Wrocław University of Technology

P2P Multicasting Network Design Problem - Heuristic Approach

Krzysztof Walkowiak

Agenda

- Motivation
- P2P Multicasting
- Problem Formulation
- Algorithm
- Results
- Conclusions

Agenda

- **Motivation**
- P2P Multicasting
- Problem Formulation
- Algorithm
- Results
- Conclusions

Motivation (1)

- Growing volume of **electronic content** in the Internet (music, movies, e-books, distance learning, software distribution, etc.)
- More than **50%** of network traffic in the Internet is produced by **Peer-to-Peer (P2P)** systems
- Increasing popularity of **overlay networks** including **P2P multicasting** systems (also called **Application-Layer Multicasting**)
- Not many papers on optimization of **link capacity in overlay** P2P multicasting systems

Agenda

- Motivation
- **P2P Multicasting**
- Problem Formulation
- Algorithm
- Results
- Conclusions

P2P Multicasting (1)

- Overlay P2P multicasting uses a **multicast delivery tree** constructed among **peers** (end hosts)
- Different to traditional IP multicast, the **uploading (non-leaf) nodes** in the tree are **normal end hosts**
- Peers are connected to the **overlay network**, which is considered as an **overprovisioned** cloud
- The **content** to be distributed through P2P multicasting can be divided into **two categories**:
  - Elastic content (e.g. data files)
  - Streaming content with specific bit rate requirements (e.g. media streaming)

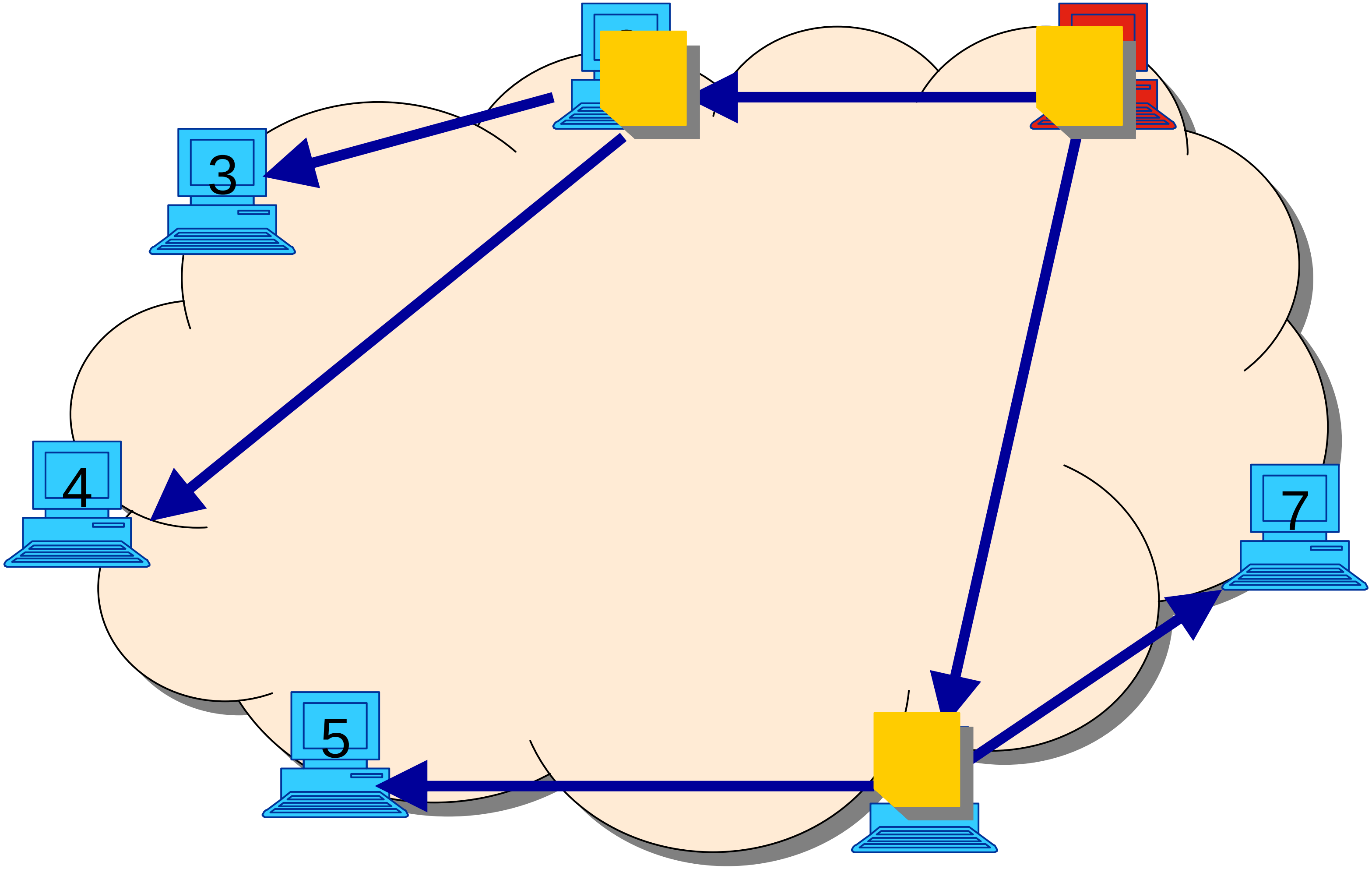
P2P Multicasting (2)

