

# ILP Modeling of Many-to-Many Replicated Multimedia Communication

Krzysztof Walkowiak<sup>a</sup>, Damian Bulira<sup>a</sup>, and Davide Careglio<sup>b</sup>

<sup>a</sup> Department of Systems and Computer Networks, Wrocław University of Technology, Wrocław, Poland

<sup>b</sup> Advanced Broadband Communication Center (CCABA), Universitat Politècnica de Catalunya, Barcelona, Spain

**Abstract**—On-line communication services were evolving from a simple text-based chats towards sophisticated videopresence appliances. The bandwidth consumption of those services is constantly growing due to the technology development and high user and business needs. That fact leads us to implement optimization mechanisms into the multimedia communication scenarios. In this paper, the authors concentrate on many-to-many (m2m) communication, that is mainly driven by the growing popularity of on-line conferences and telepresence applications. An overlay model where m2m flows are optimally established on top of a given set of network routes is formulated and a joint model where the network routes and the m2m flows are jointly optimized. In the models, the traffic traverses through replica servers, that are responsible for stream aggregation and compression. Models for both predefined replica locations and optimized server settlement are presented. Each model is being followed by a comprehensive description and is based on real teleconference systems.

**Keywords**—ILP modeling, many-to-many communication, network optimization, replica location.

## 1. Introduction

Since the beginning of the Internet, network flow paradigms have undergone significant transformation. From a one-to-one transmission that can be represented by fetching a website from a server or simple one-to-one Voice over IP call those paradigms evolved into sophisticated schemes with complex traffic matrix. To optimize the traversal of the same information from one host to the group of others, one-to-many (multicast) applications were introduced. A good example of that is IP TV streaming in triple-play services (Internet, phone and TV) [1] or synchronization messages exchange in Network Time Protocol [2]. Furthermore, one-to-one-of-many (anycast) can be distinguished. In anycast, packets are routed to one of many servers – that can be represented by a common address – with the lowest path cost from a source to a destination. Such distributed networks are called Content Delivery Networks (CDN) and they play the main role in current Internet-based business [3]. In this paper, the authors focus on many-to-many (m2m) communication as one of the fastest emerging paradigms and propose ILP models of offline problems related to optimization of m2m flows using replica servers. To achieve this, the m2m transmission with both anycast and unicast paradigm is modeled. The former, similarly to CDN, is used dur-

ing the replica selection phase and the latter to transfer the data from the selected server, back to the client. In this type of transmissions, all hosts exchange the information with every other host in the m2m group. The information is forwarded first to the replica server (rendezvous point), which in turn propagates proper data to other hosts that take part in the m2m group. The examples of such traffic are: video and teleconferencing, distance learning, multiplayer on-line gaming, distributed computing, etc. The authors focus on videoconferencing as the widespread and demanding example of m2m service. Moreover, a business need for videoconference system is not anymore a nice to have feature for the enterprise, but an essential day-to-day tool that makes the business more effective and successful. According to Cisco, business videoconferencing will grow six fold between 2011 and 2016 [4]. The authors of the report claim, that business videoconferencing traffic is growing significantly faster than overall business IP traffic, at a compound annual growth rate of 48% over the forecast period.

Furthermore, using replication in videoconferencing, bandwidth used for the transmission is significantly reduced. Replicas not only aggregate the traffic, but also perform stream modifications such as format change or compression. Currently, end nodes in videoconferencing are mobile devices, PCs, dedicated videophones or special telepresence equipment. Each of them requires different audio and videostream formats due to available computational resources. Using replicas, complex multimedia transcoding is moved from end nodes to highly efficient dedicated servers. Moreover, encoded stream requires less bandwidth, that decreases network congestion and provide higher level of Quality of Service to end users.

The main contributions of this paper are integer linear programming models for many-to-many transmission in computer networks where rendezvous points are used. The authors propose overlay and joint models assuming combined optimization of overlay and underlying networks. Moreover, two different strategies of locating replica servers in the network are presented. In the first strategy, the location of the servers is known and only the client assignment and network flows have to be optimized. In the latter case, the location of the replica servers is unknown and is a subject of optimization. The models support video conference applications, but can be easily redefined for other type of m2m traffic.

This paper is an extended version of the paper [5], presented at 17th Polish Teletraffic Symposium PTS 2012, held in Zakopane, Poland on December 5–7, 2012. This extended paper contains the new results, including a ILP formulations of replica location problem for many-to-many multimedia communication. To the best of our knowledge, this work is the first one that addresses the problem of replica location in m2m networks.

The remainder of this paper is organized as follows. Section 2 provides related works study on many-to-many communication. Section 3 describes the m2m communication in computer networks. In Section 4, an ILP model for overlay network is presented. Section 5 contains similar model for joint m2m system, using the node-link notation. Two further sections extend the previous models with the replica location problem. Section 6 describes the overlay model of the replica location problem, and Section 7 refers to the joint model. Finally, the paper is concluded in Section 8.

## 2. Related Works

The idea of many-to-many communication in the networks is not a recent invention. The author in [6] predicted that teleconferences will be as popular as television. After many years, we know how true was this prediction. Extended view on m2m applications in background of multicast is presented in [7]. The authors define m2m traffic as a group of hosts, where each of them receives data from multiple senders while it also sends data to all of them. They also highlight that this communication paradigm may cause complex coordination and management challenges. The examples of m2m applications are, among others: multimedia conferencing, synchronizing resources, distributed parallel processing, shared document editing, distance learning or multiplayer games, to name a few. Moreover, the paper presents a brief comparison of delay tolerance and mentions that m2m applications characterize in a high delay intolerance.

In [8], the authors propose scheduling architecture for m2m traffic in switched HPC (High Performance Computing) networks. The paper also mentions other applications of m2m communications in data centers, for example process and data replication [9], dynamic load-balancing [10] or moving virtual machine resources between servers connected into a cloud [11]. In [12], the authors presents optimal and nearly optimal hot potato routing algorithms for many-to-many transmissions. In hot potato (deflection) routing, a packet cannot be buffered, and is therefore always moving until it reaches its destination. This scenario is mostly applicable in parallel computing applications.

Many-to-many communication is also extensively investigated in the area of radio networks. Overview on this topic is presented in [13]. The authors of [14] propose a Middleware for Many-to-many Communication (M2MC) system architecture for m2m applications in broadcast networks (both radio and wired). Because of broadcast orientation, M2MC do not require any resource consum-

ing routing protocols. The system architecture comprises of Message Ordering Protocol, Member Synchronization Protocol and protocols for processes to join and leave the groups.

