480p	282	n/a	166	208
2	25	40	6	480p	282	n/a	166	208
3	20	40	2	360p	213	215	299	–
3	20	40	3	360p	213	209	298	–
3	20	40	4	360p	213	203	297	–
3	20	40	5	360p	213	203	296	–
3	20	40	6	360p	213	203	296	–
3	25	35	2	480p	280	n/a	118	181
3	25	35	3	480p	280	n/a	118	171
3	25	35	4	480p	280	n/a	115	168
3	25	35	5	480p	280	n/a	112	168
3	25	35	6	480p	280	n/a	112	168

Table 2 presents the upgrade cost in euro/month delivered by CPLEX solver, for different incremental scenarios. Column 1 represents the number of trees, columns 2–3 give the number of initial (*W*) and final nodes, column 4 is the hop limit constraint, column 5 gives the initial streaming quality (*IQ*). Column 6 shows the initial price of building the network in euro/month (*IP*), whereas columns 7–9 give the upgrade cost in euro/month to stream the quality of 360p, 480p and 576p, respectively. We can see, that using fixed initial tree structures limits the end quality that the system can be upgraded to. For the networks with the initial streaming quality of 360p, upgrades to 576p streaming quality are impossible. Moreover, upgrade to HD streaming quality (720p) cannot be achieved for any scenario. Also, the number of maximal hops between the source node and any additional node is a factor. Increasing it decreases the cost of the upgrade, plus when limited to 2, for some of the scenarios the transmission is impossible.

The second goal of experiments was to test the behavior of the systems with a larger number of participating nodes. Using the proposed heuristic we examined networks consisting of up to 200 initial nodes and 250 final nodes. Note that CPLEX solver cannot provide feasible results for such large networks, therefore to generate the initial network structures we used our different heuristic algorithm.

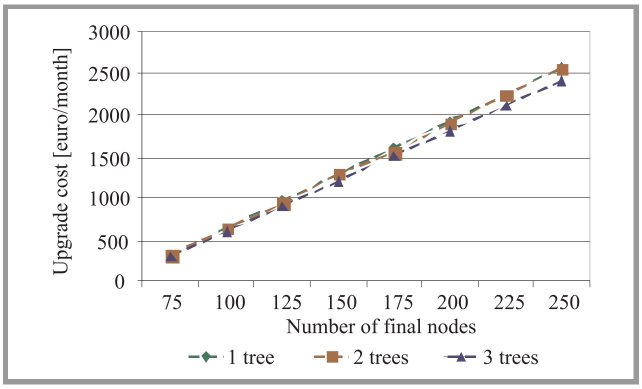

Fig. 1. Upgrade cost as a function of number of trees and number of final nodes (50 initial nodes, 576p stream quality).

Figure 1 shows the impact of introducing more trees to the system for the network consisting of 50 initial nodes, 576p streaming quality and the final network size of 75–250 nodes. Systems with three multicast trees show greater difference in price range, which comes up to over 150 euro/month, whereas the cost of upgrading the systems with one or two multicast trees is comparable.

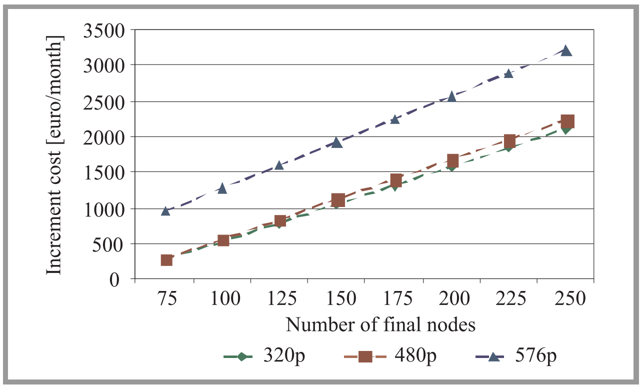

Fig. 2. Upgrade cost as a function of number of final nodes and end stream quality (50 initial nodes, 1 tree, 320p initial stream quality).