- In **dynamic** P2P multicasting systems peers are potentially vulnerable to situations as peer failures or leaving the multicast session
- In **static** P2P multicasting system all peers interested in participating in the system and are stable connected. Thus, there are not dynamic changes of the structure as in file-sharing P2P systems

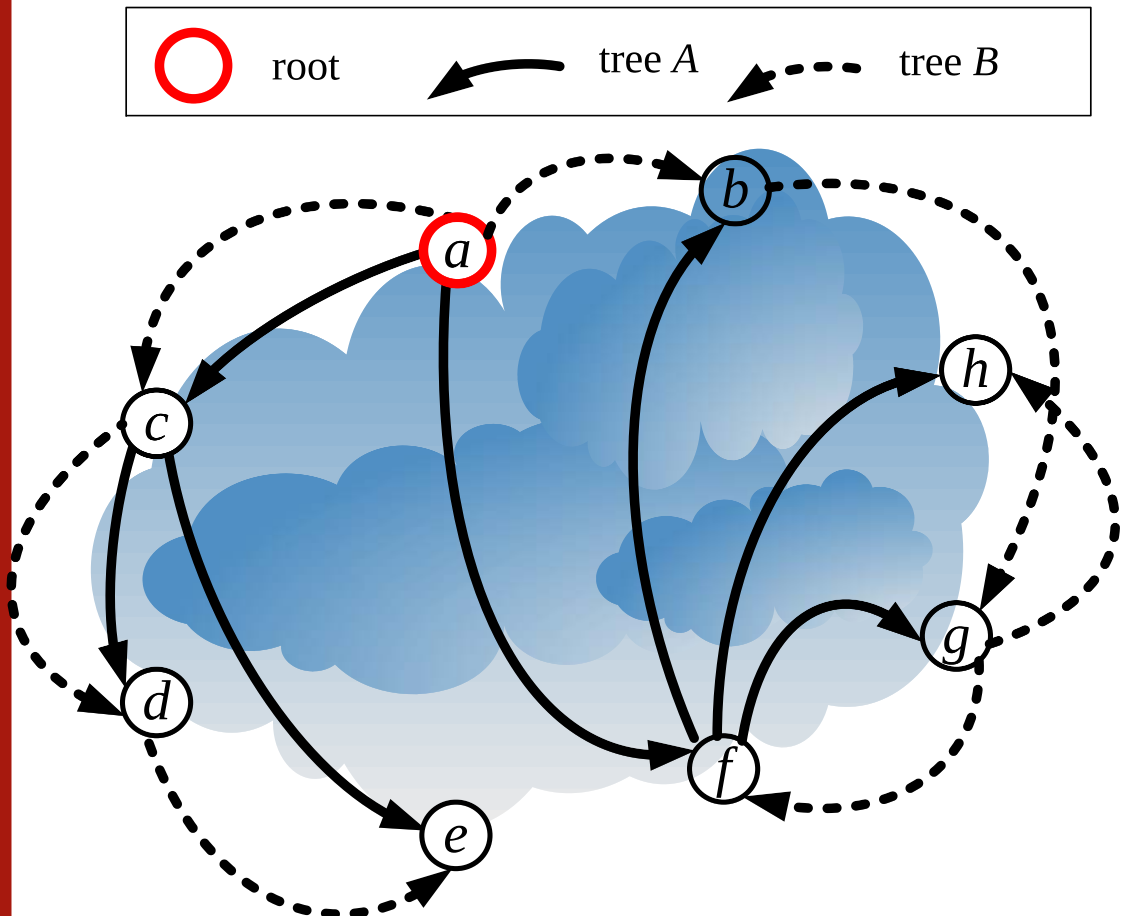
P2P Multicasting (3)

