ILP Modeling of Many-to-Many Transmissions in Computer Networks

Krzysztof Walkowiak*, Damian Bulira* and Davide Careglio†
*Department of Systems and Computer Networks
Wroclaw University of Technology, Wrocław, Poland
†Advanced Broadband Communication Center (CCABA)
Universitat Politecnica de Catalunya, Barcelona, Spain
Email: {krzysztof.walkowiak,damian.bulira}@pwr.wroc.pl, careglio@ac.upc.edu

***Abstract*—Information flow in nowadays networks spreads in the exponential way. This fact is caused by the intensive changes in the Internet communication paradigms. Transmission in the Internet evolved from the simple one-to-one communication to the more sophisticated data exchange methods. In this paper, the authors concentrate on many-to-many (m2m) communication, that is mainly driven by growing popularity of on-line conferences and telepresence applications. We propose several ILP models concerning different videoconferencing schemes. Moreover, proposed models are based both on overlay and joint scenarios. We formulate an overlay model where m2m flows are optimally established on top of a given set of network routes and a joint model where the network routes and the m2m flows are jointly optimized. Each model is being followed by a comprehensive description and is based on real teleconference system.**

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

I. Introduction

With a constant growth of the network traffic, new communication paradigms appear. The only way of communication flow in Global Network in its first phase of operation was one-to-one transmission. An example of this transmission is fetching a website from a server or simple one-to-one Voice over IP call. 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 IPTV 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. In this type of transmissions all the hosts exchange the information with every other host in the m2m group. The examples of such traffic are: video and teleconferencing, distance learning, multiplayer on-line gaming, distributed computing, etc. We 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 percent over the forecast period.

The main contributions of this paper are ILP models for many-to-many transmission in computer networks where rendezvous points are used. We propose overlay and joint models assuming combined optimization of overlay and underlying networks. The models support video conference applications, but can be easly redefined for other type of m2m traffic.

The remainder of this paper is organized as follows. Section II provides related works study on many-to-many communication. Section III describes the m2m communication in computer networks. In section IV, an ILP model for overlay network is presented. Section V contains similar model for joint m2m system, moreover, two model notations are provided link-path and node-link. Finally, the paper is concluded in section VI.

II. Related works

The idea of many-to-many communication in the networks is not the recent invention. The author in [5] 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 [6]. The authors define m2m traffic as a group of hosts, where each of them receive 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

The work was supported in part by the Polish National Science Center (NCN) under the grant which is being realized in years 2011-2014.

Damian Bulira is also with Advanced Broadband Communication Center (CCABA), Universitat Politecnica de Catalunya, Barcelona, Spain.

tolerance and mention that m2m applications characterize in a high delay intolerance.

In [7], 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, in example process and data replication [8], dynamic load-balancing [9] or moving virtual machine resources between servers connected into a cloud [10]. In [11], 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 [12]. The authors of [13] propose a M2MC middleware system architecture for m2m applications in broadcast networks (both radio and wired). Because of broadcast orientation, M2MC do not require any resource consuming routing protocols. The system architecture comprises of Message Ordering Protocol, Member Synchronization Protocol and protocols for processes to join and leave the groups.

Other application of m2m communication exist in a field of online gaming [14], [15], [16], [17]. 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 exists, the game world is usually spitted into several zones and users are assigned to the server, taking under account a zone in which their avatar currently exists.

MILP formulation for many-to-many traffic grooming in WDM networks is presented in [18] and [19]. 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 [20]. The authors compare two architectures centralized and distributed. Moreover, the video transmission is encoded using scalable video coding (SVC) [21]. 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 has been presented in [22]. The authors proposed an algorithm to build separate tree for different enhancement layers in SVC based transmission. They make a theoretical analysis to show optimality of the algorithm and prove it during extensive simulations.

In [23], 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 [24]. 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.

III. 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. We divided this abstract model 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.

