ANYCASTING IN CONNECTION–ORIENTED COMPUTER NETWORKS: MODELS, ALGORITHMS AND RESULTS

KRZYSZTOF WALKOWIAK

Chair of Systems and Computer Networks
Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50–370 Wrocław, Poland
e-mail: `krzysztof.walkowiak@pwr.wroc.pl`

Our discussion in this article centers around various issues related to the use of anycasting in connection-oriented computer networks. Anycast is defined as a *one-to-one-of-many* transmission to deliver a packet to one of many hosts. Anycasting can be applied if the same content is replicated over many locations in the network. Examples of network techniques that apply anycasting are Content Delivery Networks (CDNs), Domain Name Service (DNS), Peer-to-Peer (P2P) systems. The role of anycasting is growing concurrently with the popularity of electronic music, movies, and other content required by Internet users. In this work we focus on the optimization of anycast flows in connection-oriented networks. We formulate a model of anycast connections and next propose a heuristic algorithm based on the Lagrangean relaxation aimed to optimize jointly routes for anycast and unicast connections. Results of numerical experiments are presented and evaluated. Finally, we analyze briefly problems related to anycasting in dynamic routing and multi-layer networks.

Keywords: anycasting, connection-oriented networks, optimization, MPLS, survivability.

1. Introduction

Recently, we can observe high-speed deployment of various new networking approaches assuming that the same content (information) is stored in a distributed and surplus manner, e.g., peer-to-peer, web caching, content delivery networks, video streaming. This trend follows from the fact that, due to the increasing number of Internet users and still emerging new network services, there is a deep need to facilitate the distribution of popular content on the Internet (e.g., electronic music, movies, books, software, etc.). Since anycast is a one-to-one-of-many technique to deliver a packet to one of many hosts, it is perceived as a natural transmission technique for the case when some content is replicated in different locations of the network. Therefore, concurrently to robust development of the Internet, the anycast paradigm is to become popular. Obviously, also unicast transmission—defined as *one-to-one*—provides a sufficient framework for delivering replicated content over the network. However, anycasting—as we will try to prove in this paper—can significantly improve network performance in terms of various network metrics, especially its survivability.

1.1. Motivations. First, we will show that anycasting is a robust method to increase network performance. If anycasting is available in the network, the user can select one of many servers (locations) storing the same content according to some criteria including also QoS (Quality of Service) parameters. Consequently, anycast transmission—compared with traditional unicast transmission—reduces network traffic and congestion causing big delays in data delivery. Another benefit is that replica servers provide fault-tolerant service, since users can select another server offering the same data, and even a failure of one server does not cause the data to be unreachable.

Anycasting—as a whole—is a complicated approach and successful implementation requires solving many problems, e.g., replica location, replica ranking, replica consistency, redirection of requests, accounting, security, routing (Baentsch *et al.*, 1997; Hofmann and Beaumont, 2005; Rabinovich, 1998). In this paper we concentrate mainly on one aspect of the anycast approach, i.e., the optimization of anycast flows. We consider an existing backbone network, which is in an operational phase and augmenting its resources (links, capacity, replica servers) or changing the location of replica servers is not possible in a short time perspective. Network design research is a substantial branch of telecommunications science indispensable for telecom and network operators in order

to plan and design transport networks. However, most of previous research in this area is limited to various problems related to unicast flows. For more details, see (Piro and Medhi, 2004) and the references therein. There is a lack of optimization papers on anycast flows, especially in the context of connection-oriented networks. To make our analysis more realistic, we plan to consider joint optimization of unicast and anycast flows. The study on pure anycast flows can be helpful; however, in real networks, various types of flows are mixed.

In this work we focus on anycasting in connection-oriented networks. To motivate our decision on connection-oriented networks, we can enumerate the following arguments. First, according to many observations, e.g., (Krishnan *et al.*, 2000; Paxson, 2006; Rexford *et al.*, 2002), routing on the Internet is relatively stable—the majority of Internet routes change at a frequency lower than once a day. Moreover, most of the routes remain stable over the small lifetime of an HTTP connection (Krishnan *et al.*, 2000). Consequently, the flow on the Internet can be modelled as an unsplittable (non-bifurcated) multicommodity flow—the same as in connection-oriented networks. Second, recently many IP networks have been built over transport networks employing connection-oriented technologies, e.g., MPLS (MultiProtocol Label Switching). Third, connection-oriented approaches—compared with connectionless protocols (e.g., IP)—provide a large support for various traffic engineering mechanisms (Grover, 2004; Rosen *et al.*, 2001; Perros, 2005). Since nowadays corporate and individual clients expect more than just "best effort" network service, network operators are obliged to implement more sophisticated technologies in order to offer services based on the SLA (Service Level Agreement), compromising detailed information on various QoS parameters. Therefore, network technologies supporting QoS configuration—mainly connection-oriented—are now of great interest. Moreover, it is evident that traffic engineering and QoS guarantees are important in network services supported by anycasting. Finally, our widespread bibliographic analysis has shown that there is a relatively small number of research works touching the problem of anycasting in connection-oriented networks. Only several papers of the author cover some problems related to the optimization of anycast connection-oriented flows (Walkowiak, 2005; 2007a; 2007c; 2008). For more details, see Section 2.

1.2. Our contributions. In this paper, we propose and discuss a range of issues related to anycasting. Our contributions include what follows. We present two models of anycast flows in connection-oriented networks. As a reference connection-oriented technology, we use MPLS. We formulate an optimization problem of joint optimization of unicast and anycast unsplittable flows applying various criteria, e.g., network delay, network survivability. We propose efficient heuristic offline algorithms to compute routes for unicast and anycast flows according to various criterion functions. One of algorithms uses a method based on the flow deviation algorithm (Fratta *et al.*, 1973), the second approach is based on the Lagrangean relaxation. To our best knowledge, this is the first study that proposes a Lagrangean-based approach to optimize jointly unicast and anycast flows. We present and discuss results of experiments. The results show that anycasting can significantly improve network performance, especially in terms of network survivability. We demonstrate how to use anycasting in dynamic, online routing in the context of MPLS. We propose and analyze future opportunities in research on anycasting including multilayer networks and peer-to-peer networks.

1.3. Paper overview. The rest of the paper is organized as follows: Section 2 presents the review of related work. In Section 3 we formulate and explain optimization models of anycast flows in a connection-oriented network. Section 4 includes two heuristics for joint optimization of anycast and unicast flows. In Section 5 we present results of numerical experiments. Sections 6 and 7 discuss briefly issues related to anycasting in dynamic routing and multilayer networks, respectively. We conclude our discussion in Section 8.

2. Review of related work

In this section we present a brief review of research works on various aspects of anycasting and network applications employing this approach. The anycasting approach is mainly associated with caching and replication systems. Other examples of techniques that apply anycasting are Domain Name Service (DNS), Peer-to-Peer (P2P) systems, grids, web service, distributed database systems, host auto-configuration, overlay networks, wireless sensor networks, video streaming, telemedicine, etc. (Awerbuch *et al.*, 2003; Ballani and Francis, 2005; Byun and Yoo, 2008; Doi *et al.*, 2004; Hao *et al.*, 2002; Hou *et al.*, 2006; Leuf, 2002; Steinmetz and Wehrle, 2005; Woźniak *et al.*, 1999).

One of the most famous caching technologies that apply anycast traffic is the Content Delivery Network (CDN). The CDN is defined as mechanisms to deliver a range of content to end users on behalf of origin Web servers. The original information is offloaded from source sites to other content servers located in different locations in the network. For each request, the CDN tries to find the closest server offering the requested Web page. The CDN delivers the content from the origin server to the replicas that are much closer to end-users. The set of content stored in the CDNs' servers is selected carefully. Thus, the CDNs' servers can approach the hit ratio of 100%. This

means that almost all requests to replicated servers are satisfied (Hofmann and Beaumont, 2005; Peng, 2004).

Replication can be considered a kind of caching; however, there are some differences. Replication assumes the storing of an object at a place that cannot see the object, while caching is storing an object at a place that sees the source object. This means that a cache notices both hit and miss requests. On the contrary, a replica notices only hits since requests to the replicated server arrive only if that server is believed to have a replica of the requested object. In the presented sense a replica is sometimes called push cache. Replicas can be located in a static or a dynamic manner. In the former case, the system administrator, according to the observed access and traffic characteristics, decides where replicas should be located. In the latter case the system monitors access to various servers and adapts the replica set to the requirements (Rabinovich, 1998). Most of previous optimization works in the context of replication have focused only on problems of replica placement, e.g., (Guha *et al.*, 2001; Jain *et al.*, 2002; Li *et al.*, 1999; Markowski and Kasprzak, 2005; Qiu *et al.*, 2001). These works did not take into account the optimization of network flows.