- **Examples** of static P2P multicasting systems:
  - P2P multicast system applied for dissemination of critical information, e.g. weather forecast (hurricane warnings), security software updates, stock exchange data, traffic information, etc
  - Videoconferencing
  - Personal video broadcast in small groups
  - IPTV system using STBs (set-top boxes)
  - CDN (Content Delivery Network)
  - Collaborated workgroup

P2P Multicasting (4)



Modeling of P2P Multicasting (1)



Tree *A*

- Peers on level 1: *a*
- Peers on level 2: *c*, *f*
- Peers on level 3: *d*, *e*, *b*, *h*, *g*

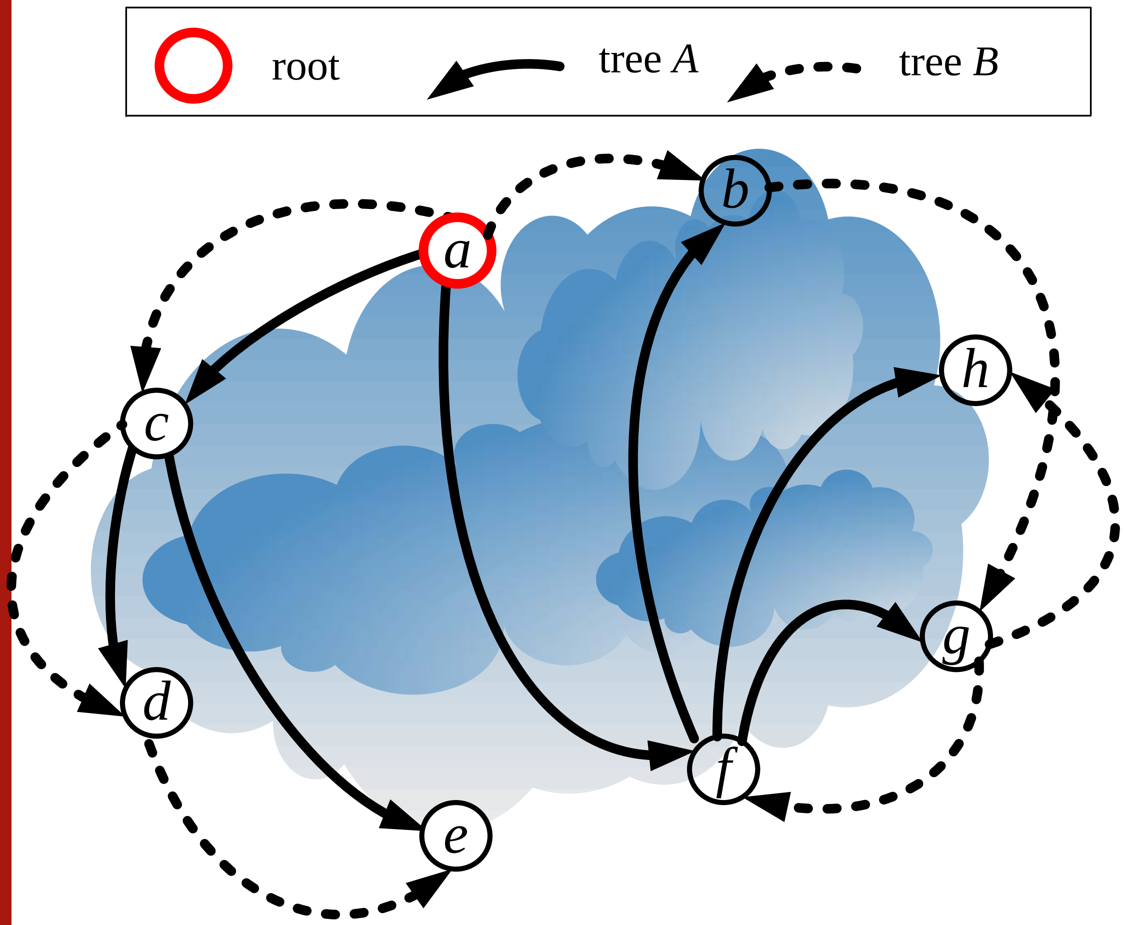
Tree *B*

- Peers on level 1: *a*
- Peers on level 2: *c*, *b*
- Peers on level 3: *d*, *g*
- Peers on level 4: *e*, *f*, *h*,

