# Network design problem for P2P multicasting

**Krzysztof Walkowiak***

* *Chair of Systems and Computer Networks, Faculty of Electronics, Wroclaw University of Technology, Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland*

**Abstract**

The P2P multicasting is a significant proposal for development of IPTV and Video on Demand services as it enables rapid deployment at low cost. This paper deals with the overlay network design for P2P multicasting. In the problem formulated as an Integer Programming both multicast flows and link capacities are to be optimized jointly. The objective is to minimize the network cost of access links. An heuristic algorithm based on the Lagrangean relaxation is proposed. Simulations were used in order to examine the efficiency of the proposed method against optimal results.

**Keywords:** overlay networks, P2P multicasting, network design problem, optimization

## 1. Introduction

Live multimedia streaming applications (such as Internet TV) are gaining much interest in recent years. Streaming systems applying only traditional unicast transmission do not scale well – network infrastructure of media streaming servers providing the content can be easily saturated with the increase of the number of participating users. Therefore to scale to larger numbers of participating users, overlay tree or mesh topologies are typically created. Thus, the system can include even millions of users without overwhelming the streaming servers. Since each node connected to the overlay tree can both download and upload the streaming content, the system can be called also Peer-to-Peer (P2P) multicasting or application multicasting. A popular example application of P2P multicasting is the broadcast of a sport event consisting of simple text description of the score (low volume), audio (medium volume), or video (high volume). We must be aware of the fact that overlay multicasting systems can yield relatively large delay varying up to 20 seconds. However, in most cases this delay is tolerable by users. An important aspect of P2P multicast is fairness, i.e. each member node should transmit the same volume that it receives. Moreover, to provide robustness any peer should not be a bottleneck. Thus, rather than a single P2P multicast tree, multiple trees should be established, then each node receives portions of the multicast stream via different routes [1], [8], [11]. For more information on various aspects of P2P and multicasting refer to [1-2], [4-6], [8-15].

Most of previous works on P2P streaming (e.g. [1-2], [8], [11], [12] [15]) focus on the problem how to construct an overlay multicast topology and assign streaming rates to created trees. All these works assume that links connecting peers to the overlay networks are given, i.e. the overlay network is given. Authors [6] and [9] take into account also dimensioning of access links, however the proposed method is based only on simulations, i.e. multicast trees are computer based on the given routing algorithm for all the trees, then by summing up the volume on each tree, the amount of bandwidth to be leased on every overlay link is determined.

In this paper we address a new network design problem for P2P multicasting. Simply put, for the given streaming rate we want to determine how much resource capacity is needed for each peer and how to economically distribute the streaming content in the overlay network using P2P multicasting. The former goal consists in selection of one access link type among options proposed by the ISP (Internet Service Provider) selected by a given peer. The latter goal is to construct the P2P multicast trees in the overlay topology subject to capacity constraints. The overall objective of the proposed problem is to minimize the cost of the network. i.e. the sum of all access link costs expressed e.g. in Euro/Month. It should be noted that since overlay multicast networks are built on top of a general Internet unicast infrastructure rather than point-to-point links, the problem of overlay network design for P2P multicasting is somewhat different than in networks that do have their own links [9].

## 2. Mathematical Formulation

In this section we present an mathematical model of the network design problem for P2P multicasting in overlay networks. Our assumptions follow from real systems and previous works. According to analysis presented in [15], nodes' capacity constraints are typically sufficient in overlay networks. Furthermore, in the concept of overlay networks usually the underlay core network is considered as overprovisioned and the only bottlenecks are access links [1], [13]. Therefore, the objective of our problem is to select the access link for each peer from the pool of link types offered by ISP. Let $y_{vk}$ denote a binary decision variable which is 1 if node $v$ is connected to the overlay network by a link of type $k$; 0 otherwise. For each access link type offered by a given ISP we know download capacity (denoted as $d_{vk}$), upload capacity (denoted as $u_{vk}$) and cost (denoted as $\xi_{vk}$).