Earlier works on anycasting that touch somehow issues of network flows concentrate on IP networks using connectionless transmission modelled as bifurcated multicommodity flows (Awerbuch *et al.*, 2003; Ballani and Francis, 2005; Doi *et al.*, 2004; Hao *et al.*, 2002; Li *et al.*, 1999; Peng, 2004). There are only few papers that consider anycasting in the context of connection-oriented transmission. Some works consider the RWA (Routing and Wavelength Assignment) taking into account anycast connections, e.g., (Hyytia, 2004; Tang *et al.*, 2003). In his previous papers (Walkowiak, 2005; 2007a; 2007c; 2008) the author presented various optimization problems and algorithms related to the use of anycasting in MPLS networks.

3. Optimization models of anycast flows in a connection-oriented network

3.1. Modelling of anycasting in connection-oriented networks.
Since we consider a connection-oriented network (e.g., MPLS), we model the network flow as a nonbifurcated multicommodity flow (Piro and Medhi, 2004). The anycast demand (request) in c-o networks can be modelled in two ways: *reduced* and *standard*. In the former case we make use of the important feature of many systems where anycasting is present, i.e., the asymmetry of the flow (Hofmann and Beaumont, 2005). Since anycasting is strongly related to the caching and replication of content in the network, in most cases access to this content is asymmetric. More precisely, a typical user usually fetches much more data from the replica than he/she sends to the replica. This phenomenon can be observed in every-

day use of the Internet—most ISPs' clients use asymmetric access lines (e.g., ADSL). Consequently, the reduced model of anycast demand includes only one connection—the downstream one (from the replica server to the client).

For the sake of simplicity we assume that the network node to which the replica server is connected is equivalent with this server, i.e., we do not take into account the physical connection between the server and the backbone network router. The same is the case in the context of the client—our model includes only the backbone network node to which the client is connected. The upstream connection (from the client to the replica server) applied to carry the client's requests is ignored, due to the fact that the bandwidth requirement of the upstream connection is much more smaller than that of the downstream connection. Therefore, in the *reduced* model, an anycast demand is defined by the following triple: the client node, the set of admissible replica servers and the downstream bandwidth requirement.

In contrast, a classic unicast demand is defined by the following triple: the origin node, the destination node and the bandwidth requirement. In the *standard* model the anycast demand consists of two connections: one from the client to the server (upstream) and the second one in the opposite direction (downstream). Thus, each anycast demand is defined by the following quadruple: the client node, the set of admissible replica servers, the upstream bandwidth requirement, and the downstream bandwidth requirement. Compared with the *standard* model, the main advantage of the *reduced* model is lower complexity, since for each anycast demand only one connection is to be set. However, the *standard* model enables more accurate modelling and next the optimization of anycast flows. In the remainder of the paper we will focus on the *standard* model. Figure 1 illustrates the *standard* model of anycasting in connection-oriented networks.

To establish a unicast demand in connection-oriented networks, a path satisfying the requested bandwidth and connecting the origin and destination nodes must be
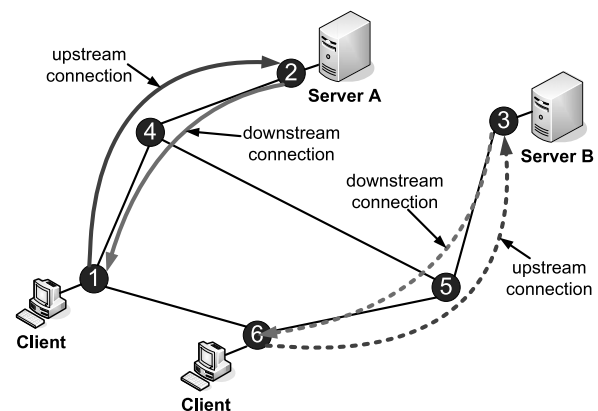

Fig. 1. Illustration of anycast transmission.

found. The case of the anycast demand is more complex. The first step is the server selection process—the client must choose one replica server that will provide the requested content. Next, both paths (upstream and downstream) can be calculated analogously to the unicast approach. However, the main constraint is that both connections associated with a particular anycast demand must connect the same pair of network nodes and one of these nodes must host a replica server.

3.2. Problem formulation. In this section we will formulate the problem of joint optimization of unicast and anycast flows in connection-oriented networks. As mentioned above, to model a real network scenario, we want to optimize not only pure anycast flows, but simultaneously unicast and anycast flows. The network we consider is an existing backbone network. In many cases the network is in an operational phase and augmenting its resources (links, capacity, replica servers) or changing the location of replica servers is not possible in a short time perspective. Therefore, only network flows are optimized.

We formulate a general model that can be applied for the optimization of various criteria—the discussion on the criteria can be found in the subsequent section. Since the *standard* model is employed, each anycast demand compromises two connections: upstream and downstream. To connect both connections associated with the same anycast demand, we introduce a constant $\tau(p)$, which denotes the index of the connection associated with connection p. If p is a downstream connection, $\tau(p)$ must be an upstream connection and vice versa. Furthermore, we assume that for each connection there is a set of candidate routes, which can include all possible paths or only a subset of paths limited, for instance, by the hop-limit approach (Herzberg *et al.*, 1995).

For the unicast connection, candidate routes connect the origin and the destination node of the connection. In the case of the anycast upstream connection, candidate routes origin at the client node and terminated at the server. Finally, for the anycast downstream connection, candidate routes connect the server and the client node. Since there can be many replica servers located in the network, the set of candidate routes of anycast connections include routes to all replica servers.

To mathematically represent the problem, we use the following notation:

Sets

V set of n vertices representing the network nodes.

A set of m arcs representing network directed links.

P set of q connections in the network. A connection can be of two types: unicast and anycast.

P^{UN} set of unicast connections in the network defined by a following triple: the origin node, the destination node and the bandwidth requirement.

P^{AN} set of anycast connections in the network. A connection can be of two types: upstream or downstream. Each anycast connection is defined by the following triple: the client node, the bandwidth requirement, the index of the associated connection.

P^{DS} set of anycast downstream connections in the network.

P^{US} set of anycast upstream connections in the network.

Π_p the index set of candidate routes (paths) for connection p. If p is a unicast connection, route π_p^k connects the origin and the destination node of p. If p is an anycast upstream connection, route π_p^k connects the client node and the replica server. Finally, if p is an anycast downstream connection, candidate routes connect the replica server and the client node.

X set of variables x_p^k, which are equal to one. X determines the unique set of currently selected routes and is called a selection.

Constants

δ_{pa}^k $= 1$, if arc a belongs to route k realizing connection p; 0 otherwise.

Q_p volume (estimated bandwidth requirement) of connection p.

c_a capacity of arc a.

$\tau(p)$ index of the connection associated with connection p. If p is a downstream connection, $\tau(p)$ must be an upstream connection and vice versa.

$o(\pi_p^k)$ origin node of route π_p^k.

$d(\pi_p^k)$ destination node of route π_p^k.

Variables

x_p^k $= 1$, if route π_p^k having index $k \in \Pi_p$ is selected for connection p; 0 otherwise.

f_a flow of arc a.

The CFA (uniCast and anyCast Flow Assignment) problem can be formulated as follows:

$$\min_X \text{FNC}(X) \tag{1}$$

subject to

$$\sum_{k \in \Pi_p} x_p^k = 1, \quad p \in P, \tag{2}$$

$$x_p^k \in \{0,1\}, \quad p \in P, \quad k \in \Pi_p, \tag{3}$$

$$f_a = \sum_{p \in P} \sum_{k \in \Pi_p} \delta_{pa}^k x_p^k Q_p, \quad a \in A, \tag{4}$$

$$f_a \leq c_a, \quad a \in A, \tag{5}$$

$$\sum_{k \in \Pi_p} x_p^k d(\pi_p^k) = \sum_{k \in \Pi_{\tau(p)}} x_{\tau(p)}^k o(\pi_{\tau(p)}^k), \quad p \in P^{AN}. \tag{6}$$

As the objective function (1) any metric $\mathrm{FNC}(X)$ formulated based on selection X can be used. In the next section we present three exemplary functions used in our experiments. The condition (2) states that each connection can use only one route, the constraint (3) assures that decision variables are binary ones, (4) is a definition of an arc flow, the inequality (5) denotes the capacity constraint, the equation (6) guarantees that two routes associated with the same anycast demand connect the same pair of nodes. Notice that the constraint (6) must be satisfied only for anycast connections.