Modeling of P2P Multicasting (2)

- Binary variable x_{vwtl} models the multicast tree
- x_{vwtl} is 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

Modeling of P2P Multicasting (3)



Tree *A*

x_{acA1} , x_{afA1}, x_{cdA2} , x_{ceA2}, x_{fbA2}, x_{fgA2}, x_{fhA2}

Tree *B*

x_{abB1}, x_{acB1}, x_{cdB2} , x_{bgB2}, x_{deB3}, x_{gfB3}, x_{ghB3}

Agenda

- Motivation
- P2P Multicasting
- **Problem Formulation**
- Algorithm
- Results
- Conclusions

Problem Formulation

Given: number of trees, streaming rates, access link proposals, participating peers, background traffic

Minimize: network cost (cost of access link)

Over: access link selection, P2P flows

Notation (1)

Indices

$w,v = 1,2,\ldots,V$ overlay network nodes (peers)

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

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

$l = 1,2,\ldots,L$ levels of the multicast tree

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)

Notation (2)

constants

a_v	download background transfer of node *v* (bps)
b_v	upload background transfer of node *v* (bps)
ξ_{vk}	cost of link type *k* for node *v*
d_{vk}	download capacity of link type *k* for node *v* (bps)
u_{vk}	upload capacity of link type *k* for node *v* (bps)
r_v	= 1, if node *v* is the root of the tree; 0, otherwise
q_t	the streaming rate of tree *t* (bps)
M	large number

Problem formulation

objective

minimize $F = \sum_v \sum_k y_{vk} \xi_{vk}$ (1)

constraints

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

$\sum_{w \neq v} \sum_t x_{vwt1} \leq M r_v$ $v = 1,2,...,V$ (3)

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

$\sum_k y_{vk} = 1$ $v = 1,2,...,V$ (5)

$a_v + \sum_t q_t \leq \sum_k y_{vk} d_{vk}$ $v = 1,2,...,V$ (6)

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

Agenda

- Motivation
- P2P Multicasting
- Problem Formulation
- **Algorithm**
- Results
- Conclusions

Algorithm Create Tree (1)

- Create Tree is a **constructive** algorithm including several **operators nad functions** (details in the paper)
- We assume that for each node $v = 1,2,\ldots,V$ the access link types are **sorted according** to increasing values of upload capacity and cost
- We assume that $first(A)$ returns the first element of table A
- Let $q = \sum_t q_t$ denote the overall streaming tree rate of all trees

Algorithm Create Tree (2)

Step 0. Initialization of variables, flow variables (*x*) set to 0, capacity variables (*y*), set to minimal feasible value

↓

Step 1. Initial tree construction, „best" nodes are connected to the root for each tree

↓

Step 2. Main loop, construction of trees and optional increasing of link capacity

Algorithm Create Tree (3)

<u>Step 0.</u> (*Initialization of variables*)

Set $x_{vwtl} = 0$ for each $w = 1,2,\dots,V$, $v = 1,2,\dots,V$, $t = 1,2,\dots,T$, $l = 1,2,\dots,L$, $v \neq w$.

Set y_{vk} as the minimal values that guarantee sufficient download capacity for each node v expect the root node (i.e. $d_{vk} \geq a_v + q$) and the sufficient upload capacity for the root node r (i.e. $u_{rj} \geq b_r + q$).

Algorithm Create Tree (4)

Step 1. (*Connecting nodes to the root*)

Set l = 1. Create table A containing all trees sorted in decreasing order of streaming rate q_t. Create table B containing all nodes except the root node sorted in decreasing order of residual upload capacity of each node.

a) If $A = \varnothing$ go to Step 2. Otherwise, go to Step 1b.

b) Calculate $t = first(A)$, $v = first(B)$ and $x_{rvtl} = 1$. Next, set $A = A - \{t\}$ and $B = B - \{v\}$. Go to Step 1a.