The second type of decision variables is necessary to construct multicast trees. A common approach in the literature is to use Steiner trees to model multicasting. However, the canonical Integer Programming (IP) formulation of Steiner trees includes a large number of cut constraints that guarantees that the tree is connected [4]. In our approach we propose to use binary variable $x_{vwtl}$ to model the mulitcast tree. We assume that $x_{vwtl} = 1$ if in the multicast tree $t$ there is a link from node (peer) $v$ to node $w$ and node $v$ is located on level $l$ of tree $t$; 0 otherwise (binary). Index $t$ is associated with multicast trees, but if there is only one tree in the network we can ignore this index. We assume that the root of the tree is located on level 1. All children of the root (peers that have a direct link from the root) are located on level 2, etc. The proposed notation enables us to set the value of $L$ as a limit on the maximal depth of the tree. Peers – besides participating in overlay trees – can also use other network services and resources. Therefore, for each peer we are given constants $a_v$ and $b_v$ denoting, respectively, download and upload background traffic. To formulate the problem we use the notation proposed in [7].

**Network Design for P2P Multicasting Problem (NDPMP)**

**indices**

| | |
|---|---|
| $v,w = 1,2,\ldots,V$ | overlay nodes (peers) |
| $k = 1,2,\ldots,K_v$ | access link types for node $v$ |
| $t = 1,2,\ldots,T$ | multicast trees |
| $l = 1,2,\ldots,L$ | levels of the multicast tree |

**constants**

| | |
|---|---|
| $a_v$ | download background transfer of node $v$ |
| $b_v$ | upload background transfer of node $v$ |
| $\xi_{vk}$ | cost of link type $k$ for node $v$ |
| $d_{vk}$ | download capacity of link type $k$ for node $v$ (b/s) |
| $u_{vk}$ | upload capacity of link type $k$ for node $v$ (b/s) |
| $r_v$ | = 1 if node $v$ is the root of the tree; 0 otherwise |
| $q_t$ | the streaming rate of tree $t$ (b/s) |
| $M$ | large number |

**variables**

| | |
|---|---|
| $x_{vwtl}$ | = 1 if there is a link from node $v$ to node $w$ is in multicast tree $t$ and $v$ is located on level $l$ of tree $t$; 0 otherwise (binary) |
| $y_{vk}$ | = 1 if node $v$ is connected to the overlay network by a link of type $k$; 0 otherwise (binary) |

**objective**

$$\text{minimize} \quad F = \sum_v \sum_k y_{vk}\, \xi_{vk} \tag{1}$$

**constraints**

$$\sum_{v \neq w} \sum_l x_{vwtl} = (1 - r_w) \quad w = 1,2,\ldots,V \quad t = 1,2,\ldots,T \tag{2}$$

$$\sum_{w \neq v} \sum_t x_{vwt1} \leq M r_v \quad v = 1,2,\ldots,V \tag{3}$$

$$\sum_{w \neq v} x_{vwt(l+1)} \leq M \sum_{w \neq v} x_{wvtl} \quad v = 1,2,\ldots,V \quad t = 1,2,\ldots,T \quad l = 1,2,\ldots,L-1 \tag{4}$$

$$\sum_k y_{vk} = 1 \quad v = 1,2,\ldots,V \tag{5}$$

$$a_v + \sum_t q_t \leq \sum_k y_{vk}\, d_{vk} \quad v = 1,2,\ldots,V \tag{6}$$

$$b_v + \sum_{w \neq v} \sum_t \sum_l x_{vwtl}\, q_t \leq \sum_k y_{vk}\, u_{vk} \quad v = 1,2,\ldots,V \tag{7}$$

The objective function (1) is the cost of access links of the overlay P2P multicast network. Since for each tree $t = 1,2,\ldots,T$ each node $w = 1,2,\ldots,V$ – except the source node of the tree $t$ ($r_w = 1$) – must have exactly one parent node, we introduce constraint (2). Constraint (3) guarantees that node $v$ can be a parent of the first level link, only if it is the root node. Constraint (4) is in the model to meet the requirement that each node $v = 1,2,\ldots,V$ cannot be a parent on level $(l + 1)$ if it is not a child on level $l$. Constraint (5) guarantees that one access link is selected for each overlay node. (6) is a download capacity constraint, i.e. download capacity of each node must be greater than or equal to the streaming rate of all trees and the download background traffic. Analogously, (7) is the upload capacity constraint Note that the left-hand side of (7) is equal to the total upload transfer of $v$ which follows from the number of children nodes, the streaming rate and the background traffic.