3.3. Criterion functions.

The most common criterion function used in network design problems is the cost of capacity installation (Piro and Medhi, 2004). However, in this work we focus on an allocation problem of unicast and anycast flows and, consequently, network links are given. Another common performance measure proposed for computer networks is the average network delay proposed by Kleinrock in (Kleinrock, 1964) and then widely used in many optimization problems (Burns *et al.*, 2003; Fratta *et al.*, 1973; Kasprzak, 2001; Piro and Medhi, 2004). For ease of reference, in the remainder of the paper we will use the abbreviation 'DEL' to refer to the network delay function. Let us define the network delay function in the following way:

$$\mathrm{DEL}(X) = \frac{1}{T} \sum_{a \in A} \frac{f_a}{c_a - f_a}, \tag{7}$$

where T denotes the total arrive rate of messages in the network, and arcs' flows are calculated according to variables included in selection X. Note that, in the problem with DEL as objective function, (5) should be a sharp inequality. This follows from the fact that the denominator of (7) is the residual, unused capacity of the link. Notice that the optimization of network flows in order to minimize the DEL function assures that flows in network arcs are allocated relatively proportionally, i.e., the relation of the link flow to the link capacity f_a/c_a is similar for all arcs.

In recent years issues of network survivability have gained much attention (Grover, 2004; Piro and Medhi, 2004). Consequently, also new criteria of network optimization concerning network survivability have been developed. In the case of capacity and flow assignment problems, the cost of spare capacity indispensable to provide 100% restoration is the most common objective function (Grover, 2004; Piro and Medhi, 2004). In our case—the allocation problem of flow optimization—first the lost flow function was proposed as the survivability metric

(Murakami and Kim, 1996). Since the network has limited resources of spare capacity, the objective is to minimize consequences of network failure, i.e., the amount of the flow that cannot be restored due to the capacity constraint.

The major drawback of this function is relatively large complexity—the KSP (K-Shortest Path) algorithm is required to compute lost flow. The following procedure is used to find the lost flow. The failed arc a is removed from the network. Next, the KSP algorithm finds k-successively shortest disjoint paths between the origin and the destination node of the failed arc. These paths one by one are saturated with the flow of the failed arc and used for the restoration of flow f_a. The fraction of flow f_a not restored during the running of the KSP algorithm is lost.

Usually an optimization algorithm consists of many subsequent iterations that try to improve the solution and in each iteration the current value of the objective function is needed to evaluate the solution. Therefore, we developed in our previous works two new functions that can be applied as criteria of flow optimization in survivable networks: LFL (Lost Flow in Link) (Walkowiak, 2004) and RCL (Residual Capacity and LFL) (Walkowiak, 2006). The former function denotes a combination of a residual flow, if any, at the origin node and the destination of the failed arc, available for rerouting of broken connections. The LFL function takes into account only the local situation around the failed arc—intuitively, arcs adjacent to the failure can become the bottleneck of the restoration process. The latter function is formulated as a combination of LFL and the DEL function—the numerator of the RCL function is based on the LFL function, the denominator is the residual capacity.

One of the main advantages of the LFL and RCL functions is relatively low complexity compared with other functions proposed for the problem considered, i.e., a maximum flow function and a k-shortest path function (Grover, 2004). A detailed formulation, discussion and results of both functions can be found in (Walkowiak, 2004; 2006) and other works of the same author.

In this work we consider the CFA problem given by (1)–(6) with the following three objective functions: DEL, LFL and RCL. We will refer to these problems as CFA/DEL, CFA/LFL and CFA/RCL, respectively.

4. Heuristics for joint optimization of anycast and unicast flows

Now we present two heuristic algorithms for the CFA problem formulated in the previous section. We will show a detailed version of the algorithms in the context of the DEL function. However, the algorithm can be easily adapted to the LFL and RCL functions. The first

algorithm is a modification of the well-known heuristic approach—the non-bifurcated Flow Deviation (FD) algorithm (Fratta *et al.*, 1973). The second method uses the Lagrangean relaxation approach with subgradient optimization.

4.1. Algorithm based on the flow deviation method.

Flow deviation, proposed in (Fratta *et al.*, 1973), has been applied to a wide range of optimization problems (Burns *et al.*, 2003; Fratta *et al.*, 1973; Kasprzak, 2001; Markowski and Kasprzak, 2005; Piro and Medhi, 2004; Ryba and Kasprzak, 2006; Walkowiak, 2004; 2006). Below we present a heuristic algorithm called CFD (uniCast and anyCast Flow Deviation) based on the original FD method proposed for non-bifurcated flows. The proposed CFD method consists of two parts. The first element— called CFD_INI (uniCast and anyCast Flow Deviation for Initial solution)—is applied to find a starting feasible solution and is based on Phase 1 of FD (Fratta *et al.*, 1973). The second element—named CFD_DEL (uniCast and anyCast Flow Deviation for DEL function)—is developed according to the non-bifurcated FD algorithm included in (Fratta *et al.*, 1973). All modifications in CFD_INI and CFD_DEL compared with the original FD method follow from the model of anycast flows presented in the previous section and connection-oriented flows.

Let $\mathrm{DEL}(H)$ denote of the objective function for a feasible selection H and l_a^{DEL} denote a metric of arc a calculated according to arc flows given by selection H. The common approach is to calculate this metric as a partial derivative of the objective function over the arc flow (Fratta *et al.*, 1973), i.e.,

$$l_a^{\mathrm{DEL}} = \frac{f_a}{T(c_a - f_a)^2}. \tag{8}$$

Let l_a^{INI} be the value of metric l_a^{DEL} calculated for the flow of arc a equal to zero:

$$l_a^{\mathrm{INI}} = \frac{1}{Tc_a}. \tag{9}$$

Algorithm CFD_INI

Let $\underline{Q} = [Q_1, Q_2, \ldots, Q_q]$ denote a vector of connections' volumes (estimated bandwidth requirement). We assume that RE_r is a real number. θ, ϵ—are tolerances applied in the algorithm.

Step 1. Find selection X_1 by assigning all connections to shortest paths calculated under metric l_a^{INI} (9) in the following way. For each unicast connection $i \in P^{\mathrm{UN}}$ find the shortest route $k \in \Pi_i$ under metric l_a^{INI}. For each anycast downstream connection $i \in P^{\mathrm{DS}}$ find the shortest route under metric l_a^{INI}. Next for each anycast upstream

connection $i \in P^{\mathrm{US}}$ find the shortest route $k \in \Pi_i$ under metric l_a^{INI}, for which the following condition is satisfied: $d(\pi_i^k) = o(\pi_{\tau(i)}^j)$, where $x_{\tau(i)}^j \in X_1$. Set $\mathrm{RE}_1 = 1, r = 1$ and go to Step 2.

Step 2. Calculate flows in each arc $a \in A$ using the formula (8) according to selection X_r and volume vector $\underline{Q}_r$. Find

$$\sigma_r = \max_{a \in A} \frac{f_a}{c_a}.$$

If $\sigma_r / \mathrm{RE}_r < 1$, then stop the algorithm. Otherwise let

$$\mathrm{RE}_{r+1} = \frac{\mathrm{RE}_r(1 - \epsilon(1 - \sigma_r))}{\sigma_r},$$

where ϵ is a proper tolerance and $0 < \epsilon < 1$. Let

$$\underline{Q}_{r+1} = \frac{\mathrm{RE}_{r+1}}{\mathrm{RE}_r}\underline{Q}_r.$$

Find

$$X_{r+1} = \mathrm{CFD\_DEL}(X_r, \underline{Q}_{r+1})$$

and go to Step 3.

Step 3. If

$$|\mathrm{RE}_{r+1} - \mathrm{RE}_r| < \theta,$$

where θ is a proper positive tolerce, stop: the problem is infeasible within tolerance θ. Otherwise set $r = r + 1$ and go to Step 2.

In Step 2 of CFD_INI we apply the algorithm CFD_DEL (see below), which is run for the values of decision variables given by selection X_r and connections' volume given by the vector $\underline{Q}_r$. The complexity of the CFD_INI algorithm depends on the selected values of tolerance parameters θ, ϵ. The values of parameters θ, ϵ should be selected to find a balance between the algorithm's precision and execution time.

Algorithm CFD_DEL

Operator $\mathrm{first}(B)$ returns the index of the first connection in set B. F and H are selections.