IV. Overlay m2m systems - optimization model

In this Section, we present the ILP model of the offline m2m flows allocation in overlay system.

First, we introduce the main assumptions of an overlay system with m2m flows. We are given a set of users (overlay nodes) indexed $v = 1, 2, \dots, V$ that participate in the system, i.e., each user generates some stream with rate h_v (defined in bps) 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 α_v is defined for each user - the user receives the overall stream compressed according to this ratio. This assumptions also follows from real teleconference systems [25], [26]. In the considered system, servers $s = 1, 2, \dots, S$ are rendezvous point. In a nutshell, each user sends its flow to a one selected server. The server aggregates all received flows, and next provides the stream to each user with the 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 –

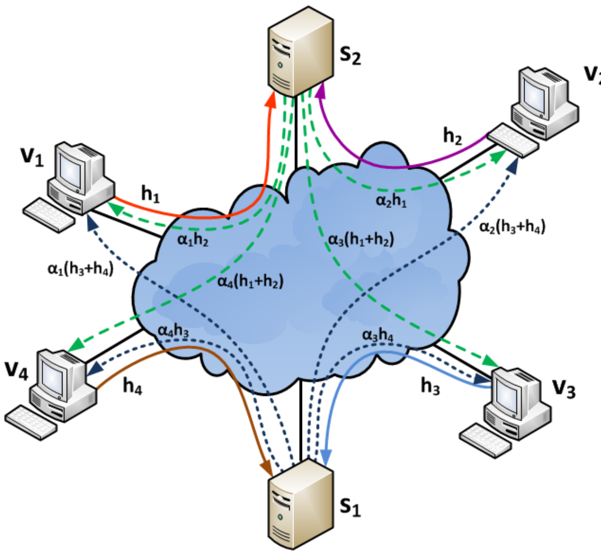

Fig. 1. Many-to-many transmission model in overlay network

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

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 α_1. The former comes from s_2 and consists of stream h_2 from client v_2 (its own steam 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 ζ_{vw} denoting the streaming cost of one capacity unit (e.g., Mbps) 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. To present the model we use notation as in [27].

indices

$v, w = 1, 2, \ldots, V$	user (overlay nodes)
$s = 1, 2, \ldots, S$	servers (overlay nodes)

constants

d_s	download capacity (bps) of server s
u_s	upload capacity (bps) of server s
ζ_{vw}	streaming cost on overlay link from node v to node w
h_v	stream rate (bps) generated by node (client) v
α_v	compression ratio of node (client) v
N_s	maximum number of users that s can server

variables

z_{vs}	= 1, if user v is assigned to server s; 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

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

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

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

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

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

The objective (1) is to minimize the streaming cost of transferring all m2m flows in the system. In more detail, function (1) compromises 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 α_v is given. Moreover, if a particular server s is selected by user v (i.e., $z_{vs} = 1$), the flow of this server is decreased by the flow of user v. Equation (2) assures that for each user v exactly one server is selected. (3) defines the flow served (aggregated) of each server, i.e., the sum of all users flows assigned to s. In (4) and (5) we define the download and upload capacity constrains for servers. Each server uploads the aggregated stream with the defined compression ratio to each user. Therefore, similarly to (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) [26]. The presented model (1)–(6) is strongly NP-hard problems since it is equivalent to the Multidimensional Knapsack Problem [28].

A special case of the overlay model (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, the model (1)–(6) 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 (8) can be used as a reference cost when evaluating multi servers scenarios.

V. Joint m2m systems - optimization model

Now, we introduce a joint model of m2m flows. The main assumptions are analogous to the overlay model. The key difference is that with the joint model, we can optimize network routes between users and servers. We will formulate ILP models using both link-path and node-link notations [27].

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

A. Link-Path Model

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. For each upstream demand there is a set of candidate paths connecting the user node and one of servers. Thus, the selection of one of the candidate paths is equivalent to the selection of the server for a particular demand. 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. 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 $\alpha_{t(d)}(H_{o(d)} - z_{t(d)o(d)} h_{t(d)})$.