Other applications of m2m communication exist in a field of online gaming [15]–[18]. All the players need to exchange with the others the current state of the game. In dynamic games delay tolerance is crucial, and online gaming protocols are designed to transfer small portions of data in often transmitted packets. When more servers are available, the game world is usually splitted into several zones and users are assigned to the server, taking under account a zone in which their avatar currently exists.

Mixed-integer Linear Programming (MILP) formulation for many-to-many traffic grooming in Wavelength-Division Multiplexing (WDM) networks is presented in [19] and [20]. The authors not only formulate MILP problems, but also present approximated heuristic algorithms. Both solutions are considered for non-splitting networks, where optical-electronic-optical conversion is used and in networks capable of splitting the signal in optical domain. In WDM networks, due to wide optical spectrum even broadband many-to-many multimedia streams may be aggregated (groomed) to use available bandwidth more efficiently.

Many-to-many transmission in telepresence appliance is presented in [21]. The authors compare two architectures, namely centralized and distributed. Moreover, the video transmission is encoded using Scalable Video Coding (SVC) [22]. In SVC, a stream consists of a base layer and several enhancement layers, that after merging with the base layer, improve a video quality. Every client receives as many layers as the link, that it is connected to the network, can handle at low delay. Finally, different approaches to the video exchange during videoconferences have been presented in [23]. The authors proposed an algorithm to build separate trees for different enhancement layers in SVC based transmission. They make a theoretical analysis to show optimality of the algorithm and prove it through extensive simulations.

In [24], the authors propose a flow control protocol based on cost-benefit approach. Practical realization of this protocol framework for many-to-many flow control in overlay networks is designed and tested both in extensive simulations and real-life experiments.

Overlay networking is a subject of interest in numerous publications. An extensive work on overlay networks can be found in [25]. The author provides a complete introduction to the topic, followed by architecture description, requirements, underlying topologies, and routing information. The work is also supplemented with a discussion about security and overlay networks applications.

Replica location problem has been addressed in previous publications [26], [27]. However, most of the work has been done in a relation to the Content Delivery Network and web-content servers [28] or transparent proxying [29]. In the topic of multimedia transmission, previous work concentrates mostly on placing Video on Demand (VoD)

servers or static multimedia replicas [30], however, in [31] the authors address anycast in a field of relaying node selection and Voice over Internet Protocol (VoIP) Session Border Controller (SBC) placement.

### 3. Many-to-Many Communication

As mentioned in the previous section, many-to-many communication is a paradigm of data exchange between group of hosts in a way that every group member gets information from the rest of hosts involved in the transmission. Basically, during the transmission every host in the group has the same set of information (i.e. all videoconference participants see video streams from other conference members). The overall set of m2m demands is known in advance and the problem consists of optimizing the establishment of the m2m flows to serve these demands. This abstract model was divided into two more specific problems for the communication in computer networks:

- **Overlay model.** In this model, the m2m flows are determined assuming a given set of network routes already established, i.e., the service layer is decoupled from the IP layer. This model is easier to deploy since there is no need of the network topology information and the traffic routing in the network layer;
- **Joint model.** In this model, the establishment of the m2m flows involves also the underlying network layers (e.g., IP layer, MPLS layer, optical layer, etc.). This model is harder to implement but allows optimizing network routes and m2m flows together in order to minimize bandwidth usage.

### 4. Overlay m2m Systems – Optimization Model

In this section, the ILP model of the offline m2m flows allocation in overlay system is presented. First, we introduce the main assumptions of an overlay system with m2m flows. A set of users (overlay nodes) indexed  $v = 1, 2, \dots, V$  that participate in the system is given, i.e., each user generates some stream with rate  $h_v$  (defined in bit/s) and receives the aggregated streams from other users. For instance in the context of teleconferencing system, the value  $h_v$  depends on the selected coding standard and resolution. A special compression ratio  $\alpha_v$  is defined for each user – the user receives the overall stream compressed according to this ratio. This assumptions also follows from real teleconferencing systems [32], [33]. In the considered system, servers  $s = 1, 2, \dots, S$  are rendezvous points. In a nutshell, each user sends its flow to one selected server. The server aggregates all received flows, and thus provides the stream to each user with the requested compression ratio. Each server  $s = 1, 2, \dots, S$  has a limited upload and download capacity ( $u_s$  and  $d_s$ , respectively). Another possible model – not

addressed here – is a case when servers exchange information with each other and the users receives the aggregated stream of all users from one selected server.

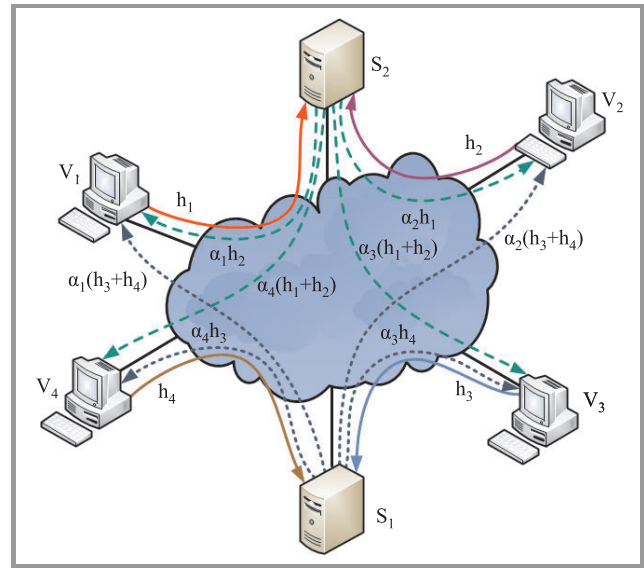


Fig. 1. Many-to-many transmission model in overlay network.

As an example, Fig. 1 shows the considered overlay model in a network with 4 clients (users) and 2 servers. Clients  $v_1$  and  $v_2$  are sending their streams to server  $s_2$  and clients  $v_3$  and  $v_4$  to  $s_1$ . Both upstream and downstream flows are presented and transmission volume is shown. For example client  $v_1$  transmits stream with volume  $h_1$  to server  $s_2$  and receives two streams compressed with requested compression ratio  $\alpha_1$ . The former comes from  $s_2$  and consists of stream  $h_2$  from client  $v_2$  (its own stream is not sent back), the latter comes from  $s_1$  and consists of streams  $h_3$  and  $h_4$  from corresponding clients  $v_3$  and  $v_4$ .