Step 1. Find feasible selection X_1 using the CFD_INI algorithm. If CFD_INI cannot find a feasible solution, stop the algorithm. Otherwise set $r = 1$, and go to Step 2.

Step 2. Compute $\mathrm{SR}(X_r)$, defined as the set of shortest routes under metric $l_a^{\mathrm{DEL}}(X_r)$ in the following way. For each unicast connection $i \in P^{\mathrm{UN}}$ find the shortest route $k \in \Pi_i$ under metric $l_a^{\mathrm{DEL}}(X_r)$. For each anycast downstream connection $i \in P^{\mathrm{DS}}$ find the shortest route $k \in \Pi_i$ under metric $l_a^{\mathrm{DEL}}(X_r)$. Next, for each anycast upstream connection $i \in P^{\mathrm{US}}$ find the shortest route $k \in \Pi_i$ under metric $l_a^{\mathrm{DEL}}(X_r)$, for which the following condition is satisfied: $d(\pi_i^k) = o(\pi_{\tau(i)}^j)$, where $x_{\tau(i)}^j \in \mathrm{SR}(X_r)$.

Step 3. Set $H = X_r$ and $K = P^{\mathrm{UN}} \cup P^{\mathrm{DS}}$.

Step 3a. Find $i = \mathrm{first}(K)$. If $i \in P^{\mathrm{UN}}$, set $F = (H - \{x_i^j\}) \cup \{x_i^k\}$, where $x_i^k \in \mathrm{SR}(X_r)$ and $x_i^j \in H$. Otherwise, if $i \in P^{\mathrm{AN}}$, set $F = (H - \{x_i^j\}) \cup \{x_i^k\}$, where $x_i^k \in \mathrm{SR}(X_r)$, $x_i^j \in H$, and next set $F = (H - \{x_{\tau(i)}^{j'}\}) \cup \{x_{\tau(i)}^{k'}\}$, where $x_{\tau(i)}^{k'} \in \mathrm{SR}(X_r)$, $x_{\tau(i)}^{j'} \in H$.

Step 3b. If F is a feasible selection and $\mathrm{DEL}(F) < \mathrm{DEL}(H)$, let $H = F$.

Step 3c. Set $K = K - \{i\}$. If $K = 0$, go to Step 4. Otherwise, go to Step 3a.

Step 4. If $H = X_r$, stop. The algorithm cannot improve the solution any further. Otherwise, let $X_{r+1} = H$, $r = r + 1$ and go to Step 2.

The algorithm CFD_DEL is related to the FD algorithm for non-bifurcated flows described in (Fratta *et al.*, 1973). However, CFD_DEL can assign jointly unicast and anycast connections, while classical FD optimizes only unicast connections. Notice that CFD_DEL processes unicast connections analogously as in the non-bifurcated version of FD, but anycast connections are processed in a different way. This follows from our model of the anycast demand, which consists of two connections: upstream and downstream. Since both associated anycast connections must connect the same pair of nodes (constraint (6)), they have to be considered jointly. However, due to the asymmetry of anycast flows mentioned above, first the shortest route of the downstream connection is calculated taking into account all replica server networks. In consequence, the upstream connection can select among routes between the client node and the server node already assigned for the downstream connection.

Theorem 1. *The maximum number of algorithm* CFD_DEL *iterations is*

$$\left(\prod_{p \in P^{\mathrm{UN}}} (\zeta_p) \right) \left(\prod_{p \in P^{\mathrm{DS}}} \left(\sum_{v \in V} \left(\zeta_p^v \zeta_{\tau(p)}^v \right) \right) \right) - 1,$$

where ζ_p is the number of candidate routes for $p \in P^{\mathrm{UN}}$. ζ_p^v is the number of candidate routes connecting the client node and the server node v for $p \in P^{\mathrm{AN}}$.

Proof. The number of all possible candidate routes for the unicast connection $p \in P^{\mathrm{UN}}$ is ζ_p. Consequently, the number of routes' combinations for all unicast connections is $\left(\prod_{p \in P^{UN}} (\zeta_p) \right)$. In the case of a pair of associated anycast connections $p \in P^{\mathrm{DS}}$ and $\tau_p \in P^{\mathrm{US}}$ the number of feasible (in terms of the constraint (6)) routes' combinations is $\sum_{v \in V} \left(\zeta_p^v \zeta_{\tau(p)}^v \right)$, where ζ_p^v denotes the number of candidate routes connecting the client node of p and the server node v for $p \in P^{\mathrm{AN}}$. Note

that we analyze all possible server nodes. Thus, the number of routes' combinations for all anycast connections is $\left(\Pi_{p \in P^{\mathrm{DS}}} \left(\sum_{v \in V} \left(\zeta_p^v \zeta_{\tau(p)}^v \right) \right) \right)$.

Taking into account both unicast and anycast connections, the number of all routes' combinations is thus $\left(\Pi_{p \in P^{\mathrm{UN}}} (\zeta_p) \right) \left(\Pi_{p \in P^{\mathrm{DS}}} \left(\sum_{v \in V} \left(\zeta_p^v \zeta_{\tau(p)}^v \right) \right) \right)$. As we start with a feasible initial selection and repetitions of the same flow are impossible, the maximum number of CFD_DEL iterations is $\left(\Pi_{p \in P^{\mathrm{UN}}} (\zeta_p) \right) \left(\Pi_{p \in P^{\mathrm{DS}}} \left(\sum_{v \in V} \left(\zeta_p^v \zeta_{\tau(p)}^v \right) \right) \right) - 1$. ∎

For more information on the algorithms CFD_INI and CFD_DEL, see (Walkowiak, 2008).

4.2. Lagrangean heuristic. The main idea of Lagrangean heuristic is to formulate the dual problem by relaxing some constraints of the primal problem and next solving the dual by a subgradient algorithm. As in the previous subsection, we present below the Lagrangean approach in the context of the network delay function. However, we will show how to modify this method to apply also other objective functions.

In order to formulate a dual problem for (1)–(6), we will use the same approach as in (Gavish and Huntler, 1983; Walkowiak, 2007b). First, we transform the problem CFA/DEL into an equivalent formulation, which is better suited for a Lagrangean relaxation procedure. Since the DEL function (7) is not decreasing with f_a, we can replace (4) with an inequality leading to the following problem:

$$\omega = \min_X \frac{1}{T} \sum_{a \in A} \frac{f_a}{(c_a - f_a)} \tag{10}$$

subject to (2)–(3), (5)–(6) and

$$\sum_{p \in P} \sum_{k \in \Pi_p} \delta_{pa}^k x_p^k Q_p \leq f_a, \quad a \in A. \tag{11}$$

For ease of reference, we call this problem CFA2/DEL. Notice that problems CFA/DEL and CFA2/DEL are equivalent—the optimal solution obtained for one of these two problems holds the optimality also for the second problem.

A popular approach of the Lagrangean relaxation applied to various communication problems is to relax the capacity constraint, e.g., (Grover, 2004; Holmberg, 1995; Holmberg and Yuan, 1998; 2000; Piro and Medhi, 2004). In our case we use the approach proposed in (Gavish and Huntler, 1983) and relax the constraint (11) using a vector $\underline{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_m]$ of positive Lagrangean multipliers λ_a, for each $a \in A$. Consequently, the following La-

grangean relaxation of problem CFA2/DEL is formulated:

$$\varphi(\underline{\lambda})$$

$$= \min_{f_a, x_p^k} \left\{ \frac{1}{T} \sum_{a \in A} \frac{f_a}{c_a - f_a} \right.$$

$$\left. + \sum_{a \in A} \lambda_a \left(\sum_{p \in P} \sum_{k \in \Pi_p} \delta_{pa}^k x_p^k Q_p - f_a \right) \right\} \quad (12)$$

subject to (2)–(3), (5)–(6) and

$$f_a \geq 0, \quad a \in A. \quad (13)$$

Notice that we add the additional constraint (13). In the problem CFA2/DEL, the constraint (13) follows directly from the conditions (3) and (11). Since (11) is incorporated in the dual function (12), we have to add to the problem the constraint (13) to guarantee that variables f_a are positive.