There are two additional variables to ensure selection of routes: x_{dp} denotes the flow of demand d allocated to candidate path p and u_{dp} equals to 1, if demand d uses candidate path p and 0 otherwise.

indices (additional)

$d = 1, 2, \dots, D$	demands (upstream from user to server and downstream from server to user)
$p = 1, 2, \dots, P_d$	candidate paths for flows realizing demand d. If d is upstream, path p connects the user node and the server node. If d is downstream, path p connects the server node and the user node
$e = 1, 2, \dots, E$	network links

constants (additional)

δ_{edp}	= 1, if link e belongs to path p of demand d; 0, otherwise
h_d	volume (requested bit-rate) of upstream demand d
ζ_e	streaming cost on link e
c_e	capacity of link e
$ds(d)$	= 1, if d is a downstream demand; 0, otherwise
$up(d)$	= 1, if d is an upstream demand; 0, otherwise
$o(p)$	origin (source) node of path p
$t(p)$	destination node of path p
$o(d)$	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
M	large number

variables (additional)

x_{dp}	flow of demand d allocated to candidate path p (continuous)
u_{dp}	= 1, if demand d uses candidate path p; 0, otherwise (binary)

objective

$$min\ F = \sum_d \sum_p \sum_e \delta_{edp} x_{dp} \zeta_e \quad (9)$$

subject to

$$\sum_p x_{dp} = h_d \quad d = 1, 2, \dots, D \quad up(d) = 1 \quad (10)$$

$$\sum_p x_{dp} = \alpha_{t(d)}(H_{o(d)} - z_{t(d)o(d)} h_{t(d)}) \quad d = 1, 2, \dots, D \quad ds(d) = 1 \quad (11)$$

$$\sum_p u_{dp} = 1 \quad d = 1, 2, \dots, D \quad (12)$$

$$x_{dp} \le M u_{dp} \quad d = 1, 2, \dots, D \quad p = 1, 2, \dots, P_d \quad (13)$$

$$H_s = \sum_{d:up(d)=1} z_{o(d)s} h_d \quad s = 1, 2, \dots, S \quad (14)$$

$$s z_{o(d)s} = \sum_p u_{dp} t(p) \quad d = 1, 2, \dots, D \quad up(d) = 1 \quad s = 1, 2, \dots, S \quad (15)$$

$$\sum_s z_{o(d)s} = 1 \quad d = 1, 2, \ldots, D \quad up(d) = 1 \tag{16}$$

$$\sum_d \sum_p \delta_{edp} x_{dp} \leq c_e \quad e = 1, 2, \ldots, E \tag{17}$$

$$\sum_{d:up(d)=1} z_{o(d)s} \leq N_s \quad s = 1, 2, \ldots, S \tag{18}$$

The objective (9) is to minimize the streaming cost defined as the summary cost of all flows sent on network links. Constraints (10)–(13) define the single path allocation for both upstream and downstream demands. More precisely, (10) guarantees that the whole flow of each upstream demand is allocated. Equation (11) assures the same for downstream demands. Notice that the right-hand side of (11) is the flow received by the user from each server. Recall that the compression ratio is applied and the original stream generated by the node is not sent back. Constraint (12) meets the requirement of single path routing, while constraint (13) binds the flow variable x_{dp} and binary selection variable u_{dp}. Equation (14) – similarly to (3) – defines the flow of server s according to assignment of users to servers. Constraints (15) and (16) define variable z_{vs}. In more detail, the right-hand side of (15) denotes the index of the server node selected for demand d (destination node of the selected candidate path). Therefore, the corresponding variable z_{vs} must be switched on (left-hand side of (15)). Constraint (17) is the link capacity constraint. Finally, (18) limits the number of clients server by each server. Model (9)-(18) is NP-complete since it is equivalent the single path allocation problem [27].