There are two sets of decision variables in the model. First,  $z_{vs}$  denotes the selection of server  $s$  for demand  $v$ . The second variable  $H_s$  is auxiliary and defines the flow of all users connected to server  $s$ . The objective is to minimize the overall streaming cost according to the allocation of users to servers. For each pair of overlay nodes (both users and/or servers) we are given constant  $\zeta_{vw}$  denoting the streaming cost of one capacity unit (i.e., Mbit/s) on an overlay link from node  $v$  to node  $w$ . The cost can be interpreted in many ways, e.g., as network delay (in ms), bandwidth consumption, number of Autonomous Systems (ASes) on the path, etc., or a weighted combination of them. To present the model notation as in [34] is used:

- **indices**
  - $v, w = 1, 2, \dots, V$  user (overlay nodes),
  - $s = 1, 2, \dots, S$  servers (overlay nodes);
- **constants**
  - $d_s$  download capacity (bit/s) of server  $s$ ,
  - $u_s$  upload capacity (bit/s) of server  $s$ ,
  - $\zeta_{vw}$  streaming cost on overlay link from node  $v$  to node  $w$ ,



- $h_v$  stream rate (bit/s) generated by node (client)  $v$ ,  
 $\alpha_v$  compression ratio of node (client)  $v$ ,  
 $N_s$  maximum number of users that  $s$  can serve;

– **variables**

$z_{vs} = 1$ , if user  $v$  is assigned to server  $s$  and 0 otherwise (binary),

$H_s$  flow aggregated at server  $s$  (continuous);

– **objective**

$$\min F = \sum_v \sum_s z_{vs} h_v \zeta_{vs} + \sum_v \sum_s \alpha_v (H_s - z_{vs} h_v) \zeta_{sv}, \quad (1)$$

– **subject to**

$$\sum_s z_{vs} = 1 \quad v = 1, 2, \dots, V, \quad (2)$$

$$H_s = \sum_v z_{vs} h_v \quad s = 1, 2, \dots, S, \quad (3)$$

$$H_s \leq d_s \quad s = 1, 2, \dots, S, \quad (4)$$

$$\sum_v \alpha_v (H_s - z_{vs} h_v) \leq u_s \quad s = 1, 2, \dots, S, \quad (5)$$

$$\sum_v z_{vs} \leq N_s \quad s = 1, 2, \dots, S. \quad (6)$$

The objective (1) is to minimize the streaming cost of transferring all m2m flows in the system. In more detail, function (1) comprises two elements. The first one (i.e.,  $\sum_v \sum_s z_{vs} h_v \zeta_{vs}$ ) denotes the cost of streaming the data from users to servers. The second part (i.e.,  $\sum_v \sum_s \alpha_v (H_s - z_{vs} h_v) \zeta_{sv}$ ) defines the cost of streaming the data in the opposite direction from each server to each user. Recall that for each user a special compression ratio  $\alpha_v$  is given. Moreover, if a particular server  $s$  is selected by user  $v$  (i.e.,  $z_{vs} = 1$ ), the flow of this server  $s$  is decreased by the flow of user  $v$ . Constraint (2) assures that for each user  $v$  exactly one server is selected. In (3), the aggregated flow entering each server  $s$  is defined as the sum of all users' flows assigned to  $s$ . In constraints (4) and (5) the download and upload capacity constraints for servers is defined. Each server uploads the aggregated stream with the defined compression ratio to each user. Therefore, similarly to obj. (1), the original flow of user  $v$  is not sent back to this node. Since the upload and download flows of users are constant, we do not formulate capacity constraint in the case of user nodes. Finally, constraint (6) bounds the number of users to be served by each server. This limit follows from real m2m systems (e.g., teleconferencing systems) [33]. The presented model in (1)–(6) is strongly NP-hard problem since it is equivalent to the Multidimensional Knapsack Problem [35].

A special case of the overlay model presented in (1)–(6) is a scenario where only one server ( $S = 1$ ) is applied to provide the m2m transmissions in the network. Notice that in this case, this model becomes an analytical model, since there are no variables as all users are assigned to the same

server (variable  $z_{vS}$ ). As a consequence, the aggregated flow at the server is constant and given by

$$H_1 = \sum_v h_v. \quad (7)$$

The cost of one server scenario is as follows

$$F = \sum_v h_v \zeta_{v1} + \sum_v \alpha_v (H_1 - h_v) \zeta_{1v}. \quad (8)$$

Notice that Eq. (8) can be used as a reference cost when evaluating multi servers scenarios.

## 5. Joint m2m Systems – Optimization Model

Now, a joint model of m2m flows is introduced. The main assumptions are analogous to the overlay model. The key difference is that with the joint model, network routes between users and servers can be optimized. The authors will formulate joint system ILP model using node-link notation [34].

The considered network is modeled as a directed graph consisting of nodes and links. Nodes are divided into two subsets: nodes hosting servers (indexed by  $s = 1, 2, \dots, S$ ) and all other nodes (indexed by  $v = 1, 2, \dots, V$ ). Users can be connected only to nodes  $v = 1, 2, \dots, V$ . We assume that server nodes are connected to the graph by a bridge (cut-edge), i.e., removal of the edge disconnects the server node from the rest of the graph. This follows from the fact that server nodes cannot be used as a transit node for forwarding data that does not originate or terminate at the server node. In contrast, nodes  $v = 1, 2, \dots, V$  can be used as transit nodes. Links are denoted using index  $e = 1, 2, \dots, E$ . Recall that in the case of overlay systems, the notion of a node was used to denote a user. To simplify the notation, in this section we apply the notion of a demand  $d = 1, 2, \dots, D$  to denote all flows in the system between users and servers. Let  $o(d)$  and  $t(d)$  denote the origin and destination node of each demand, respectively. There are two types of demands: upstream and downstream. The former one denotes the flow from a user to one of the servers, thus for each upstream demand  $d$ ,  $o(d)$  denotes the user node. The upstream demand is an anycast demand, since one of the end nodes is to be selected among many possible nodes. The volume of this demand is constant and given by  $h_d$ . Since an upstream demand is defined by the user node  $o(d)$  we can write that that  $h_d = h_{o(d)}$ , i.e., volume of upstream demand  $d$  is equivalent to the bitrate generated by client located at node  $o(d)$ .