The objective function of the Lagrangean problem can be rewritten as

$$\varphi(\underline{\lambda}) = \min_{f_a, x_p^k} \left\{ \left(\frac{1}{T} \sum_{a \in A} \left(\frac{f_a}{c_a - f_a} - \lambda_a f_a \right) \right) \right.$$

$$\left. + \left(\sum_{a \in A} \lambda_a \sum_{p \in P} \sum_{k \in \Pi_p} \delta_{pa}^k x_p^k Q_p \right) \right\}. \quad (14)$$

Since there are no coupling constraints between variables f_a and variables x_p^k, the problem can be divided into two independent subproblems:

Subproblem 1:

$$\varphi_1(\underline{\lambda}) = \min_{f_a} \left(\frac{1}{T} \sum_{a \in A} \left(\frac{f_a}{c_a - f_a} - \lambda_a f_a \right) \right) \quad (15)$$

subject to (5) and (13).

Subproblem 2:

$$\varphi_2(\underline{\lambda}) = \min_{x_p^k} \left(\sum_{a \in A} \lambda_a \sum_{p \in P} \sum_{k \in \Pi_p} \delta_{pa}^k x_p^k Q_p \right) \quad (16)$$

subject to (2)–(3), and (6).

According to (Gavish and Huntler, 1983), Subproblem 1 can be devided into $m = |A|$ subproblems and each of such subproblems is solved by

$$f_a = \begin{cases} \left(c_a - \sqrt{\dfrac{c_a}{T\lambda_a}} \right) & \text{when} \quad \lambda_a > \dfrac{1}{Tc_a}, \\ 0 & \text{otherwise.} \end{cases} \quad (17)$$

Notice that Subproblem 2 can be separated into two kinds of subproblems: unicast and anycast. For unicast connections, there are $\left| P^{\text{UN}} \right|$ subproblems—one for each unicast connection $p \in P^{\text{UN}}$:

$$\min_{x_p^k} \sum_{k \in \Pi_p} \phi_p^k x_p^k \quad (18)$$

subject to (2)–(3). Here ϕ_p^k denotes the length of route $k \in \Pi_p$ calculated according to link metrics equal to $\lambda_a Q_p$ for each arc $a \in A$:

$$\phi_p^k = \sum_{a \in A} \lambda_a \delta_{pa}^k Q_p. \quad (19)$$

The minimum of (18) is achieved if we select the shortest route calculated under metric $\lambda_a Q_p$. Hence, to solve Subproblem 2 in the context of unicast connections, a number of shortest path problems must be solved, e.g., using the Dijkstra algorithm.

For anycast connections we obtain $\left| P^{\text{AN}} \right| / 2$ subproblems formulated for each anycast downstream connection $p \in P^{DS}$ in the following way:

$$\min_{x_p^k, x_{\tau(p)}^j} \left(\sum_{k \in \Pi_p} \phi_p^k x_p^k + \sum_{k \in \Pi_{\tau(p)}} \phi_p^k x_{\tau(p)}^k \right) \quad (20)$$

subject to (2)–(3), and (6).

Solving this subproblem is a little bit more complicated compared with the related unicast problem. First of all, the additional constraint (6) related to the anycast flow must be considered. Therefore, upstream and downstream connections of the same anycast demand must be processed jointly. Consequently, for each pair of connections we calculate a shortest path using Dijkstra under the metric $\lambda_a Q_p$ to each of content servers. Next, we select a pair of paths that satisfy the constraint (6) and have the smallest sum of lengths. Such a procedure guarantees optimal solution of Subproblem 2.

Since the procedure of solving the dual problem is known, we can proceed to the subgradient search. Let $x_p^k(\underline{\lambda})$ and $f_a(\underline{\lambda})$ be an optimal solution of the Lagrangean relaxation for a fixed vector of multipliers $\underline{\lambda}$. Let $\underline{x}$ denote a set of all variables x_p^k, which are equal to one. The corresponding subgradient of the dual function (14) at $\underline{\lambda}$ is given by

$$\gamma(\underline{\lambda}) = \sum_{p \in P} \sum_{k \in \Pi_p} \delta_{pa}^k x_p^k(\underline{\lambda}) Q_p - f_a(\underline{\lambda}), \quad a \in A. \quad (21)$$

We use the subscript i to denote the current iteration of the subgradient algorithm. The multipliers are updated after iteration i in the following way:

$$\lambda_a^{i+1} = \max(0, \lambda_a^i + t^i \gamma_a^i), \quad a \in A. \quad (22)$$

The step-size, t^i, can be given as (Gavish and Huntler, 1983; Piro and Medhi, 2004)

$$t^i = \frac{\rho(\overline{\varphi} - \varphi(\underline{\lambda}^i))}{\|\gamma^i\|}. \quad (23)$$

Here $\overline{\varphi}$ denotes un upper bound on the dual function, which can be calculated using a heuristic algorithm that can find a feasible solution of the primal problem. Note that ρ is commonly used in the range $0 < \rho < 2$ (Piro and Medhi, 2004).

Let $\underline{f} = [f_1, f_2, \ldots, f_m]$ denote a vector of flows f_a in each arc $a \in A$. The following algorithm—called CLR_DEL (uniCast and anyCast Lagrangean Relaxation for DEL function)—presents the subgradient optimization procedure applied to the Lagrangean relaxation of the CFA/DEL problem.

Algorithm CLR_DEL

Step 0. Select an initial $\underline{\lambda}^0$. Set $\rho = 2$, $\rho_{\min} = 0.005$, $i = 0$, $\rho_{\text{iter}} = 0$, $i_{\max} = 100$, $\rho_{\text{maxiter}} = 3$, $\omega^{best} = \infty$, $\underline{\lambda}^* = \underline{\lambda}^0$. Using a heuristic, compute overestimate $\overline{\varphi}$ of $\varphi(\underline{\lambda}^*)$ or set $\overline{\varphi}$ to an arbitrary large value.

Step 1. Set $i = i + 1$, $\rho_{\text{iter}} = \rho_{\text{iter}} + 1$. Given $\underline{\lambda}^i$, solve the Lagrangean relaxation of CFA/DEL as decoupled subproblems to obtain $\varphi(\underline{\lambda}^i)$, $\underline{x}^i$, $\underline{f}^i$.

Step 2a. If $\varphi(\underline{\lambda}^i) > \varphi(\underline{\lambda}^*)$, then $\underline{\lambda}^* > \underline{\lambda}^i$ and $\rho_{\text{iter}} = -1$.

Step 2b. Use $\underline{x}^i$, $\underline{f}^i$ to compute the feasible primal objective ω of the problem using the CFD_DEL algorithm. If $\omega < \omega^{\text{best}}$, then $\omega^{\text{best}} = \omega$, $\underline{x}^{\text{best}} = \underline{x}^i$, $\underline{f}^{\text{best}} = \underline{f}^i$, $\overline{\varphi} = \omega^{\text{best}}$.

Step 2c. If $\rho_{\text{iter}} > \rho_{\text{maxiter}}$, then $\rho = \max(\rho/2, \rho_{\min})$ and $\rho_{\text{iter}} = 0$.

Step 3. Use decoupled solutions $\underline{x}^i$, $\underline{f}^i$ to compute: subgradient $\gamma^i(\underline{\lambda}^i)$ (refer to (21)); step size t^i (refer to (23)); multipliers $\underline{\lambda}^{i+1}$ (refer to (22)).

Step 4. If $i > i_{\max}$, stop. Otherwise go to Step 1.

The objective of the CLR_DEL algorithm is to find a suboptimal, feasible solution of the CFA/DEL problem. Notice that in Step 1 the algorithm generates vector $\underline{x}^i$ denoting routes of each connection. This solution is optimal for the Lagrangean dual, however, there is no guarantee that $\underline{x}^i$ is feasible (in terms of the problem constraints) for the primal problem. Therefore, it is substantial to obtain a primal feasible solution from the dual solution. Thus, we employ the CFD_DEL heuristic procedure that tries to improve the solution yielded by the subgradient search. The execution time of CLR_DEL depends mainly on the number of iterations given by parameter $i_{\max}$. The most time consuming part of each iteration is the CFD_DEL algorithm. For a discussion on computational complexity of CFD_DEL algorithm, see Section 4.1.

If vector $\underline{x}^i$ obtained in an iteration of CLR_DEL constitutes a feasible solution (in terms of the capacity constraint) of the primal problem, we use it as an initial solution for CFD_DEL. Otherwise, vector $\underline{x}^i$ is an input solution for the CFD_DEL algorithm. In conclusion,

Table 1. Average performance of heuristic algorithms compared to optimal results.

Function	No. of cases	Gap to optimal	
		CFD	CLR
DEL	90	2.4%	0.1%
RCL	99	4.1%	0.4%
LFL	71	9.6%	1.6%

CLR_DEL can be viewed as intelligent iterative application of the CFD_DEL initial solution generated by the subgradient optimization procedure solving the dual problem.