Notice that model (9)–(18) assumes non-bifurcated (single path) routing. To enable bifurcated routing the model should be modified by removing variables u_{dp} and constraints (12)–(13).

B. Node-Link Model

In this section, we present the joint model using node-link notation. The assumptions are analogous to link-path formulation (9)–(18). 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.

constants (additional)

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 (additional)

x_{ed} flow of demand d on link e (continuous)

u_{ed} = 1, if demand d uses link e; 0, otherwise (binary)

objective

$$min\ F = \sum_d \sum_e x_{ed} \zeta_e \tag{19}$$

subject to

$$\sum_e a_{es} x_{ed} - \sum_e b_{es} x_{ed} = \alpha_{t(d)}(H_{o(d)} - z_{t(d)o(d)} h_{t(d)}) \\ d = 1, 2, \ldots, D \quad ds(d) = 1 \\ s = 1, 2, \ldots, S \quad o(d) = s \tag{20}$$

$$\sum_e a_{ev} x_{ed} - \sum_e b_{ev} x_{ed} = -\alpha_{t(d)}(H_{o(d)} - z_{t(d)o(d)} h_{t(d)}) \\ \text{if } v = t(d) \quad d = 1, 2, \ldots, D \\ ds(d) = 1 \quad v = 1, 2, \ldots, V \tag{21}$$

$$\sum_e a_{ev} x_{ed} - \sum_e b_{ev} x_{ed} = 0 \\ \text{if } v \neq t(d) \quad d = 1, 2, \ldots, D \\ ds(d) = 1 \quad v = 1, 2, \ldots, V \tag{22}$$

$$\sum_e a_{ev} x_{ed} - \sum_e b_{ev} x_{ed} = h_d \\ \text{if } \ v = o(d) \quad d = 1, 2, \ldots, D \\ us(d) = 1 \quad v = 1, 2, \ldots, V \tag{23}$$

$$\sum_e a_{es} x_{ed} - \sum_e b_{es} x_{ed} = -h_d z_{o(d)s} \\ d = 1, 2, \ldots, D \quad us(d) = 1 \\ s = 1, 2, \ldots, S \quad t(d) = s \tag{24}$$

$$\sum_e a_{ev} x_{ed} - \sum_e b_{ev} x_{ed} = 0 \\ \text{if } \ v \neq o(d) \quad d = 1, 2, \ldots, D \\ us(d) = 1 \quad v = 1, 2, \ldots, V \tag{25}$$

$$\sum_e a_{es} u_{ed} - \sum_e b_{es} u_{ed} = 1 \\ d = 1, 2, \ldots, D \quad ds(d) = 1 \\ s = 1, 2, \ldots, S \quad o(d) = s \tag{26}$$

$$\sum_e a_{ev} u_{ed} - \sum_e b_{ev} u_{ed} = -1 \\ \text{if } v = t(d) \quad d = 1, 2, \ldots, D \\ ds(d) = 1 \quad v = 1, 2, \ldots, V \tag{27}$$

$$\sum_e a_{ev} u_{ed} - \sum_e b_{ev} u_{ed} = 0 \\ \text{if } v \neq t(d) \quad d = 1, 2, \ldots, D \\ ds(d) = 1 \quad v = 1, 2, \ldots, V \tag{28}$$

$$\sum_e a_{ev} u_{ed} - \sum_e b_{ev} u_{ed} = 1 \\ \text{if } \ v = o(d) \quad d = 1, 2, \ldots, D \\ us(d) = 1 \quad v = 1, 2, \ldots, V \tag{29}$$

$$\sum_e a_{es} u_{ed} - \sum_e b_{es} u_{ed} = -z_{o(d)s} \\ d = 1, 2, \ldots, D \quad us(d) = 1 \\ s = 1, 2, \ldots, S \quad t(d) = s \tag{30}$$