Figure 2 depicts the upgrade cost for the initial network consisting of 50 nodes, one tree and 320p streaming qual-

Table 3
Traditional versus incremental approach for initial stream quality of 360p

T	*W*	*V*	*EQ*	*TA*	*UC*	*IA*	*GAP*
					[euro/month]		[%]
1	10	15	360p	170	54	170	0.00
1	10	20	360p	229	113	229	0.00
1	10	25	360p	282	166	282	0.00
1	10	15	480p	183	70	186	–1.64
1	10	20	480p	251	138	254	–1.20
1	10	25	480p	311	198	314	–0.96
2	10	15	360p	159	50	159	0.00
2	10	20	360p	213	104	213	0.00
2	10	25	360p	262	153	262	0.00
2	10	15	480p	169	92	201	–18.93
2	10	20	480p	229	151	260	–13.54
2	10	25	480p	282	206	315	–11.70
3	10	15	360p	159	50	159	0.00
3	10	20	360p	213	104	213	0.00
3	10	15	480p	169	86	201	–18.93
3	10	20	480p	226	144	259	–14.60

Table 4
Traditional versus incremental approach for initial stream quality of 360p

T	*W*	*V*	*EQ*	*TA*	*UC*	*IA*	*GAP*
					[euro/month]		[%]
1	50	75	360p	796	263	796	0.00
1	50	100	360p	1058	526	1059	–0.09
1	50	125	360p	1322	789	1322	0.00
1	50	150	360p	1585	1052	1585	0.00
1	50	175	360p	1850	1315	1848	0.11
1	50	200	360p	2113	1578	2111	0.09
1	50	225	360p	2376	1841	2374	0.08
1	50	250	360p	2639	2106	2639	0.00
1	150	175	360p	1850	263	1848	0.11
1	150	200	360p	2113	526	2111	0.09
1	150	225	360p	2376	789	2374	0.08
1	150	250	360p	2639	1054	2639	0.00
2	100	125	360p	1294	258	1294	0.00
2	100	150	360p	1552	515	1551	0.06
2	100	175	360p	1809	773	1809	0.00
2	100	200	360p	2067	1030	2066	0.05
2	100	225	360p	2325	1288	2324	0.04
2	100	250	360p	2582	1546	2582	0.00
2	100	125	480p	1402	703	1823	–30.03
2	100	150	480p	1679	956	2076	–23.65
2	100	175	480p	1956	1227	2347	–19.99
2	100	200	480p	2238	1506	2626	–17.34
2	100	225	480p	2515	1783	2903	–15.43
2	100	250	480p	2797	2062	3182	–13.76
3	150	175	360p	1938	275	1936	0.10
3	150	200	360p	2213	552	2213	0.00
3	150	225	360p	2490	827	2488	0.08
3	150	250	360p	2765	1104	2765	0.00
3	150	175	480p	2121	665	2484	–17.11
3	150	200	480p	2423	918	2737	–12.96
3	150	225	480p	2725	1199	3018	–10.75
3	150	250	480p	3027	1488	3307	–9.25

ity, to networks of up to 250 nodes and three different streaming qualities: 320p, 480p and 576p. There is slight price difference when upgrading from 320p streaming quality to 480p, however obtaining mid quality stream (576p) from low (320p) initial stream is about twice as expensive. Due to the fixed initial tree structure, HD streaming quality cannot be delivered.

Comparison of the traditional approach versus the incremental approach is presented in Table 3 and Table 4 for small and large networks, respectively. Column 1 represents the number of trees, columns 2–3 give the number of initial and final nodes, column 4 is the end quality stream. Column 5 (*TA*) is the cost of building the network using the traditional approach in euro/month, whereas columns 6–7 give the upgrade cost (*UC*) and the total price (*IA*) in euro/month of building the network using the incremental approach, respectively. Column 7 is related to the comparison of those two approaches.

The results show for both small and large network sizes, that upgrading the network in the sense of introducing to the system more participating peers without the change of the streaming quality, brings almost the same price as using the traditional approach (building the network from the scratch). For larger networks (Table 4), where CPLEX couldn't deliver optimal results and the heuristic algorithm to generate the input data was used, for some of the scenarios the incremental approach is slightly better. This is caused by the fact, that the heuristic approach brings close to optimal results, and there is always room for the improvement. The second trend is seen when introducing more nodes to the system, combined with the streaming quality increase. This contributes to traditional approach outperforming the incremental one by up to 30%. This is again caused by the fixed initial multicast tree structure creating bottlenecks for faster transmission.

6. Conclusion

This paper addressed the problem of Capacity Increment Approach with Additional Nodes and no existing tree modifications. The objective of the optimization was to minimize the cost of upgrading the system. Authors proposed a new heuristic algorithm and illustrated this approach by showing the results using both CPLEX solver and newly proposed heuristic. In numerical experiments different incremental scenarios were considered. Results delivered by proposed algorithm were comparable to the solutions yielded by CPLEX. Moreover, for networks consisting of the larger number of nodes, CPLEX solver couldn't provide feasible solutions in one hour time limit, and either couldn't find any solution or yielded out-of-memory exception. According to the obtained results, we can conclude, that the biggest factor influencing the upgrade cost is the initial tree structure, which is the bottleneck for the bigger throughput and prevents the streaming quality upgrade. Other parameters, like the maximal number of hops from the source of the content to any of the newly connected nodes, have smaller influence on the upgrade cost.