## 3. Lagrangean Relaxation and Decomposition

In this section we propose the Lagrangean relaxation (LR) of the problem NDPMP given by (1)-(7) in order to construct subgradient-based approach. The dual solution of variables $y_{vk}$ generated in subsequent iterations of the subgradient algorithm will be used as an input solution for a heuristic algorithm aimed to find a feasible primal solution. In other words, the subgradient approach will be used as an intelligent search of the solution space. Our approach follows from the fact that the structure of the considered problem suggests that the optimal primal solution point may not be available from the optimal dual solution.

We relax constraint (7) in a Lagrangean fashion by associating nonnegative variables $\lambda_v$. Let $\boldsymbol{\lambda} = [\lambda_1, \lambda_2,\ldots,\lambda_v]$ denote the a vector of nonnegative Lagrangean multipliers. The resulting formulation of Lagrangean dual problem can be stated as follows:

***LR*1**
**objective**

$$\text{minimize} \quad \varphi(\boldsymbol{\lambda}) = \textstyle\sum_v\sum_k y_{vk}\, \xi_{vk} + \sum_v \lambda_v(b_v + \sum_{w\neq v}\sum_t\sum_l x_{vwtl}\, q_t - \sum_k y_{vk}\, u_{vk}) \qquad (8)$$

**subject to** (2), (3), (4), (5) and (6)

After regrouping the relevant terms, the dual function $\varphi(\boldsymbol{\lambda})$ can be rewritten as

$$\varphi(\boldsymbol{\lambda}) = \textstyle\sum_v\sum_k y_{vk}\,(\xi_{vk} - \lambda_v\, u_{vk}) + \sum_v \lambda_v b_v + \sum_v\sum_{w\neq v}\sum_t\sum_l \lambda_v\, x_{vwtl}\, q_t \qquad (9)$$

It is easy to see that $LR1$ decomposes into two subproblems, where the first subproblem is in the $y$ variables and is stated as follows:

***SP*1**
**objective**

$$\text{minimize} \quad \textstyle\sum_v\sum_k y_{vk}\,(\xi_{vk} - \lambda_v\, u_{vk}) \qquad (10)$$

**subject to** (5), (6).

Let $q = \sum_t q_t$ denote the overall streaming rate. $SP1$ decomposes into $V$ following subproblems, one for each node $v = 1,2,\ldots,V$:

**$SP1_v$**
**objective**

$$\text{minimize} \quad \textstyle\sum_k y_{vk}\,(\xi_{vk} - \lambda_v\, u_{vk}) \qquad (11)$$

**subject to**

$$a_v + q \leq \textstyle\sum_k y_{vk}\, d_{vk} \qquad (12)$$

$$\textstyle\sum_k y_{vk} = 1 \qquad (13)$$

Since the number of access link types (index $k$) is relatively small, to solve problem $SP1_v$ we can simply check by inspection all possible solutions. If access link types are sorted according to increasing values of capacity and cost we can start the inspection setting $k = 1$ and checking subsequent link types until the constraint (12) is satisfied.

The second subproblem of $LR1$ is as follows:

***SP*2**
**objective**

$$\text{minimize} \quad \sum_v \sum_{w \neq v} \sum_t \sum_l \lambda_v x_{vwtl} q_t \tag{14}$$

**subject to** (2), (3) and (4).

Note that *SP*2 can be decomposed into following subproblems for each $t = 1,2,\ldots,T$:

***SP*2$_t$**
**objective**

$$\text{minimize} \quad \sum_v \sum_{w \neq v} \sum_l \lambda_v x_{vwtl} q_t \tag{15}$$

**subject to**

$$\sum_{v \neq w} \sum_l x_{vwtl} = (1 - r_w) \quad w = 1,2,\ldots,V \tag{16}$$

$$\sum_{w \neq v} x_{vwtl} \leq M r_v \quad v = 1,2,\ldots,V \quad l = 1 \tag{17}$$

$$\sum_{w \neq v} x_{vwt(l+1)} \leq M \sum_{w \neq v} x_{wvtl} \quad v = 1,2,\ldots,V \quad l = 1,2,\ldots,L-1 \tag{18}$$