For each user (node  $v$ ) there are  $S$  downstream demands to transmit the aggregated flow from each server to the user node. The destination node  $t(d)$  of each downstream demand is always located in a user node. Consequently, candidate paths for each demand connect the server node and the user node. Downstream demands are unicast since both end nodes are defined a priori. Moreover, with every down-stream demand we introduce the index of associated

up-stream demand  $\tau(d)$ . Both associated demands  $d$  and  $\tau(d)$  of the same request must connect the same pair of nodes: the client node and the selected replica node. However, the main novelty is that the volume of downstream demands is a variable and depends on the allocation of users to servers. In more detail, the volume of downstream demand  $d$  is defined as  $\alpha_{t(d)}(H_{o(d)} - z_{t(d)o(d)}h_{\tau(d)})$ .

Let  $a_{ev}$  and  $b_{ev}$  denote the binary constants that define the dependency between adjacent links and nodes. More precisely,  $a_{ev}$  is 1, when link  $e$  originates at node  $v$  and 0 otherwise. Similarly,  $b_{ev}$  is 1, if link  $e$  terminates at node  $v$  and 0 otherwise.

– **indices**

- $v = 1, 2, \dots, V$  network client nodes,  
 $s = 1, 2, \dots, S$  network server nodes,  
 $d = 1, 2, \dots, D$  demands (upstream from user to server and downstream from server to user),  
 $e = 1, 2, \dots, E$  network links;

– **constants**

- $h_d$  volume (requested bit-rate) of upstream demand  $d$ ,  
 $\zeta_e$  streaming cost on link  $e$ ,  
 $c_e$  capacity of link  $e$ ,  
 $ds(d) = 1$ , if  $d$  is a downstream demand, 0 otherwise,  
 $us(d) = 1$ , if  $d$  is an upstream demand, 0 otherwise,  
 $o(d)$  origin (source) node of demand  $d$ , for an upstream demand  $o(d)$  denotes the user node, for a downstream demand  $o(d)$  denotes the server node,  
 $t(d)$  destination node of demand  $d$ , in the case of a upstream demand  $t(d)$  denotes the server, while in the case of downstream demand  $t(d)$  is the user node,  
 $\tau(d)$  index of a demand associated with demand  $d$ . If  $d$  is a downstream demand, then  $\tau(d)$  must be an upstream connection and vice versa,  
 $M$  large number,  
 $N_s$  maximum number of users that  $s$  can serve,  
 $a_{ev} = 1$ , if link  $e$  originates at node  $v$ , 0 otherwise,  
 $b_{ev} = 1$ , if link  $e$  terminates at node  $v$ , 0 otherwise;

– **variables**

- $z_{vs} = 1$ , if user  $v$  is assigned to server  $s$ , 0 otherwise (binary),  
 $H_s$  flow aggregated at server  $s$  (continuous),  
 $x_{ed}$  flow of demand  $d$  on link  $e$  (continuous),  
 $u_{ed} = 1$ , if demand  $d$  uses link  $e$ , 0 otherwise (binary);

– **objective**

$$\min F = \sum_d \sum_e x_{ed} \zeta_e \quad (9)$$

– **subject to**

$$\begin{aligned} \sum_e a_{es} x_{ed} - \sum_e b_{es} x_{ed} &= \alpha_{t(d)} (H_{o(d)} - z_{t(d)o(d)} h_{\tau(d)}) \\ d &= 1, 2, \dots, D \quad ds(d) = 1 \\ s &= 1, 2, \dots, S \quad o(d) = s \end{aligned} \quad (10)$$

$$\begin{aligned} \sum_e a_{ev} x_{ed} - \sum_e b_{ev} x_{ed} &= -\alpha_{t(d)} (H_{o(d)} - z_{t(d)o(d)} h_{\tau(d)}) \\ \text{if } v &= t(d) \quad d = 1, 2, \dots, D \\ ds(d) &= 1 \quad v = 1, 2, \dots, V \end{aligned} \quad (11)$$

$$\begin{aligned} \sum_e a_{ev} x_{ed} - \sum_e b_{ev} x_{ed} &= 0 \\ \text{if } v &\neq t(d) \quad d = 1, 2, \dots, D \\ ds(d) &= 1 \quad v = 1, 2, \dots, V \end{aligned} \quad (12)$$

$$\begin{aligned} \sum_e a_{ev} x_{ed} - \sum_e b_{ev} x_{ed} &= h_d \\ \text{if } v &= o(d) \quad d = 1, 2, \dots, D \\ us(d) &= 1 \quad v = 1, 2, \dots, V \end{aligned} \quad (13)$$

$$\begin{aligned} \sum_e a_{es} x_{ed} - \sum_e b_{es} x_{ed} &= -h_d z_{o(d)s} \\ d &= 1, 2, \dots, D \quad us(d) = 1 \\ s &= 1, 2, \dots, S \quad t(d) = s \end{aligned} \quad (14)$$

$$\begin{aligned} \sum_e a_{ev} x_{ed} - \sum_e b_{ev} x_{ed} &= 0 \\ \text{if } v &\neq o(d) \quad d = 1, 2, \dots, D \\ us(d) &= 1 \quad v = 1, 2, \dots, V \end{aligned} \quad (15)$$

$$\begin{aligned} \sum_e a_{es} u_{ed} - \sum_e b_{es} u_{ed} &= 1 \\ d &= 1, 2, \dots, D \quad ds(d) = 1 \\ s &= 1, 2, \dots, S \quad o(d) = s \end{aligned} \quad (16)$$

$$\begin{aligned} \sum_e a_{ev} u_{ed} - \sum_e b_{ev} u_{ed} &= -1 \\ \text{if } v &= t(d) \quad d = 1, 2, \dots, D \\ ds(d) &= 1 \quad v = 1, 2, \dots, V \end{aligned} \quad (17)$$

$$\begin{aligned} \sum_e a_{ev} u_{ed} - \sum_e b_{ev} u_{ed} &= 0 \\ \text{if } v &\neq t(d) \quad d = 1, 2, \dots, D \\ ds(d) &= 1 \quad v = 1, 2, \dots, V \end{aligned} \quad (18)$$

$$\begin{aligned} \sum_e a_{ev} u_{ed} - \sum_e b_{ev} u_{ed} &= 1 \\ \text{if } v &= o(d) \quad d = 1, 2, \dots, D \\ us(d) &= 1 \quad v = 1, 2, \dots, V \end{aligned} \quad (19)$$

$$\begin{aligned} \sum_e a_{es} u_{ed} - \sum_e b_{es} u_{ed} &= -z_{o(d)s} \\ d &= 1, 2, \dots, D \quad us(d) = 1 \\ s &= 1, 2, \dots, S \quad t(d) = s \end{aligned} \quad (20)$$