Algorithm Create Tree (5)

Step 2. (*Main loop of the algorithm*)

If $l > L$ go to Step 3

a) If *istrasnfer*(l) = 0 set $l = l + 1$ and go to Step 2. Otherwise go to Step 2b.

b) Set t = *ftree*(l), v = *fpnode*(t, l), w = *fcnode*(v, t, l) and $xvwtl = 1$. Go to Step 2a

Step 3. If *istree*() = 1 stop the algorithm, a feasible solution exists. Otherwise, go to Step 4.

Step 4. If *isupdate*() = 0, stop the algorithm, there is not feasible solution. Otherwise, go to Step 5.

Step 5. Set v = *updatenode*(). Find k, for which $y_{vk} = 1$. Set $y_{vk} = 0$, $k = k + 1$, $y_{vk} = 1$, $l = 1$ and go to Step 2

Agenda

- Motivation
- P2P Multicasting
- Problem Formulation
- Algorithm
- **Results**
- Conclusions

Results (1)

- 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 selects any option included in the price list
- The values of **download background** transfer are selected at random between 512 kbps and 1024 kbps
- Analogously the values of **upload background** transfer were selected at random between 64 kbps and 128 kbps

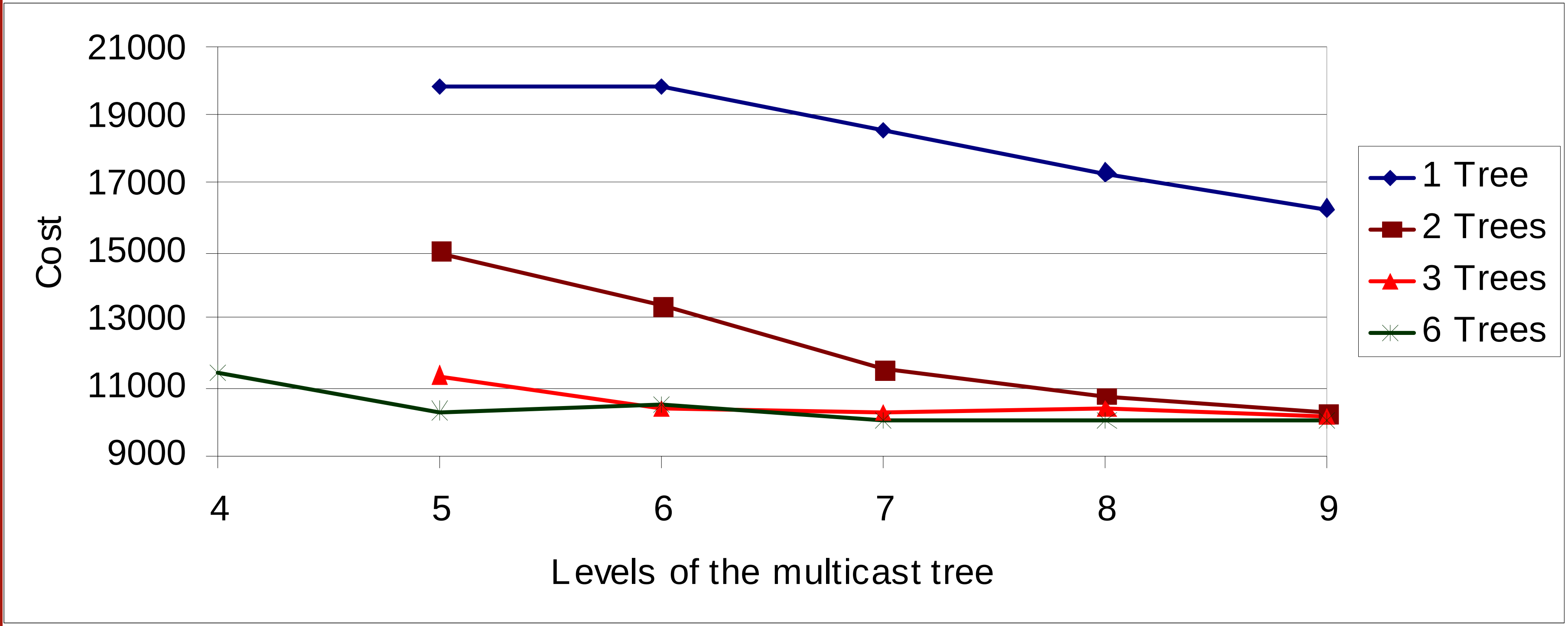
Results (2)