Notice that $\beta_{vt} = \lambda_v q_t$ can be interpreted as a cost of node $v$ to be a parent node in tree $t$. The objective of $SP2_t$ is to minimize the cost of parent nodes. The following simple algorithm solves problem $SP2_t$:

**Algorithm ASP2**
Step 0. Set $x_{vwtl} = 0$ for each $v = 1,2,\ldots,V$, $w = 1,2,\ldots,V$, $l = 1,2,\ldots,L$, $v \neq w$.
Step 1. Find $s = \text{argmin } \lambda_v \quad v = 1,2,\ldots,V$. If $r_s = 1$ go to step 2. Otherwise go to step 3.
Step 2. Set $x_{swt1} = 1$ for each $w = 1,2,\ldots,V$, $w \neq s$.
Step 3. Set $x_{vst1} = 1$ for $v : r_v = 1$. Set $x_{swt2} = 1$ for each $w = 1,2,\ldots,V$, $w \neq s$, $w \neq v$.

The idea behind the algorithm is as follows. In step 1 we set all variables to 0. Recall that $\beta_{vt} = \lambda_v q_t$ denotes a cost of node $v$ in tree $t$. Since the objective is to minimize (15) we must select as parents the "cheapest" (in terms of $\beta_{vt}$) nodes. Therefore, in Step 1 we find the "cheapest" node denoted as $s$. If $s$ is the root of tree $t$ ($r_s = 1$) links to all other nodes $w = 1,2,\ldots,V$, $w \neq s$ are originated at node $s$ and are located on level $l = 1$ (Step 2). Otherwise, due to constraint (17) guaranteeing that each link in level $l = 1$ must originate at root, we establish a link between the root of tree $t$ ($v : r_v = 1$) and the "cheapest" node $s$. Next we connect all other nodes (expect root node $v$ and node $s$) using level $l = 2$. This simple procedure assures that the obtained solution is feasible (all constraints of $SP2_t$ are satisfied) and optimal.

The solution yielded by algorithm ASP2 leads to the situation that the "cheapest" node $s$ ($s = \text{argmin } \lambda_v \quad v = 1,2,\ldots,V$) is the parent node of all other peers except the root node and itself. Since the upload capacity of each node is limited even if we select the largest capacity option, the obtained solution in almost all cases will not satisfy the upload capacity constraint (7) of the primal problem. Therefore, we modify the $SP2_t$ problem by adding an additional capacity constraint on upload flow. Let $u_v^{max} = \max u_{vk} \quad k = 1,2,\ldots,K_v$ denote the maximum value of upload capacity that can be selected for node $v$. We assume that for each node $v$ the upload flow of $v$ and the background traffic of $v$ cannot exceed $u_v^{max}$. The modified problem $SP2_t$ is as follows:

***MSP*2$_t$**
**objective** (15)
**subject to** (16), (17), (18) and

$$b_v + \sum_{w \neq v} \sum_t \sum_l x_{vwtl} q_t \leq u_v^{max} \quad v = 1,2,\ldots,V \tag{19}$$

Note that $MSP2_t$ is an NP-complete problem, since it can be reduced to the knapsack problem. Therefore, we propose an algorithm that finds a lower bound of the $MSP2_t$ problem. Let $f_v$ denote the upload flow of node $v$. We assume that set $A$ include indices of nodes sorted according to increasing values of $\lambda_v$. Let *first*($A$) return the first index of set $A$. Let $C$ denote the value of the objective function (15).

**Algorithm AMSP2**
Step 0. Set $f_v = b_v$ for each $v = 1,2,\ldots,V$. Set $C = 0$.
Step 1. Let $v$ be the root node, i.e. $r_v = 1$. Set $f_v = f_v + q_t$ and $F = F + \lambda_v q_t$ .

Step 2. Find $v = first(A)$.
Step 3. If $f_v + q_t \le u_v^{max}$ go to Step 4. Otherwise set $A = A - \{v\}$ and go to step 2.
Step 4. Set $f_v = f_v + q_t$ and $F = F + \lambda_v q_t$ . Go to step 3.