$$\begin{aligned} \sum_e a_{ev} u_{ed} - \sum_e b_{ev} u_{ed} &= 0 \\ \text{if } v &\neq o(d) \quad d = 1, 2, \dots, D \\ us(d) &= 1 \quad v = 1, 2, \dots, V \end{aligned} \quad (21)$$

$$x_{ed} \leq M u_{ed} \quad (22)$$

$$d = 1, 2, \dots, D \quad e = 1, 2, \dots, E$$

$$H_s = \sum_{d:up(d)=1} z_{o(d)s} h_d \quad (23)$$

$$s = 1, 2, \dots, S$$

$$\sum_s z_{o(d)s} = 1 \quad (24)$$

$$d = 1, 2, \dots, D \quad up(d) = 1$$

$$\sum_d x_{ed} \leq c_e \quad (25)$$

$$e = 1, 2, \dots, E$$

$$\sum_{d:up(d)=1} z_{o(d)s} \leq N_s \quad (26)$$

$$s = 1, 2, \dots, S$$

The objective function (9) minimizes the cost of all network flows. Constraints (10)–(12) define the flow conservation laws for downstream demands. Recall that in our model the downstream demand is a unicast demand from a server to a user. Therefore, as a source node only server nodes are considered, see constraint (10). The right-hand side of (10) denotes the flow of downstream demand  $d$ , which is the flow received by the user from each server. The compression ratio is applied and the original stream generated by the node is not sent back. Constraint (11) relates to the destination node of the demand, i.e., user node. Finally, constraint (12) is formulated for other so called transit nodes. Furthermore, in (13)–(15) the flow conservation of upstream demands is defined, which are anycast. In more detail, (13) denotes the flow conservation for the user node. Constraint (14) meets the guarantee that one of the servers (defined by the value of  $z_{vs}$  variable) is selected as the destination node. Constraint (15) defines the flow conservation law for remaining transit nodes. Notice that we assume that server nodes can be used as transit nodes to forward traffic of demands not terminated or originated at particular server node. Since we assume single path routing, constraints (16)–(18) and (19)–(21) denote the flow conservation constraints for corresponding binary flow variables  $u_{ed}$ . Both flow variables are bound through using constraint (22).

Constraint (23) – similarly to (3) – defines the flow of server  $s$  according to assignment of users to servers. Constraint (24) defines variable  $z_{vs}$ . Constraint (25) is the link capacity. Finally, (26) limits the number of clients served by each server. Model (9)–(26) is NP-complete since it is equivalent to the single path allocation problem [34].

Notice that in order to obtain bifurcated version of the link-node model variables  $u_{ed}$  and constraints (16)–(22) must be removed from the above model.

## 6. Overlay System Replica Location Problem – Optimization Model

In this section, the ILP model of replica location problem in overlay m2m systems is introduced, that belongs to the

group of LFA (Location and Flow Allocation) problems. In the previous two models, the authors assumed that the location of the replica servers is fixed. Here, the problem is to choose  $R$  replicas among  $V$  potential sites ( $R < V$ ) taking under consideration demands in the network. In comparison to the equivalent problem (1)–(6), where location of the replicas is known, we do not distinguish client and server nodes. We are given  $v, w = 1, 2, \dots, V$  nodes from which  $R$  replica nodes will be selected. Therefore, binary variable  $z_w$  is used, which is 1 when  $w$  hosts a replica server and 0 otherwise. The problem of locating replicas in the network is NP-hard, since it is equivalent to the facility location problem [28], [36]:

- **indices**  
 $v, w = 1, 2, \dots, V$  overlay nodes;
- **constants**  
 $d_v$  download capacity (bit/s) of node  $v$ ,  
 $u_v$  upload capacity (bit/s) of node  $v$ ,  
 $\zeta_{vw}$  streaming cost on overlay link from node  $v$  to node  $w$ ,  
 $h_v$  streaming rate (bit/s) generated by node  $v$ ,  
 $\alpha_v$  compression ratio of node  $v$ ,  
 $N_v$  maximum number of users that  $v$  can serve,  
 $R$  number of replica servers,
- **variables**  
 $z_{vw} = 1$ , if node  $v$  is assigned to replica node  $w$ ,  
 0 otherwise (binary),  
 $z_w = 1$ , if node  $w$  is selected to host a replica server, 0 otherwise (binary),  
 $H_w$  flow aggregated at replica node  $w$  (continuous);

- **objective**

$$\min F = \sum_v \sum_w z_{vw} h_v \zeta_{vw} + \sum_v \sum_w \alpha_v (H_w - z_{vw} h_v) \zeta_{vw} \quad (27)$$

- **subject to**

$$\sum_w z_{vw} = 1 \quad v = 1, 2, \dots, V \quad (28)$$

$$H_w = \sum_{v:v \neq w} z_{vw} h_v = 1 \quad w = 1, 2, \dots, V \quad (29)$$

$$H_w < d_w \quad w = 1, 2, \dots, V \quad (30)$$

$$\sum_v \alpha_v (H_w - z_{vw} h_v) \leq u_w \quad w = 1, 2, \dots, V \quad (31)$$

$$\sum_v z_{vw} \leq N_w \quad w = 1, 2, \dots, V \quad (32)$$

$$\sum_w z_w \leq R \quad (33)$$

$$z_{vw} \leq z_w \quad v, w = 1, 2, \dots, V \quad (34)$$

The objective (27) is to minimize the streaming cost of transferring all m2m flows in the system. First component denotes the cost of streaming the data from users to servers. The second part defines the cost of streaming the data in the opposite direction. Constraint (28) assures that each user is assigned to exactly one replica node. The flow ag-

gregated at each replica is defined in (29). Constraints (30) and (31) are defining download and upload capacity boundaries. The number of users to be served by each server is constrained in (32). Constraint (33) guarantees that  $R$  nodes are selected to host replica servers. Finally, (34) binds variables  $z_{vw}$  and  $z_w$ , i.e., node  $w$  can be selected as the replica node for any user  $v$ , only if node  $w$  is assigned with a replica node ( $z_w = 1$ ).

## 7. Joint System Replica Location Problem – Optimization Model

Analogously to the problem presented in the previous section, the base problem with replica servers selection is extended. Due to the simplicity of the model representation node-link notation is used.