Notice that the CLR_DEL algorithm can be adapted to a CFA problem with another objective function. The required modifications are the solution of Subproblem 1 and the application of a different version of CFD algorithm. In addition to the CLR_DEL algorithm (the objective is the network delay function), we have developed the CLR_LFL and CLR_RCL algorithms with the LFL and RCL functions, respectively. The functions LFL and RCL have been defined in the context of flow optimization in survivable connection-oriented networks. More information on these functions and unicast FD algorithms applying these functions can be found in (Walkowiak, 2004; 2006). The method of solving Subproblem 1 for the LFL function is included in (Walkowiak, 2007b).

5. Results

We implemented the above presented heuristic algorithms in C++ to verify the performance of the proposed algorithms against optimal results. To obtain optimal results in the context of a piecewise linear LFL function, we used the lp_solve version 5.5.0.6. For the functions DEL and RCL, which are not linear, we developed branch-and-cut algorithms. In order to find an optimal solution in reasonable time of about one hour, we had to consider small networks consisting of ten nodes (four topologies) and 14 nodes (one topology). The number of replica servers varies from 2 to 4. We generated various demand patterns. In Table 1 we report the average gap between results of the heuristic algorithms CFD and CLR and the optimal results. We can see that in the case of the DEL function the algorithm based on the Lagrangean relaxation is only 0.1% worse than the optimal one. In the context of the RCL and LFL functions, the corresponding gap is 0.4% and 1.6%, respectively. The optimality gap varies for tested functions. This follows mainly from the construct of the functions considered. The DEL function is convex, while the other functions have a more irregular structure, especially LFL. These results prove that the proposed CLR method yields results very close to optimal ones.
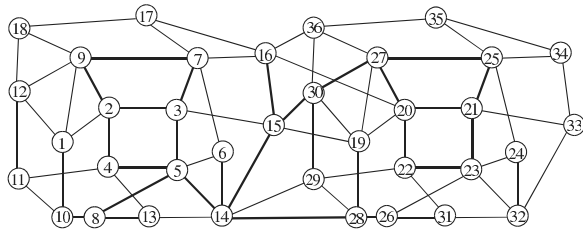
Fig. 2. Topology of the tested network.

Table 2. Location of replica servers—simulation cases.

Number of servers	Location of servers
2	5,23
2	5,25
2	7,23
2	7,25
3	5,23,30
3	5,25,30
3	7,14,23
3	7,14,25
4	5,9,23,30
4	5,9,25,30
4	7,14,23,30
4	7,14,25,30

Fig. 3. Average difference of the objective function value between the algorithms ULR and CLR for various objective functions.

Next we performed extensive tests of the proposed heuristics over a larger network consisting of 36 nodes and 144 directed links (Fig. 2). The bold lines represent links of the size of 96 units while other lines are links of the size of 48 units. The link capacities were selected to model capacity ratio of OC-48 circuits used in optical networks. In the simulations, the link capacities were scaled by a factor of 100 to enable establishing thousands of connections. As the tested network is an operational phase and augmenting network resources is not possible, the number of replica servers is constant. However, to verify the performance of our approach in various cases, we test 12 various server location scenarios (Table 2). The number of servers varies from 2 to 4. For each server location we test 13 different demand patterns consisting of 180 anycast demands (360 anycast connections) and 2500 unicast connections. In experiments we apply the Anycast Ratio (AR) parameter defined as the ratio of the anycast demands' bandwidth requirement to the bandwidth requirement of all demands issued in the network. Demand patterns are generated randomly satisfying various values of the AR.

The objective of the simulation tests was twofold. First, we want to compare the performance of the CLR algorithm with the approach used commonly in CDNs, i.e., each request is assigned to the closest (in terms of the hop number) replica server. Thus all anycast requests can be treated as unicast ones (the server node is constant). Consequently, the routes for both unicast
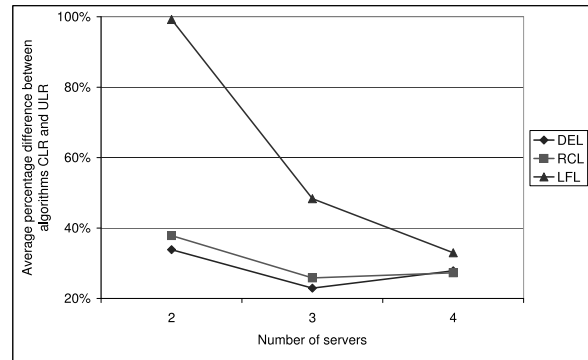
and anycast demands are selected using a traditional unicast version of the Lagrangean relaxation heuristic developed for the DEL, LFL and RCL functions in (Gavish and Huntler, 1983; Walkowiak, 2007b), respectively. For ease of reference, we call these algorithms ULR_DEL (Unicast Lagrangean Relaxation with DEL function), ULR_LFL and ULR_RCL, respectively. The major distinction between the CLR and ULR algorithms is that the former enables the anycast demand to change the content server during subsequent iterations of the algorithm, while the latter assigns clients to servers in a static manner. The second goal of simulations is to examine how the use of the anycast flow can influence network survivability.

In Fig. 3 we compare the performance of CLR with that of ULR. Each curve in the graph represents the average percentage difference between both algorithms obtained for one of the following objective functions: DEL (network delay), LFL and RCL. We can observe that the largest difference between both tested algorithms is obtained for the LFL function, which follows from the formulation of the LFL function (Walkowiak, 2004). The average difference between CLR and ULR is 26%, 28% and 48% for the functions DEL, RCL and LFL, respectively. Figure 4 shows the execution time of all tested algorithms.

Figure 5 presents the performance of the CLR_DEL, CLR_RCL, CLR_LFL algorithms in terms of the average link utilization, which shows the average saturation of network links. We can easily observe that all algorithms provide similar results; however, the use of CLR_LFL leads to the least consumption of network capacity resources. Another important observation is that the average link utilization decreases when the number of content servers grows.

The next figures relate to network survivability. The scenario of experiments is as follows. First we calculate working routes of all connections (anycast and unicast) applying the CLR or the ULR algorithm. Next, we simulate a failure of each link and perform the network restora-
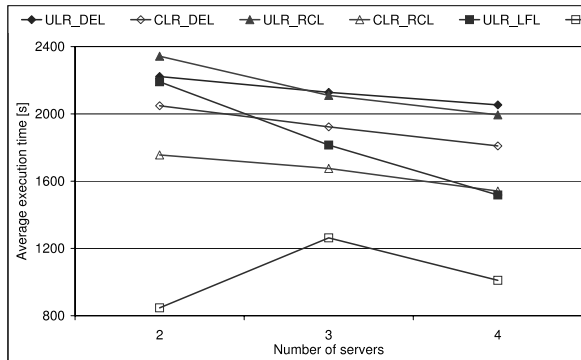
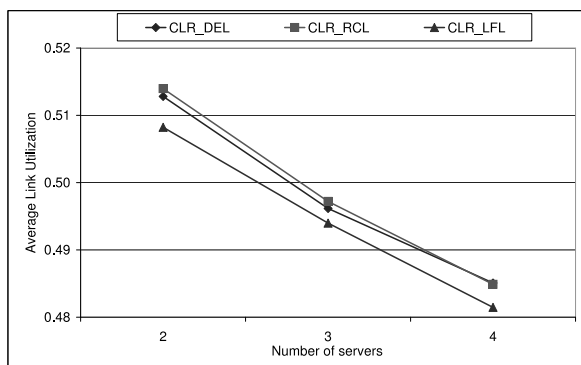Fig. 4. Average execution time of the algorithms ULR and CLR for various objective functions.

Fig. 5. Average link utilization for various objective functions.

tion process. Finally, we calculate the lost flow, i.e., the flow that was not restored during the restoration process.

Figures 6 and 7 show the comparison of the CLR_DEL, CLR_RCL and CLR_LFL algorithms in terms of the lost flow function. In Fig. 6 we present the average ranking of each algorithm—the best algorithm (providing the lowest value of the lost flow function) in one individual case receives three points, the second one two points and the worse one point point. Figure 7 reports the number of first places of the algorithms in terms of the lost flow function. In general, the use of the RCL function guarantees the best network survivability. Similar results were obtained in the context of unicast flows (Walkowiak, 2006).

Figure 8 shows the average difference in terms of the lost flow function between the algorithms ULR and CLR for all three functions. From this figure we can observe that the application of CLR to optimize working routes can significantly reduce the amount of the lost flow (about 50%). For more issues on the use anycasting in survivable networks, refer to (Walkowiak, 2007a).