The idea of algorithm AMSP2 is to saturate subsequent nodes sorted according to increasing values of $\lambda_v$ up to the maximal upload capacity limit of each node. In step 1 we provide that the root node has at least one child. Steps 3-4 update the upload flow and objective function according to the selected strategy. To solve the dual problem and obtain starting values of $y_{vk}$ variables for the heuristic algorithm we apply classical subgradient algorithm as proposed in [7]. The value $C$ and values of upload flow $f_v$ $v = 1,2,\ldots,V$ yielded by the AMSP2 algorithm are used in the subgradient algorithm to calculate the dual function and subgradient of dual function at $\lambda$, respectively. Since some of the constraints are relaxed by the Lagrangean multipliers, the solution might be infeasible. To construct a feasible solution a heuristic algorithm has to be employed. The values of $y_{vk}$ variables yielded by subgradient search are taken as initial values. Next the following procedure is applied. For each tree we create at least one link from the root node to a peer selected according to the decreasing values of upload capacity, thus the constraint (3) is satisfied. Next, we create subsequent levels of the tree saturating upload capacity of peers, which are processed in decreasing order of upload capacity. This guarantees that peers having relatively more upload capacity will be located closer to the root and provide streaming for peers having lower bandwidth. If after this procedure there are peers not connected to each tree, we increase capacity of selected peer and try to connect missing peers.

## 4. Preliminary Computational Results

In order to solve the model (1)-(7) in optimal way we used CPLEX 11.0 solver [3]. The heuristic based on Lagrangean relaxation approach was coded in C++. Since the first goal of simulations was to compare results of LR approach against optimal results, we had to reduce sizes of the problem instances in order to obtain optimal results approximately in reasonable time. Due to several experiments we decide to test networks consisting of 20 overlay nodes (peers). We use DSL price lists of four ISPs: two of them operate in Poland (TP and Dialog) and two other operate in Germany (DT and Arcor). Each node is randomly assigned to one of ISPs and choose any option included in the price list. The values of download background transfer were selected at random between 512 kb/s and 1024 kb/s Analogously, the values of upload background transfer were selected at random between 64 kb/s and 128 kb/s. The streaming rate of 360 kb/s was divided proportionally to 1, 2, 3 or 4 multicast trees. We examined trees consisting of 2-8 levels. In Table 1 we show the comparison of optimal and heuristic results. Results of Lagrangean relaxation approach are in general 3.72% worse then optimal ones. The detailed analysis indicates larger gaps between both approaches are observed for smaller number of levels. When the number of levels increases, LR heuristic approaches optimal results.

**Table 1.** Results of CPLEX (OPT) and Lagrangean relaxation (LR) on 20-node network

| Levels | Trees | OPT | LR | Trees | OPT | LR | Trees | OPT | LR | Trees | OPT | LR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 1 | 1232 | 1257 | 2 | 809 | 854 | 3 | 789 | 834 | 4 | 660 | 854 |
| 3 | 1 | 819 | 869 | 2 | 620 | 670 | 3 | 600 | 635 | 4 | 605 | 625 |
| 4 | 1 | 635 | 670 | 2 | 620 | 625 | 3 | 600 | 625 | 4 | 605 | 625 |
| 5 | 1 | 635 | 660 | 2 | 620 | 630 | 3 | 600 | 620 | 4 | 605 | 620 |
| 6 | 1 | 635 | 660 | 2 | 620 | 640 | 3 | 600 | 630 | 4 | 605 | 605 |
| 7 | 1 | 635 | 660 | 2 | 620 | 625 | 3 | 600 | 605 | 4 | 605 | 605 |
| 8 | 1 | 635 | 660 | 2 | 620 | 625 | 3 | 600 | 605 | 4 | 605 | 605 |

On Fig. 1 we report results of LR algorithm obtained for 100-node network. The simulation scenario was analogous to 20-node network. The streaming rate of 360 kb/s was divided proportionally to 1, 2, 3 or 4 multicast trees. The tests were run for the number of levels in the range from 4 to 9. We can notice that for 4 levels dividing of the stream into several trees can significantly reduce the network cost. However, when the number of levels grows, the gap in results between particular number of trees decreases.