### – indices

- $v, w = 1, 2, \dots, V$  network nodes,  
 $d = 1, 2, \dots, D$  demands (upstream from user to server and downstream from server to user),  
 $e = 1, 2, \dots, E$  network links;

### – constants

- $h_d$  volume (requested bit-rate) of upstream demand  $d$ ,  
 $\zeta_e$  streaming cost on link  $e$ ,  
 $c_e$  capacity of link  $e$ ,  
 $ds(d) = 1$ , if  $d$  is a downstream demand, 0 otherwise,  
 $us(d) = 1$ , if  $d$  is an upstream demand, 0 otherwise,  
 $a_{ev} = 1$ , if link  $e$  originates at node  $v$ , 0 otherwise,  
 $b_{ev} = 1$ , if link  $e$  terminates at node  $v$ , 0 otherwise,  
 $\alpha_v$  compression ratio of node  $v$ ,  
 $N_v$  maximum number of users that  $v$  can serve,  
 $o(d)$  origin (source) node of demand  $d$ , for an upstream demand  $o(d)$  denotes the user node, for a downstream demand  $o(d)$  denotes the server node,  
 $t(d)$  destination node of demand  $d$ , in the case of a upstream demand  $t(d)$  denotes the server, while in the case of downstream demand  $t(d)$  is the user node,  
 $\tau(d)$  index of a demand associated with demand  $d$ ; if  $d$  is a downstream demand, then  $\tau(d)$  must be an upstream connection and vice versa,  
 $R$  number of replica servers,  
 $M$  large number;

### – variables

- $z_{vw} = 1$ , if node  $v$  is assigned to replica node  $w$ , 0 otherwise (binary),  
 $z_w = 1$ , if node  $w$  is selected to host a replica server, 0 otherwise (binary),

- $H_w$  flow aggregated at replica node  $w$  (continuous),  
 $x_{ed}$  flow of demand  $d$  on link  $e$  (continuous),  
 $u_{ed} = 1$ , if demand  $d$  uses link  $e$ , 0 otherwise (binary);

### – objective

$$\min F = \sum_d \sum_e x_{ed} \zeta_e \quad (35)$$

### – subject to

$$\begin{aligned} \sum_e a_{ev} x_{ed} - \sum_e b_{ev} x_{ed} &= \alpha_{t(d)} (H_{o(d)} - z_{t(d)o(d)} h_{\tau(d)}) \\ d &= 1, 2, \dots, D \quad ds(d) = 1 \\ v &= 1, 2, \dots, V \quad v = o(d) \end{aligned} \quad (36)$$

$$\begin{aligned} \sum_e a_{ev} x_{ed} - \sum_e b_{ev} x_{ed} &= -\alpha_{t(d)} (H_{o(d)} - z_{t(d)o(d)} h_{\tau(d)}) \\ d &= 1, 2, \dots, D \quad ds(d) = 1 \\ v &= 1, 2, \dots, V \quad v = t(d) \end{aligned} \quad (37)$$

$$\begin{aligned} \sum_e a_{ev} x_{ed} - \sum_e b_{ev} x_{ed} &= 0 \\ d &= 1, 2, \dots, D \quad ds(d) = 1 \\ v &= 1, 2, \dots, V \quad v \neq t(d) \quad v \neq o(d) \end{aligned} \quad (38)$$

$$\begin{aligned} \sum_e a_{ev} x_{ed} - \sum_e b_{ev} x_{ed} &= h_d (1 - z_v) \\ d &= 1, 2, \dots, D \quad us(d) = 1 \\ v &= 1, 2, \dots, V \quad v = o(d) \end{aligned} \quad (39)$$

$$\begin{aligned} \sum_e a_{ev} x_{ed} - \sum_e b_{ev} x_{ed} &= -h_d z_{o(d)v} \\ d &= 1, 2, \dots, D \quad us(d) = 1 \\ v &= 1, 2, \dots, V \quad v \neq o(d) \end{aligned} \quad (40)$$

$$\begin{aligned} \sum_e a_{ev} u_{ed} - \sum_e b_{ev} u_{ed} &= z_{o(d)} \\ d &= 1, 2, \dots, D \quad ds(d) = 1 \\ v &= 1, 2, \dots, V \quad v = o(d) \end{aligned} \quad (41)$$

$$\begin{aligned} \sum_e a_{ev} u_{ed} - \sum_e b_{ev} u_{ed} &= -z_{o(d)} \\ d &= 1, 2, \dots, D \quad ds(d) = 1 \\ v &= 1, 2, \dots, V \quad v = t(d) \end{aligned} \quad (42)$$

$$\begin{aligned} \sum_e a_{ev} u_{ed} - \sum_e b_{ev} u_{ed} &= 0 \\ d &= 1, 2, \dots, D \quad ds(d) = 1 \\ v &= 1, 2, \dots, V \quad v \neq t(d) \quad v \neq o(d) \end{aligned} \quad (43)$$

$$\begin{aligned} \sum_e a_{ev} u_{ed} - \sum_e b_{ev} u_{ed} &= 1 - z_v \\ d &= 1, 2, \dots, D \quad us(d) = 1 \\ v &= 1, 2, \dots, V \quad v = o(d) \end{aligned} \quad (44)$$

$$\begin{aligned} \sum_e a_{ev} u_{ed} - \sum_e b_{ev} u_{ed} &= -z_{o(d)v} \\ d &= 1, 2, \dots, D \quad us(d) = 1 \\ v &= 1, 2, \dots, V \quad v \neq o(d) \end{aligned} \quad (45)$$