$$\sum_e a_{ev} u_{ed} - \sum_e b_{ev} u_{ed} = 0 \\ \text{if } \ v \neq o(d) \quad d = 1, 2, \ldots, D \\ us(d) = 1 \quad v = 1, 2, \ldots, V \tag{31}$$

$$x_{ed} \leq M u_{ed} \\ d = 1, 2, \ldots, D \quad p = 1, 2, \ldots, P_d \tag{32}$$

$$H_s = \sum_{d:up(d)=1} z_{o(d)s} h_w d \\ s = 1, 2, \ldots, S \tag{33}$$

$$\sum_s z_{o(d)s} = 1 \\ d = 1, 2, \ldots, D \quad up(d) = 1 \tag{34}$$

$$\sum_d x_{ed} \leq c_e \\ e = 1, 2, \ldots, E \tag{35}$$

$$\sum_{d:up(d)=1} z_{o(d)s} \leq N_s \\ s = 1, 2, \ldots, S \tag{36}$$

The objective function (19) minimizes the cost of all network flows. Constraints (20)–(22) 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 (constraint (20)). The right-hand side of (20) denotes the flow of downstream demand d which calculated analogous to eq. (11). Constraint (21) relates to the destination node of the demand, i.e., user node. Finally, constraint (22) is formulated for other so called transit nodes. Furthermore, in (23)–(25) we define the flow conservation of upstream demands, which are anycast. In more detail, eq. (23) denotes the flow conservation for the user node. Constraint (24) meets the guarantee that one of the servers (defined by the value of z_{vs} variable) is selected as the destination node. Eq. (25) 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 (26)–(28) and (29)–(31) denote the flow conservation constraints for corresponding binary flow variables u_{ed}. Both flow variables are bound through using constraint (32). Constraints (33)–(36) are analogous to the link-path model – the only modification is in the link capacity constraint (35), since the link flows follow from flow variables x_{ed}. Notice that in order to obtain bifurcated version of the link-node model variables u_{ed} and constraints (26)–(32) must be removed from the above model.

VI. Concluding remarks

In this paper, we focused on many-to-many transmissions in computer networks. According to many recent developments in computer networks, m2m transmissions have been gaining much popularity. We proposed generic ILP models of m2m flows optimization in overlay model and joint model assuming combined optimization of overlay and underlying networks (e.g., IP layer, MPLS layer, optical layer, etc.). 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, we plan to implement the models in ILP solvers as well as to develop some heuristic algorithms to obtain numerical results. Moreover, we plan to formulate models of m2m systems using multicasting for effective transmission.

References

[1] K. C. Almerotha and M. H. Ammar, *The Use of Multicast Delivery to Provide a Scalable and Interactive Video-on-Demand Service*, IEEE journal on selected areas in telecommunications, vol 14, no. 6, 6 August 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 Communications Magazine, vol. 49, issue 3, pp. 121–127, March 2011

[4] *Cisco Visual Networking Index: Forecast and Methodology 2011-2016*, 2012.

[5] C. H. Stevens, *Many-to-Many Communication*, Technical Report MIT/Sloan/TR-175 Sloan School of Management, Massachusetts Institute of Technology, 1981.

[6] B. Quinn, K. Almeroth, *IP Multicast Applications: Challenges and Solutions*, RFC 3170, September 2001.

[7] S.Banerjee, *et al.*, *Contention-Free Many-to-Many Communication Scheduling for High Performance Clusters*, Distributed Computing and Internet Technology Lecture Notes in Computer Science Volume 6536, pp. 150–161, Springer Berlin Heidelberg, 2011.

[8] N. Touheed, *et al.*, A comparison of dynamic load-balancing algorithms for a parallel adaptive flow solver, Parallel Computing 26(12), pp. 15351554, 2000.