In short, we can conclude that the CLR algorithm yields much better performance than ULR in many aspects. However, detailed observation shows that the difference depends on the objective function and the number of servers. For results showing the performance of the

CFD algorithms developed for the functions DEL, LFL and RCL, see (Walkowiak, 2008).

6. Dynamic routing of anycast demands

The above discussion on anycasting was focused on offline problems, in which the decision on route selection do not have to be made online. However, in many cases, network devices (e.g., routers) must made a decision on route selection in an online manner. Therefore, dynamic routing algorithms are developed. There has been a lot of research on unicast dynamic routing algorithms in connection-oriented MPLS networks, e.g., see (Bagula *et al.*, 2004; Kar *et al.*, 2000; Szeto *et al.*, 2002). The most widespread approach to unicast dynamic routing is the Shortest Path First (SPF) algorithm based on an administrative weight (metric). A popular metric is the number of hops applied in the Min Hop Algorithm (MHA). However, more complex approaches to calculate the link metric have been developed, e.g., (Bagula *et al.*, 2004; Kar *et al.*, 2000; Szeto *et al.*, 2002). Since in anycasting one of the connection's endpoints is unknown and is to be selected, for dynamic routing of anycast requests new methods are required. We propose two approaches to dynamic routing of anycast demands. In the first one, the process of anycast dynamic routing is decomposed into two
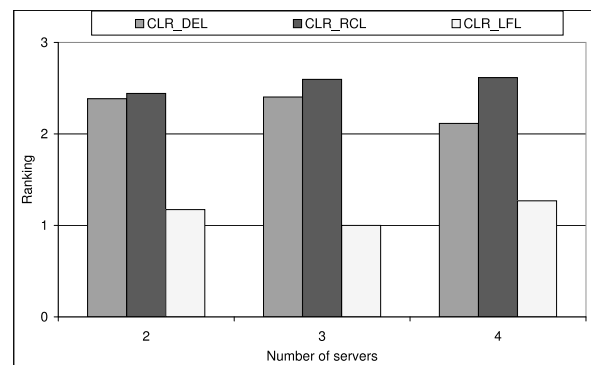

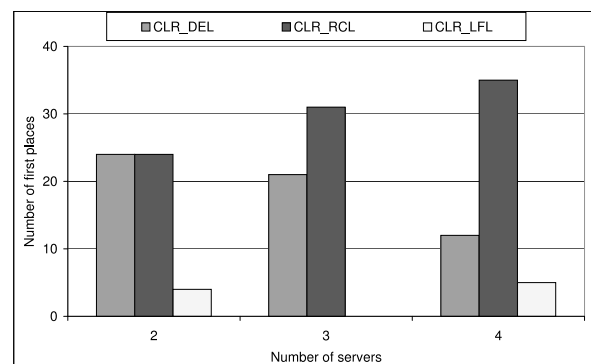Fig. 6. Ranking of algorithms in terms of the lost flow function.

Fig. 7. Number of first places of algorithms in terms of the lost flow function.
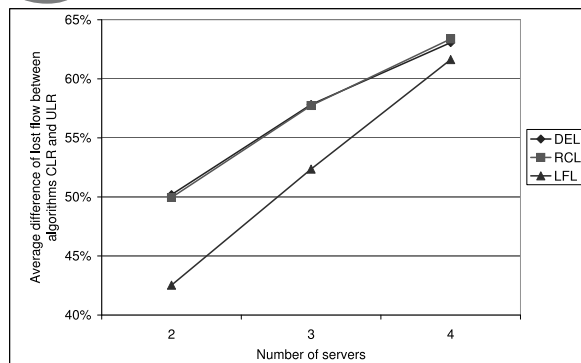
Fig. 8. Average difference in terms of the lost flow function between the algorithms ULR and CLR.

phases. Initially, the replica server selection process is completed—three server selection algorithms to solve this problem were proposed in (Walkowiak, 2005). When all anycast demands are assigned to replica servers, both endpoints of anycast connections are established and a traditional unicast dynamic routing algorithm can be applied. The second approach consists in simultaneous selection of replica servers and routes, thus new routing algorithms are indispensable. For more information on dynamic routing of anycast demands, refer to (Walkowiak, 2005; 2007c).

7. Anycasting in multilayer networks

The model of the network considered in the previous sections is a single-layer network. However, resources of computer networks are organized in a multi-layered manner and form a hierarchical structure with each layer being a proper network on its own. The links of an upper layer are formed using paths of the lower layer, and this pattern repeats as one goes down the resource hierarchy. Consequently, in the upper layers, flows of individual network users are considered, while in the lower layers, the network flow is formed by aggregated transmission produced by a large number of individual transmissions (Piro and Medhi, 2004). Since most of optimization problems related to computer networks are very difficult, in many cases only one network layer is included in the model. However, in recent years we have noticed growing interest in multi-layer network design and optimization. In this section we will try to answer the question if anycasting can be applied in multi-layer networks.

As mentioned above, anycast transmission can be applied if the same content is replicated and stored in many locations in the network. The access to content servers is usually realized on the level of individual users—due to the heterogeneity of Internet users it is difficult to aggregate it. Therefore, it is obvious that anycasting should be applied in upper layers of the multi-layer network. The modeling can be carried out in the same way as presented above in the context of a single-layer network. In the

multi-layer architecture the aggregate flow of the upper layer is a kind of input to the lower layer. The lower layer is realized autonomously, i.e., independently of the nature of individual upper layer flows the lower layer works in the same manner. Consequently, the application of anycasting in the upper layer will not oblige us to any changes in the lower layer.

8. Concluding remarks

This work has been focused on various aspects of anycast flows in connection-oriented networks. We have formulated the problem of joint optimization of unicast and anycast flows in connection-oriented networks. This problem is motivated by growing popularity of network services that provide Internet users with content like music, movies, software, etc. We have developed and examined two heuristic algorithms to the optimization problem. There has not been, however, any study we are aware of that uses the Lagrangean relaxation approach to joint optimization of unicast and anycast flows in computer networks. Results of simulations have proven that the application of anycasting can significantly reduce the value of different performance metrics including network delay and network survivability. We have also examined how anycasting can be used in the context of dynamic routing and multi-layer networks.

Our study has several practical implications. One is to provide the designers of computer networks with a new method that enables effective optimization of unicast and anycast flows. Another important implication follows from the fact that network flows in peer-to-peer networks can be modeled as anycast (in P2P systems the same content is replicated in many peers). Thus, the results presented in this paper can be used to optimize network flows in P2P systems. In our ongoing works we plan to work on the optimization of flows in P2P systems taking advantage of our experience on anycasting.

References

Awerbuch, B., Brinkmann, A. and Scheideler, C. (2003). Anycasting in adversarial systems: Routing and admission control, *in* J.C.M. Baeten, J.K. Lenstra, J. Parrow and G.J. Woeginger (Eds), *Automata, Languages and Programming*, Lecture Notes in Computer Science, Vol. 2719, Springer-Verlag, Berlin, pp. 1153–1168.

Baentsch, M., Baum, L., Molter, G., Rothkugel, S. and Sturm, P. (1997). World wide web caching: The application-

level view of the internet, *IEEE Communications Magazine* **35**(6): 170–178.

Bagula, B., Botha, M. and Krzesinski, A. (2004). Online traffic engineering: The least interference optimization algorithm, *Proceedings of the IEEE International Conference on Communications, ICC 2004, Paris, France*, pp. 1232–1236.

Ballani, H. and Francis, P. (2005). Towards a global IP anycast service, *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM SIGCOMM 2004, Philadelphia, PA, USA*, pp. 301–312.

Burns, J., Ott, T., Krzesinski, A. and Muller, K. (2003). Path selection and bandwidth allocation in MPLS networks, *Performance Evaluation* **52**(2): 133–152.

Byun, S. and Yoo, C. (2008). Minimum DVS gateway deployment in DVS-based overlay streaming, *Computer Communications* **31**(3): 537–550.

Doi, S., Ata, S., Kitamura, K. and Murata, M. (2004). IPv6 anycast for simple and effective service-oriented communications, *IEEE Communications Magazine* **42**(5): 166–171.

Fratta, L., Gerla, M. and Kleinrock, L. (1973). The flow deviation method: An approach to store-and-forward communication network design, *Networks* **3**(2): 97–133.

Gavish, B. and Huntler, S. (1983). An algorithm for optimal route selection in SNA networks, *IEEE Transactions on Communications* **31**(10): 1154–1160.