$$x_{ed} \leq Mu_{ed} \quad (46)$$

$$d = 1, 2, \dots, D \quad e = 1, 2, \dots, E$$

$$H_v = \sum_{d:up(d)=1} z_{o(d)v} h_d \quad (47)$$

$$v = 1, 2, \dots, V$$

$$\sum_v z_{o(d)v} = 1 \quad (48)$$

$$d = 1, 2, \dots, D \quad up(d) = 1$$

$$\sum_d x_{ed} \leq c_e \quad (49)$$

$$e = 1, 2, \dots, E$$

$$\sum_{d:up(d)=1} z_{o(d)v} \leq N_v \quad (50)$$

$$v = 1, 2, \dots, V$$

$$\sum_v z_v \leq R \quad (51)$$

$$z_{vw} \leq z_w \quad v, w = 1, 2, \dots, V \quad (52)$$

The objective function (35) minimizes the cost of all network flows. Constraints (36)–(38) define the flow conservation laws for downstream demands. In detail, (36) presents the case, when  $v$  is a source of demand  $d$ , so it is a potential replica. If so, right hand side of (36) denotes the flow of demand  $d$ , otherwise equals 0. Constraint (37) is defined for the destination node of demand  $d$  ( $v = t(d)$ ), hence the left-hand denotes the flow that enters to the client node  $v$ . We assume that the replica node can be located only in the nodes that are not the client nodes. Finally in (38)  $v$  represents an intermediate node and flow balance equals 0. In analogous way we formulate the flow conservation law for upstream demands (39)–(40). Constraint (40) represents two cases - when  $v$  is a replica node or an intermediate node. In the former, variable  $z_{o(d)v}$  is set to 1 and right-hand side of (40) denotes flow of demand  $d$  incoming to replica  $v$ . In the latter,  $z_{o(d)v}$  is set to 0 and right-hand side equals 0. In this model a single path routing is considered, thus constraints (41)–(43) and (44)–(45) denote the flow conservation constraints for corresponding binary flow variables  $u_{ed}$ . This variable is bound with continuous flow variable  $x_{ed}$  in constraint (46). Constraints (47)–(50) are analogous to the node-link problem model with known server location. Constraints (51)–(52) are equivalent of (33)–(34) in the overlay model.

## 8. Concluding Remarks

In this paper, ILP optimization models of computer networks with many-to-many multimedia flows was formulated. The authors addressed two problems of replica server settlement – with known replica location, and with optimized replica location selection. According to many recent developments in computer networks, m2m transmissions have been gaining much popularity in different areas. The models presented can be easily adapted for other traffic patterns and applications. Generic ILP models of m2m flows optimization in overlay model and joint mode as-l

suming combined optimization of overlay and underlying networks (e.g., IP layer, MPLS layer, optical layer, etc.) was proposed. The models assume that special servers (rendezvous point) collect flows of individual clients and sent them back to users using some compression. In future work, the authors plan to implement the models in ILP solvers as well as to develop some heuristic algorithms to obtain numerical results, and to formulate models of m2m systems using multicasting for effective transmission.

## Acknowledgements

The work was supported by the Polish National Science Centre under the grant N N519 650440. The work on this paper was held when Damian Bulira was on the PhD internship in Advanced Broadband Communication Center (CCABA), Universitat Politecnica de Catalunya, Barcelona, Spain.

## References

- [1] K. C. Almeroth and M. H. Ammar, “The use of multicast delivery to provide a scalable and interactive Video-on-Demand service”, *IEEE J. Sel. Areas Telecommun.*, vol. 14, no. 6, 6 1996.
- [2] D. Mills *et al.*, “Network Time Protocol Version 4: Protocol and Algorithms Specification”, RFC 5905, June 2010.
- [3] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi, “A survey on content-oriented networking for efficient content delivery”, *IEEE Commun. Mag.*, vol. 49, iss. 3, pp. 121–127, 2011
- [4] Cisco Visual Networking Index: Forecast and Methodology 2011–2016, 2012.
- [5] K. Walkowiak, D. Bulira, and D. Careglio, “ILP modeling of many-to-many transmissions in computer networks”, in *Proc. 17th Polish Telegraf. Symp. 2012*, Zakopane, Poland, 2012, pp. 123–128.
- [6] C. H. Stevens, “Many-to-Many Communication”, Tech. rep. MIT/Sloan/TR-175 Sloan School of Management, Massachusetts Institute of Technology, 1981.
- [7] B. Quinn and K. Almeroth, “IP Multicast Applications: Challenges and Solutions”, RFC 3170, September 2001.
- [8] S. Banerjee *et al.*, “Contention-Free Many-to-Many Communication Scheduling for High Performance Clusters”, *Distributed Computing and Internet Technology*, LNCS, vol. 6536, pp. 150–161. Heidelberg: Springer, 2011.
- [9] N. Touheed *et al.*, “A comparison of dynamic load-balancing algorithms for a parallel adaptive flow solver”, *Parallel Comput.*, vol. 26, no. 12, pp. 1535–1554, 2000.
- [10] K. Sinha *et al.*, “Efficient load balancing on a cluster for large scale online video surveillance”, in *Proc. 10th Int. Conf. Distrib. Comput. Netw. ICDCN 2009*, Hyderabad, India, 2009, V. Garg *et al.*, Eds. LNCS, vol. 5408, pp. 450–455, Heidelberg: Springer, 2009.
- [11] T. C. Wilcox Jr., “Dynamic load balancing of virtual machines hosted on Xen”, Master thesis, Dept. of Computer Science, Brigham Young University, April 2009.
- [12] A. Borodin, Y. Rabani, and B. Schieber, “Deterministic many-to-many hot potato routing”, *IEEE Trans. Paral. Distrib. Sys.*, vol. 8, no. 6, pp. 587–596, 1997.
- [13] B. S. Chlebus, D. R. Kowalski, T. Radzik, “On many-to-many communication in packet radio networks”, in *Proc. 10th Int. Conf. Princip. Distrib. Sys. OPODIS 2006*, Bordeaux, France, 2006, A. A. Shvartsman, Ed. LNCS, vol. 4305, pp. 260–274, Heidelberg: Springer, 2006.
- [14] C. K. Bhavanasi, “M2MC: Middleware for many to many communication over broadcast networks”, in *Proc. 1st Int. Conf. Commun. Sys. Softw. Middlew. COMSWARE 2006*, New Delhi, India, 2006.



[15] F. Glinka, A. Ploss, J. Müller-Iken, and S. Gorlatch, "RTF: A real-time framework for developing scalable multiplayer online games", in *Proc. 6th ACM Worksh. Netw. Sys. Support Games NETGAMES 2007*, Melbourne, Australia, 2007, pp. 81–86.

[16] A. Ploss, S. Wichmann, F. Glinka, and S. Gorlatch, "From a Single-to Multi-Server Online Game: A Quake 3 Case Study Using RTF", in *Proc. ACM Int. Conf. Adv. Comp. Entertain. Technol. ACE 2008*, Yokohama, Japan, 2008.

[17] P. Prata, E. Pinho, and E. Aires, "Database and state replication in multiplayer online games", in *Proc. 24th IEEE Int. Conf. Adv. Infor. Netw. Appl. Worksh. WAINA 2010*, Perth, Australia, 2010.

[18] M. Assiotis and V. Tzanov, "A distributed architecture for MMORPG", in *Proc. 5th Worksh. Netw. Sys. Sup. Games NETGAMES 2006*, Singapore, 2006.

[19] M. A. Saleh and A. E. Kamal, "Many-to-many traffic grooming in WDM networks", *IEEE/OSA J. Opt. Commun. Netw.*, vol. 1, iss. 5, pp. 376–391, 2009.