- To obtain **optimal results** we use CPLEX 11.0 solver
- Due to complexity of the problem only for **small networks (20 nodes)** the CPLEX can yield optimal results in reasonable time of approximately one hour
- The **number of multicast trees** was in the range 1-4
- The **number of levels** was in the range 2-8
- We compared the Create Tree algorithm also against **Lagrangean Relaxation** algorithm
- The Create Tree algorithm was tested also for **300-node networks**

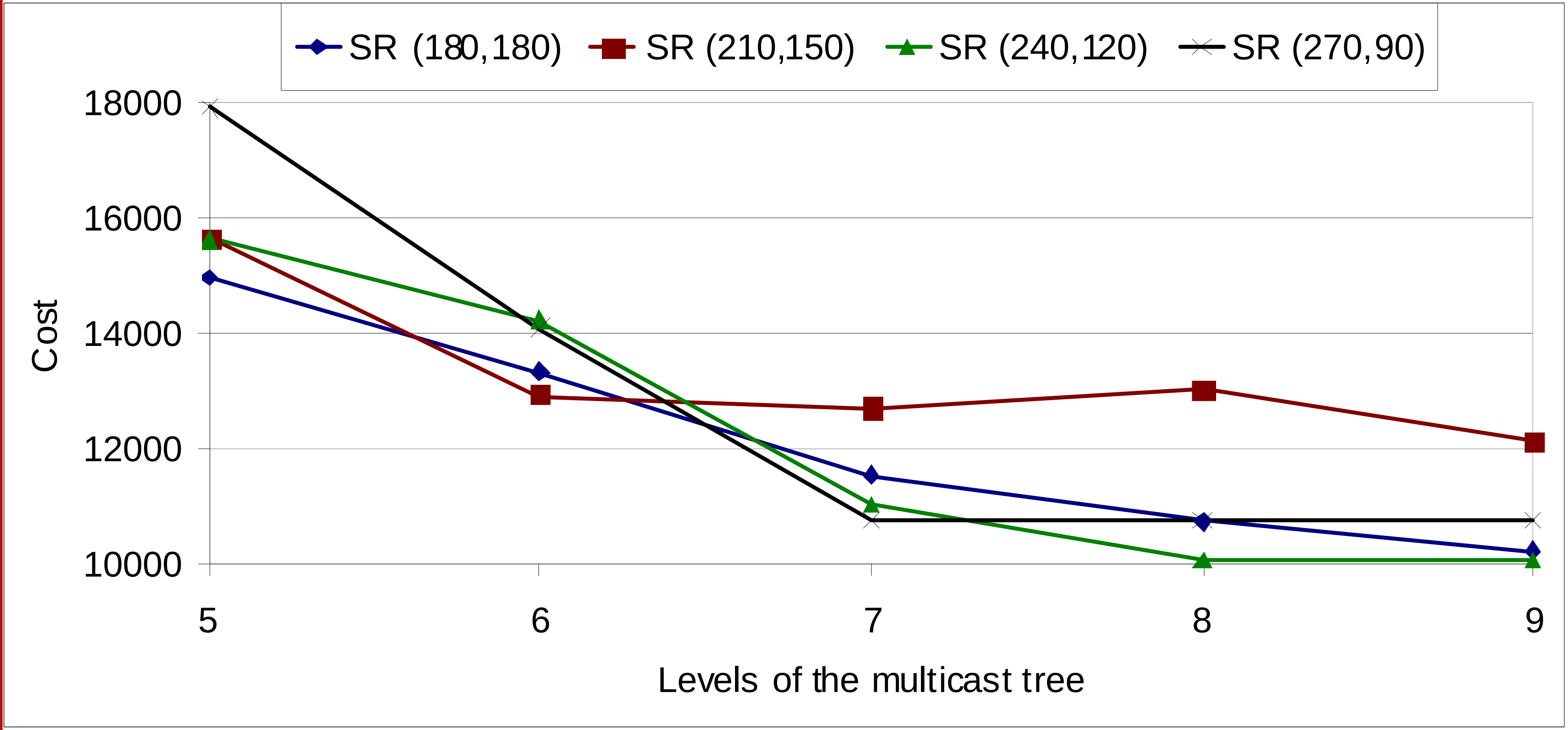
Heuristic algorithm versus optimal results and Lagrangean relaxation results (20-node network)