Grover, W. (2004). *Mesh-based Survivable Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking*, Prentice-Hall PTR, Upper Saddle River, NJ.

Guha, S., Meyerson, A. and Munagala, K. (2001). Improved approximation algorithms for fault tolerant facility location, *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2001, Washington, DC, USA*, pp. 636–641.

Hao, F., Zegura, E. and Ammar, M. (2002). QoS routing for anycast communications: Motivation and an architecture for DiffServ networks, *IEEE Communications Magazine* **40**(6): 48–56.

Herzberg, M., Bye, S. and A.Utano (1995). The hop-limit approach for spare-capacity assignment in survivable networks, *IEEE/ACM Transactions on Networking* **3**(6): 775–784.

Hofmann, M. and Beaumont, L. (2005). *Content Networking: Architecture, Protocols, and Practice*, Morgan Kaufmann, San Francisco, CA.

Holmberg, K. (1995). Lagrangean heuristics for linear cost multicommodity network flow problems, *Technical Report LiTH-MAT-OPT/WP-1995-01*, Department of Mathematics, Linköping Institute of Technology, Linköping.

Holmberg, K. and Yuan, D. (1998). A Lagrangean approach to network design problems, *International Transactions in Operational Research* **5**(6): 529–539.

Holmberg, K. and Yuan, D. (2000). A Lagrangean heuristic based branch-and-bound approach for the capacitated network design problem, *Operations Research* **48**(3): 461–481.

Hou, Y., Yi, S. and Sherali, H. (2006). Optimal base station selection for anycast routing in wireless sensor networks, *IEEE Transactions on Vehicular Technology* **55**(3): 813–821.

Hyytia, E. (2004). Heuristic algorithms for the generalized routing and wavelength assignment problem, *Proceedings of the 17th Nordic Teletraffic Seminar, NTS-17, Fornebu, Norway*, pp. 373–386.

Jain, K., Mahdin, M. and Saberi, A. (2002). A new greedy approach for facility location problems, *Proceedings of the 34th Annual ACM Symposium on Theory of Computing, STOC 2002, Montreal, Canada*, pp. 731–740.

Kar, K., Kodialam, M. and Lakshman, T. (2000). Minimum interference routing of bandwidth guaranteed tunnels with mpls traffic engineering applications, *IEEE Journal on Selected Areas in Communications* **18**(12): 2566–2579.

Kasprzak, A. (2001). *Designing of Wide Area Networks*, Wrocław University of Technology Press, Wrocław.

Kleinrock, L. (1964). *Communication Nets: Stochastic Message Flow and Delay*, McGraw-Hill, New York, NY.

Krishnan, P., Raz, D. and Shavitt, Y. (2000). The cache location problem, *IEEE/ACM Transactions on Networking* **8**(5): 568–582.

Leuf, B. (2002). *Peer to Peer: Collaboration and Sharing over the Internet*, Addison Wesley, Boston, MA.

Li, B., Golin, M., B., Italiano, Deng, X. and Sohraby, K. (1999). On the optimal placement of web proxies in the internet, *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies, INFO-COM 1999, New York, NY, USA*, pp. 1282–1290.

Markowski, M. and Kasprzak, A. (2005). An approximate algorithm for web replica allocation and topology assignment problem in WAN, *Proceedings of the 17th IMACS World Congress, Paris, France*.

Murakami, K. and Kim, H. (1996). Virtual path routing for survivable ATM networks, *IEEE/ACM Transactions on Networking* **4**(2): 22–39.

Paxson, V. (2006). End-to-end routing behavior in the Internet, *SIGCOMM Computer Communication Review* **36**(5): 41–46.

Peng, G. (2004). CDN: Content distribution network, *Technical Report cs-NI-0411069*, Computing Research Repository, http://arxiv.org/abs/cs.NI/0411069.

Perros, H. (2005). *Connection-oriented Networks, SONET/SDH, ATM, MPLS and Optical Networks*, John Wiley and Sons, Ltd, Chichester.

Piro, M. and Medhi, D. (2004). *Routing, Flow, and Capacity Design in Communication and Computer Networks*, Morgan Kaufman Publishers, San Francisco, CA.

Qiu, L., Padmanabhan, V. and Voelker, G. (2001). On the placement of web server replicas, *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2001, Anchorage, AK, USA*, pp. 1587–1596.

Rabinovich, M. (1998). Issues in web content replication, *Data Engineering Bulletin* **24**(4): 21–29.

Rexford, J., Wang, J., Xiao, Z. and Hang, Y. (2002). BGP routing stability of popular destinations, *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement, Marseille, France*, pp. 197–202.

Rosen, E., Viswanathan, A. and Callon, R. (2001). Multiprotocol label switching architecture, *Technical Report RFC 3031*, Internet Engineering Task Force, http://www.ietf.org/.

Ryba, P. and Kasprzak, A. (2006). The gateways location and capacity assignment problem in hierarchical WANs: An approximate algorithm and computational results, *Proceedings of the 18th European Meetings on Cybernetics and Systems Research, Vienna, Austria,* pp. 46–51.

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

Szeto, W., Boutaba, R. and Iraqi, Y. (2002). Dynamic online routing algorithm for MPLS traffic engineering, *in* E. Gregori, M. Conti, A.T. Cambell, G. Omidyar and M. Zukerman, *Networking 2002: Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications*, Lecture Notes in Computer Science, Vol. 2345, Springer-Verlag, Berlin, pp. 936–946.

Tang, M., Jia, W., Wang, H. and Wang, J. (2003). Routing and wavelength assignment for anycast in WDM networks, *Proceedings of the 3rd International Conference on Wireless and Optical Communication WOC, Banff, Canada*, pp. 301–306.

Walkowiak, K. (2004). A new method of primary routes selection for local restoration, *in* N. Mitrou, K. Kontovasilis, G. Rouskas, I. Iliadis and L. Merakos, (Eds), *Networking 2004: Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications*, Lecture Notes in Computer Science, Vol. 3042, Springer-Verlag, Berlin, pp. 1024–1035.

Walkowiak, K. (2005). QoS dynamic routing in content delivery network, *in* R. Boutaba, K. Almeroth, R. Puigjaner, S. Shen and J.P. Black (Eds), *Networking 2005: Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communication Systems*, Lecture Notes in Computer Science, Vol. 3462, Springer-Verlag, Berlin, pp. 1120–1132.

Walkowiak, K. (2006). A new function for optimization of working paths in survivable MPLS networks, *in* A. Levi, E. Savas, H. Yenign, S. Balcisoy and Y. Saygin (Eds), *Computer and Information Sciences—ISCIS 2006*, Lecture Notes in Computer Science, Vol. 4263, Springer-Verlag, Berlin, pp. 424–433.

Walkowiak, K. (2007a). Anycast communication—A new approach to survivability of connection-oriented networks, *in* V. Gorodetsky, I. Kotenko and V. Skormin (Eds), *Computer Network Security*, Communications in Computer and Information Science, Vol. 4236, Springer-Verlag, Berlin, pp. 378–389.

Walkowiak, K. (2007b). Lagrangean heuristic for primary routes assignment in survivable connection-oriented networks, *Computational Optimization and Applications* **40**(2): 119–141.

Walkowiak, K. (2007c). Survivable routing of unicast and anycast flows in MPLS networks, *Proceedings of the 3rd EURO-NGI Conference on Next Generation Internet Networks, Trondheim, Norway*, pp. 72–79.

Walkowiak, K. (2008). A flow deviation algorithm for joint optimization of unicast and anycast flows in connection-oriented networks, *in* O. Gervasi, B. Murgante, A. Laganà, D. Taniar, Y. Mun and M. Gavrilova, *Computational Science and Its Applications—ICCSA 2008*, Lecture Notes in Computer Science, Vol. 5073, Springer-Verlag, Berlin, pp. 797–807.

Woźniak, M., Kurzyński, M. and Puchała, E. (1999). Intelligent internet databases for family doctor practise, *Medical and Biological Engineering and Computing* **37**(2): 1410–1411.

Krzysztof Walkowiak received his M.Sc. Ph.D. and D.Sc (habilitation) degrees in computer science from the Wrocław University of Technology in 1997, 2000 and 2008, respectively. Since 2009 he has been an associate professor at the Chair of Systems and Computer Networks, Faculty of Electronics, Wrocław University of Technology. His research interest is mainly focused on the optimization of connection-oriented networks, survivability issues of MPLS, optimization of grids and P2P systems, and the application of soft-optimization techniques for the design of computer networks.