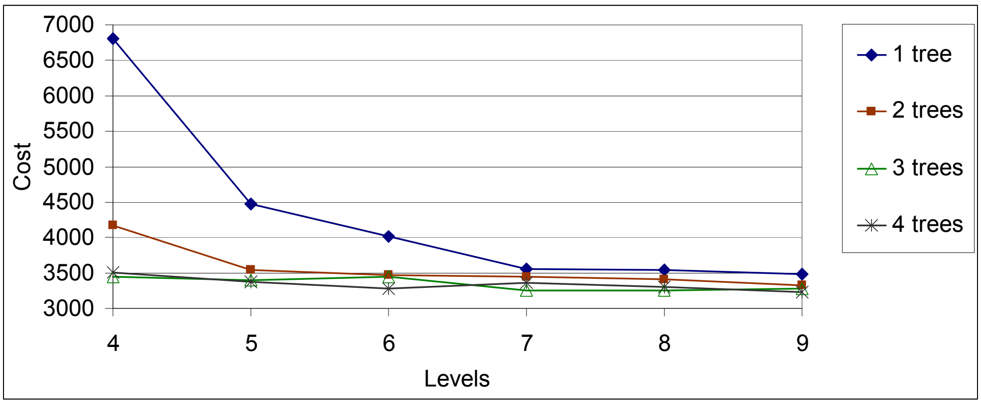


**Figure 1.** Network cost as a function of number of levels and number of multicast trees– results of LR algorithm on 100-node network

**Acknowledgement.** This work is supported by The Polish Ministry of Science and Higher Education under the grant which is being realized in years 2008-2011.

## References

[1] B. Akbari, H. Rabiee and M. Ghanbari, *An optimal discrete rate allocation for overlay video multicasting*, Computer Communications, No. 31, (2008) pp. 551-562.

[2] Y. Cui, Y. Xue and K. Nahrstedt, *Optimal Resource Allocation in Overlay Multicast*, IEEE Transactions on Parallel and Distributed Systems, Vol. 17, No. 8 (2006) pp. 808-823.

[3] *ILOG CPLEX 11.0 User's Manual*, France (2007).

[4] T. Koch and A. Martin, *Solving Steiner tree problems in graphs to optimality*, Networks Vol.32, No. 3 (1998) pp. 207-232.

[5] B. Leuf, *Peer to Peer: Collaboration and Sharing over the Internet*, Addison Wesley (2002).

[6] L. Lao, J. Cui and M. Gerla, *Multicast Service Overlay Design*, in Proc. of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'05), Philadelphia, Pennsylvania, USA, July 2005.

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

[8] T. Small, B. Li and B. Liang, *Outreach: peer-to-peer topology construction towards minimized server bandwidth costs*, IEEE Journal on Selected Areas in Communications, Vol. 25, No. 1, (2007) pp. 35-45.

[9] S. Shi and J. Turner, *Multicast Routing and Bandwidth Dimensioning in Overlay Networks*, IEEE Journal on Selected Areas in Communications, Vol. 20, No. 8 (2002) pp. 1444-1455.

[10] R. Steinmetz and K. Wehrle (eds.), *Peer-to-Peer Systems and Applications*, Lecture Notes in Computer Science, Vol. 3485, Springer Verlag (2005).

[11] V. Venkataraman, P. Francis and J. Calandrino, *Chunkyspread: Multitree unstructured peer to peer multicast*, in Proc. of the Fifth International Workshop on Peer-to-Peer Systems, Santa Barbara, CA, 2006.

[12] C. Wu and B. Li, *On Meeting P2P Streaming Bandwidth Demand with Limited Supplies*, in Proc. of the Fifteenth Annual SPIE/ACM International Conference on Multimedia Computing and Networking (2008).

[13] G. Wu and C. Tzi-cker, *Peer to Peer File Download and Streaming*, RPE report, TR-185 (2005).

[14] J. Wu (ed.), *Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks*, Auerbach Publications (2006).

[15] Y. Zhu and B. Li, *Overlay Networks with Linear Capacity Constraints*, IEEE Transactions on Parallel and Distributed Systems, Vol. 19, No. 2, (2008) pp. 159-173.