		Objective function (cost)				Execution time [s]		
T	*L*	HEUR	OPT	LR	HEUR vs. OPT	HEUR	OPT	LR
1	2	1262	1232	1257	2.38%	0.04	0.74	1.53
1	3	869	819	869	5.75%	0.05	3.28	2.98
1	4	670	635	670	5.22%	0.05	0.43	3.98
1	5	645	635	660	1.55%	0.04	0.51	4.44
1	6	645	635	660	1.55%	0.04	0.91	5.11
1	7	645	635	660	1.55%	0.05	1.31	5.91
1	8	645	635	660	1.55%	0.06	1.54	6.73

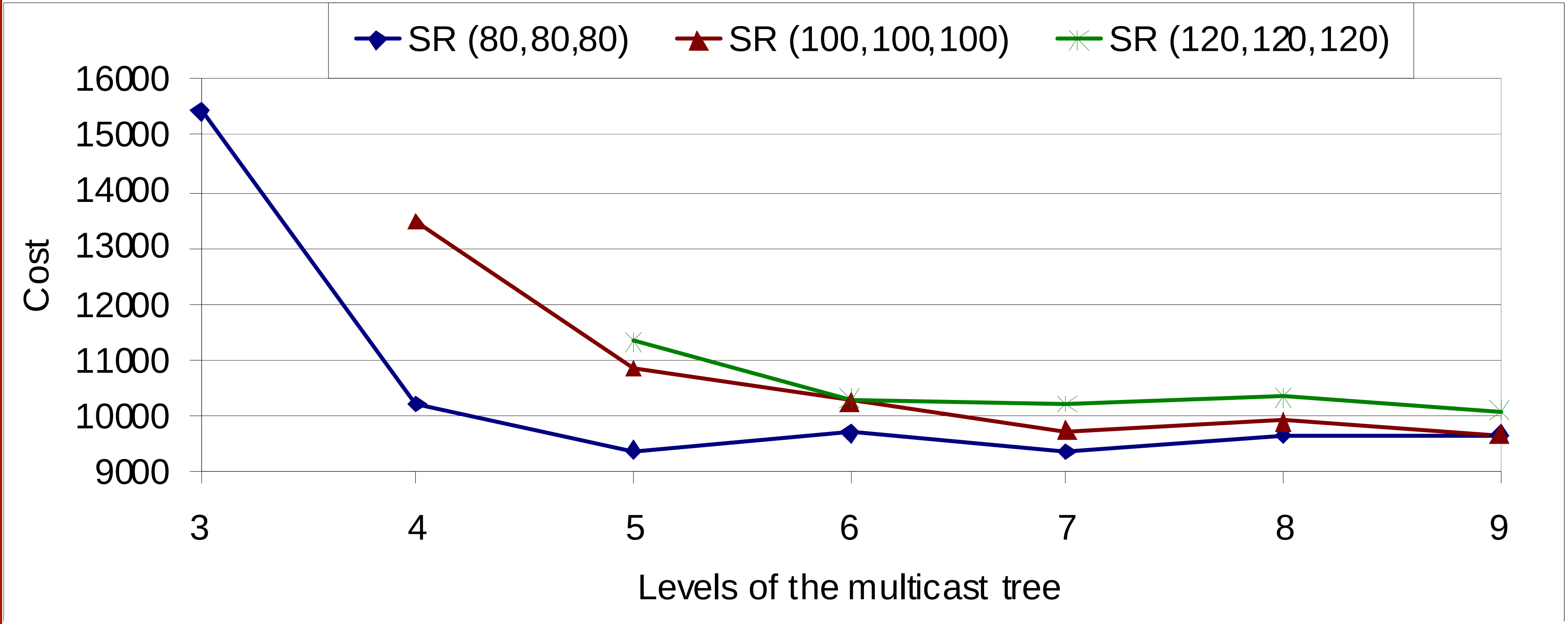
The network cost as a function of tree levels for various number of multicast trees (300-node network)