References

[1] "Cisco visual networking index: Forecast and methodology, 2011–2016", Cisco, 2012.

[2] S. Banerjee *et al.*, "Omni: An efficient overlay multicast infrastructure for real-time applications", *Comp. Netwo.: The Int. J. Comp. Telecommun. Netw.*, vol. 50, iss. 6, pp. 826–841, 2006.

[3] J. Bufford, H. Yu, and E. K. Lua, *P2P Networking and Applications*. Morgan Kaufmann Series in Networking, D. Clark, Ed. Morgan Kaufmann, 2008.

[4] M. Szostak and K. Walkowiak, "Incremental approach to Overlay Network Design Problem", in *Proc. 17th Polish Teletraf. Symp. 2012*, Zakopane, Poland, 2012.

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

[6] T. Small, B. Li, and B. Liang, "Outreach: Peer-to-peer topology construction towards minimized server bandwidth costs", *IEEE J. Selec. Areas Commun.*, vol. 25, no. 1, pp. 35–45, 2007.

[7] C. Wu and B. Li, "On meeting P2P streaming bandwidth demand with limited supplies", in *Proc. 15th Ann. SPIE/ACM Int. Conf. Multime. Comput. Netwo. MMCN 2008*, San Jose, CA, USA, 2008.

[8] B. Akbari, H. Rabiee, and M. Ghanabari, "An optimal discrete rate allocation for overlay video multicasting", *Comp. Commun.*, vol. 31, pp. 551–562, 2008.

[9] Y. Cui, Y. Xue, and K. Nahrstedt, "Optimal resource allocation in overlay multicast", *IEEE Trans. Parall. Distrib. Sys.*, vol. 17, no. 8, pp. 808–823, 2006.

[10] M. Kwon and S. Fhamy, "Path-aware overlay multicast", *Comp. Netw.*, vol. 47, no. 1, pp. 23–45, 2005.

[11] Y. Zhu and B. Li, "Overlay networks with linear capacity constraints", *IEEE Trans. Parall. Distrib. Sys.*, vol. 19, no. 2, pp. 159–173, 2008.

[12] N. Geary, A. Antonopoulos, E. Drakopoulos, and J. O'Reilly, "Analysis of optimization issues in multi-period DWDM network planning", in *Proc. 20th Ann. Joint Conf. IEEE Comp. Commun. Soc. IEEE INFOCOM 2001*, Anchorage, Alaska, USA, 2001, vol. 1, pp. 152–158.

[13] S. Gopal and K. Jain, "On network augumentation", *IEEE Trans. Reliabil.*, vol. 35, no. 5, pp. 541–543, 1986.

[14] H. Lee and D. Dooly, "Heuristic algorithms for the fiber optic network expansion problem", *Telecommun. Sys.*, vol. 7, pp. 355–378, 1997.

[15] A. Meyerson, K. Munagala, and S. Plotkin, "Designing networks incrementally", in Proc. *42nd Ann. Symp. Foundations Comp. Sci. FOCS 2001*, Las Vegas, Nevada, USA, 2001.

[16] A. Tero *et al.*, "Rules for biologically inspired adaptive network design", *Science*, vol. 327, pp. 439–442, 2010.

[17] S. Shi and J. Turner, "Multicast routing and bandwidth dimensioning in overlay networks", *IEEE J. Selec. Areas Commun.*, vol. 45, no. 8, pp. 1444–1455, 2002.

[18] M. Szostak and K. Walkowiak, "Two approaches to network design problem for overlay multicast with limited tree delay – model and optimal results", *Int. J. Electron. Telecommun.*, vol. 57, no. 3, pp. 335–340, 2011.

[19] "Ilog cplex version 12.5 user's manual", IBM Software.

Maciej Szostak received his M.Sc. degree from the Wrocław University of Technology in 2008. Currently, he is a Ph.D. student at the Department of System and Computers Networks, Faculty of Electronics, Wrocław University of Technology. His research interests include overlay networks, peer-to-peer computing, multimedia service delivery and virtualization. He is a member of reviewer committees of three international conferences on simulations and computer systems and reviewer in two international journals.

E-mail: maciej.szostak@pwr.wroc.pl
Department of Systems and Computer Networks
Wrocław University of Technology
Wybrzeże Wyspiańskiego st 27
50-370 Wrocław, Poland

Krzysztof Walkowiak – for biography, see this issue, p. 64.