[20] M. A. Saleh and A. E. Kamal, "Approximation algorithms for many-to-many traffic grooming in optical WDM networks", *IEEE/ACM Trans. Netw.*, vol. 20, iss. 5, pp. 1527–1540, 2012.

[21] T. A. Le and H. Nguyen, "Centralized and distributed architectures of scalable video conferencing services", in *Proc. 2nd Int. Conf. Ubiquitous Future Netw. ICUFN 2010*, Jeju Island, Korea, 2010, pp. 394–399.

[22] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC Standard", *IEEE Trans. Circuit Sys. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, 2007.

[23] M. Ponc *et al.*, "Optimizing multi-rate peer-to-peer video conferencing applications", *IEEE Trans. Multim.*, vol. 13, no. 5, pp. 856–868, 2011.

[24] Y. Amir *et al.*, "Flow control for many-to-many multicast: a cost-benefit approach", in *Proc. IEEE Conf. Open Architect. Netw. Program. OPENARCH 2001*, Anchorage, Alaska, USA, 2001.

[25] S. Tarkoma, *Overlay Networks: Toward Information Networking*. Auerbach Publications, 2010.

[26] M. Rabinovich, "Issues in web content replication", *Data Engin. Bull.*, vol. 21, no. 4, pp. 21–29, 1998.

[27] T. Loukopoulos, I. Ahmad, and D. Papadias, "An overview of data replication on the Internet", in *Proc. 6th Int. Sym. Parall. Architect., Algorithms Netw. I-SPAN'02*, Metro Manila, Philippines, 2002.

[28] L. Qiu, "On the placement of web server replicas", in *Proc. 20th Ann. Joint Conf. IEEE Comp. Commun. Societ. INFOCOM 2001*, Anchorage, AK, USA, 2001, vol. 3, pp. 1587–1596.

[29] Y. Shavitt, "Proxy location problems and their generalizations", in *Proc. 23rd Int. Conf. Distrib. Comput. Sys. Worksh.*, Providence, RI, USA, 2003.

[30] Y. Tu, J. Yan, and S. Prabhakar, "Quality-aware replication of multimedia data", in *Proc. 16th Int. Conf. Datab. Expert Sys. Appl. DEXA '05*, Copenhagen, Denmark, 2005, K. V. Andersen, J. Debenham, and R. Wagner, Eds. LNCS 3588, pp. 240–249. Berlin Heidelberg: Springer, 2005.

[31] R. Cohen and G. Nakibly, "A traffic engineering approach for placement and selection of network services", *IEEE/ACM Trans. Netw.(TON)*, vol. 17, no. 2, pp. 487–500, 2009.

[32] D. Maldow, "Videoconferencing Infrastructure: A Primer", 2012 [Online]. Available: [http://www.telepresenceoptions.com/2012/07/videoconferencing\\_infrastructu/](http://www.telepresenceoptions.com/2012/07/videoconferencing_infrastructu/)

[33] Polycom RealPresence Platform: Scalable Infrastructure for Distributed Video, Polycom Whitepaper, 2011.

[34] M. Pioro and D. Medhi, *Routing, Flow and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann, 2004.

[35] J. Puchinger, G. Raidl, and U. Pferschy, "The multidimensional Knapsack problem: structure and algorithms", *INFORMS J. Comput.*, vol. 22, no. 2, pp. 250–265, Spring 2010.

[36] "A compendium of NP optimization problems", P. Crescenzi, and V. Kann, Eds., 2005 [Online]. Available: <http://www.nada.kth.se/~viggo/wwwcompendium/>



**Krzysztof Walkowiak** received his Ph.D. and D.Sc. (habilitation) degrees in Computer Science from the Wrocław University of Technology, Poland, in 2000 and 2008, respectively. Currently, he is an Associate Professor at the Department of Systems and Computer Networks, Faculty of Electronics, Wrocław University of Technol-

ogy. His research interest is mainly focused on optimization of content-oriented networks; network survivability; elastic optical networks; optimization of distributed computing systems (cloud computing, grids); application of soft-optimization techniques for design of computer networks. He was involved in many research projects related to optimization of computer networks. Moreover, he was a consultant for projects for large companies including TPSA, PZU, PKO BP, Energia Pro, Ernst & Young. He has published more than 160 scientific papers. He serves as a reviewer for many international journals including: IEEE Communications Magazine, IEEE/ACM Transactions on Networking, IEEE Journal on Selected Areas in Communications, Computer Communication, Computational Optimization and Applications. He is/was actively involved in many international conferences. Prof. Walkowiak is a member of IEEE and ComSoc.

E-mail: [krzysztof.walkowiak@pwr.wroc.pl](mailto:krzysztof.walkowiak@pwr.wroc.pl)  
 Faculty of Electronics  
 Department of Systems and Computer Networks  
 Wrocław University of Technology  
 Wybrzeże Wyspiańskiego st 27  
 50-370 Wrocław, Poland



**Damian Bulira** received his M.Sc. degree in Computer Science from Wrocław University of Technology, Poland, in 2011. Since that time he is a Ph.D. student at the Department of Systems and Computer Networks, Faculty of Electronics, Wrocław University of Technology. Currently, he is doing a Ph.D. internship at the Broad-

band Communication Systems research group, Department of Computer Architecture, Universitat Politècnica de Catalunya, Barcelona, Spain. His research interests include network optimization, many-to-many networking, multimedia systems optimization, to name a few. He is a IEEE Student member and IEEE Communication Society member. E-mail: [damian.bulira@pwr.wroc.pl](mailto:damian.bulira@pwr.wroc.pl)  
 Faculty of Electronics  
 Department of Systems and Computer Networks  
 Wrocław University of Technology  
 Wybrzeże Wyspiańskiego st 27  
 50-370 Wrocław, Poland



**Davide Careglio** received the M.Sc. and Ph.D. degrees in Telecommunications Engineering both from Universitat Politecnica de Catalunya (UPC), Barcelona, Spain, in 2000 and 2005, respectively, and the Laurea degree in Electrical Engineering from Politecnico di Torino, Turin, Italy, in 2001. He is currently an Associate

Professor in the Department of Computer Architecture at UPC. His research interests include algorithm and protocol design in communication networks.

E-mail: [careglio@ac.upc.edu](mailto:careglio@ac.upc.edu)

Advanced Broadband Communication  
Center (CCABA)

Universitat Politecnica de Catalunya

Jordi Girona 1-3

D6-103

08034 Barcelona, Spain