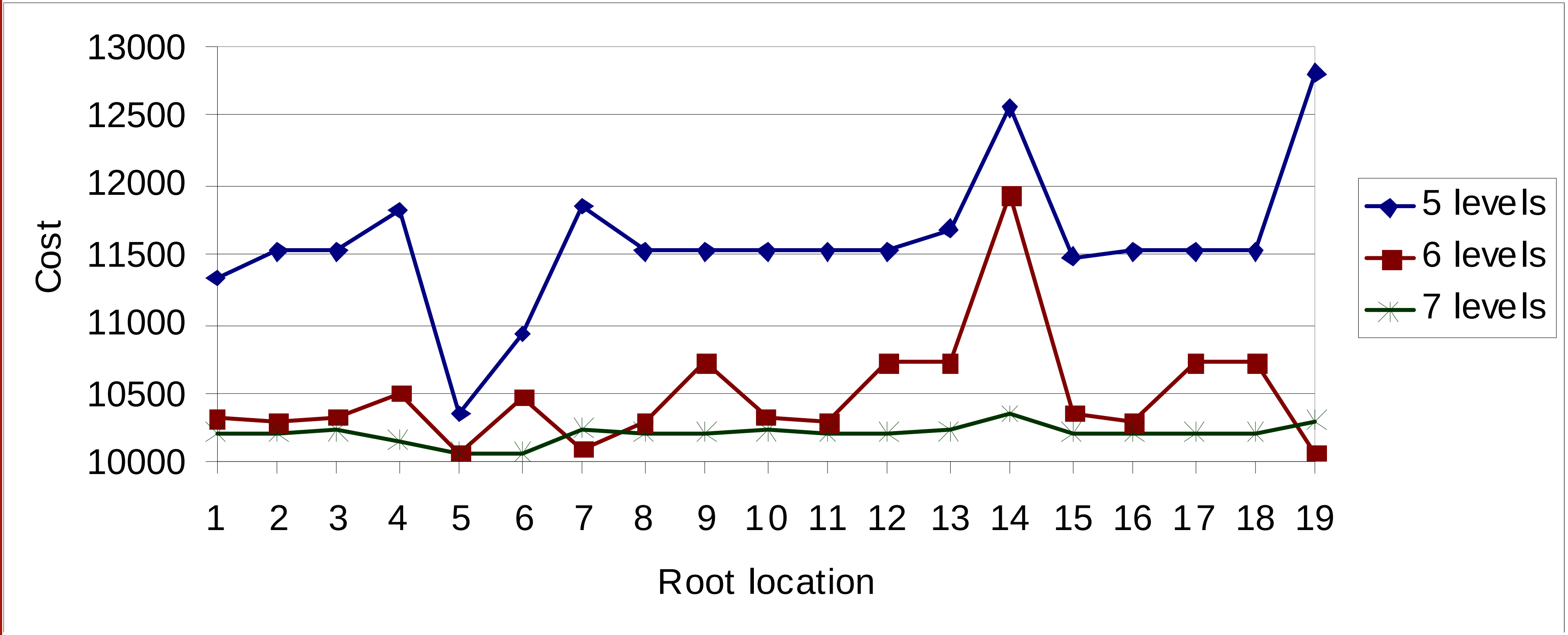
The network cost as a function of tree levels for 2 trees and various scenarios of streaming rate (SR) split (300-node network)



The network cost as a function of tree levels for 3 trees and various scenarios of streaming rate (SR) split (300-node network)



The network cost as a function of the root location for various values of tree levels (300-node network)



Agenda

- Motivation
- P2P Multicasting
- Problem Formulation
- Algorithm
- Results
- **Conclusions**

Conclusions

- An **optimization model** of a P2P multicasting system as an Integer Program was formulated
- **Heuristic algorithm** was developed and examined
- According to numerical experiments the proposed algorithm provides results **close to optimal**
- The influance of various parameters (e.g., number of levels, streaming rate) on the cost was evaluated
- In future work we want to develop:
  - **New MIP formulations** related to P2P multicasting
  - Effective **heuristic algorithms** including evolutionary computing

Thank you for attention

Krzysztof.Walkowiak@pwr.wroc.pl