[9] K. Sinha, *et al.*, *Efficient load balancing on a cluster for large scale online video surveillance*, in: V. Garg, *et al.* (eds.) ICDCN 2009. LNCS, vol. 5408, pp. 450455. Springer, Heidelberg, 2008.

[10] T.C. Wilcox Jr., *Dynamic load balancing of virtual machines hosted on Xen*, MS Thesis, Dept. of Computer Science, Brigham Young University, April 2009.

[11] A. Borodin, Y. Rabani, and B. Schieber, *Deterministic many-to-many hot potato routing*, IEEE Transactions on Parallel and Distributed Systems, 8(6):587–596, 1997.

[12] B. S. Chlebus, D. R. Kowalski, T. Radzik, On many-to-many communication in packet radio networks, In: A. A. Shvartsman (ed.), OPODIS 2006, LNCS, vol. 4305, pp. 260–274, Springer, Heidelberg, 2006.

[13] C.K. Bhavanasi, *M2MC: Middleware for Many to Many Communication over broadcast networks*, First International Conference on Communication System Software and Middleware, Comsware, 2006.

[14] F. Glinka, A. Plo, J. Mller-Iden, and S. Gorlatch, *RTF: A Real-Time Framework for Developing Scalable Multiplayer Online Games*, Proceedings of the 6th ACM SIGCOMM workshop on Network and system support for games, p.81-86, Melbourne, Australia, September 19–20, 2007.

[15] A. Ploss, S. Wichmann, F. Glinka, and S. Gorlatch, *From a Single- to Multi-Server Online Game: A Quake 3 Case Study Using RTF*, Proceedings of ACM ACE 2008, Yokohama, Japan, Dec 2008.

[16] P. Prata, E. Pinho, E. Aires, *Database and State Replication in Multiplayer Online Games*, 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA), 2010.

[17] M. Assiotis and V. Tzanov ,*A Distributed Architecture for MMORPG*, NetGames '06 Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games, Article No. 4, 2006.

[18] M.A. Saleh, A.E. Kamal, *Many-to-Many Traffic Grooming in WDM Networks*, IEEE/OSA Journal of Optical Communications and Networking, vol. 1, issue 5, pp. 376–391, October 2009.

[19] M.A. Saleh, A.E. Kamal, *Approximation Algorithms for Many-to-Many Traffic Grooming in Optical WDM Networks*, IEEE/ACM Transactions on Networking, vol. 20 , issue 5, pp. 1527–1540, October 2012.

[20] T. A. Le, H. Nguyen, *Centralized and distributed architectures of scalable video conferencing services*, The Second International Conference on Ubiquitous and Future Networks (ICUFN 2010), Jeju Island, Korea, June 2010, pp. 394–399.

[21] H. Schwarz, D. Marpe, and T. Wiegand *Overview of the Scalable Video Coding Extension of the H.264/AVC Standard*, IEEE Transactions on Circut and Systems for Video Technology, vol. 17, no. 9, September 2007, pp. 1103–1120.

[22] M. Ponec, *et al.*, *Optimizing Multi-rate Peer-to-Peer Video Conferencing Applications*, Proceedings of the 19th ACM international conference on Multimedia, pp. 493–502, NY, USA, 2011.

[23] Y. Amir *et al.*, *Flow Control for Many-to-Many Multicast: A Cost-Benefit Approach*, IEEE Open Architectures and Network Programming (OpenArch), 2001.

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

[25] D. Maldow, *Videoconferencing Infrastructure: A Primer*, Available online: http://www.telepresenceoptions.com/2012/07/videoconferencing_infrastructu/, 2012.

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

[27] M. Piro and D. Medhi, *Routing, flow and capacity design in communication and computer networks*, Morgan Kaufmann, 2004.

[28] J. Puchinger, G. Raidl, U. Pferschy, *The Multidimensional Knapsack Problem: Structure and Algorithms*, Informs Journal on Computing, DOI:10.1287/ijoc.1090.0344, 2009.