Lagrangean Heuristic for Anycast Flow Assignment in Connection-Oriented Networks

Krzysztof Walkowiak

Chair of Systems and Computer Networks, Faculty of Electronics,
Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
Krzysztof.Walkowiak@pwr.wroc.pl

Abstract. In this work we address the problem of anycast flow assignment. *Anycast* is a one-to-one-of-many delivery technique that allows a client to choose a content server of a set of replicated servers. We formulate an optimization problem of anycast flows assignment in a connection-oriented network, which is 0/1 and NP-complete. Thus, we propose a new effective heuristic algorithm based on Lagrangean relaxation technique. To our best survey, this is the first study that applies the Lagrangean relaxation to anycast flow problem. We evaluate the performance of the proposed scheme by making a comparison with its counterpart using a sample network topology and different scenarios of traffic demand patterns and replica location. Obtained results show advantage of the Lagrangean heuristic over a previously proposed algorithm.

1 Introduction

The influence of the Internet can be noticed in many areas of people's life, and in the future its importance will grow. Currently, most of transmissions issued in the Internet are unicast – two individual systems connect *one-to-one* to exchange data. However, new approaches are developed to overcome problems now being encountered in computer networks, e.g. network latency, congestion, growing costs. One of the most interesting approaches that can facilitate the development of new services in the Internet and enable QoS (Quality of Service) guarantees is the *anycast* paradigm. Anycast is a *one-to-one-of-many* technique to deliver a packet to one of many hosts. Significance of anycast transmission will grow with popularity of the Internet and development of new services, e.g. distribution of electronic music, movies, books. One of the most famous technologies that applies anycast traffic is Content Delivery Network (CDN). CDN uses many content servers offering the same content replicated in various locations. User-perceived latency and the other QoS parameters can be easily and inexpensively improved by the CDN.

In this work we concentrate on problems of CDN design in MPLS (Multiprotocol Label Switching) environment. The MPLS proposed by the Internet Engineering Task Force (IETF) in [8] is a networking technology that enables traffic engineering and QoS performance for carrier networks. MPLS is a connection-oriented (c-o) technique and is widely applied in backbone networks. Thus, MPLS should be addressed in designing of CDNs. The considered problem can be formulated as follows:

<i>Given</i>	network topology, traffic demand pattern, location of replica servers, link capacity
<i>Minimize</i>	network delay
<i>Over</i>	selection of replica server, routing (path assignment)
<i>Subject to</i>	connection-oriented flow constraints, capacity constraints

The considered problem was formulated and solved in [9]. In this paper we propose a Lagrangean relaxation method combined with subgradient optimization to iteratively solve the problem by a heuristic algorithm ANBFD [9] using as initial solutions results given solving dual problems. Lagrangean relaxation with subgradient optimization has been used successfully for solving various communication network routing problems [3], [7]. Our starting point is the algorithm proposed in [3]. However, we modified this approach according to anycast flow constraints.

Due to limited size of the paper we cannot present more information on anycast flow, CDN and networks optimization. For more information refer to [1], [4-6], [7-9].

2 Anycast Flow Assignment Problem

Since we consider a connection-oriented network, we model the network flow as non-bifurcated multicommodity flow. In c-o networks an anycast demand must consist of two connections: one from the client to the server (upstream) and the second one in the opposite direction (downstream). Upstream connection is used to send user's requests. Downstream connection carries requested data. Consequently, each anycast demand is defined by a following triple: client node, upstream bandwidth requirement and downstream bandwidth requirement. In contrast, a unicast demand is defined by a following triple: origin node, destination node and bandwidth requirement. To establish a unicast demand a path satisfying requested bandwidth and connecting origin and destination nodes must be calculated. Optimization of anycast demands is more complex. The first step is the server selection process. Next, when the server node is selected, both paths: upstream and downstream can be calculated analogously to unicast approach. However, the main constraint is that both connections associated with a particular anycast demand must connect the same pair of network nodes.

To mathematically represent the problem, we introduce the following notations:

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: downstream or upstream.
Π_p	the index set of candidate routes (paths) π_p^k for connection p . For upstream connection paths are between client node and a node hosting a content server. Analogously, downstream candidate paths connect a server and a client node.
X_r	set of route selection variables x_p^k , which are equal to one. X_r determines the unique set of currently selected routes.
c_a	capacity of arc a .
Q_p	bandwidth requirement of connection p .
$\tau(p)$	index of the connection associated with connection p . If p is a downstream connection $\tau(p)$ must be an upstream connection and vice versa.

δ_{pa}^k binary variable, which is 1 if arc a belongs to route π_p^k and is 0 otherwise.

$o(\pi)$ origin node of route π .

$d(\pi)$ destination node of route π .

T total arrive rate of messages in the network

The anycast non-bifurcated flow assignment (AFA1) problem is as follows

$$\min_{X_r} \frac{1}{T} \sum_{a \in A} \frac{f_{ar}}{(c_a - f_{ar})} \quad (1)$$

subject to

$$f_{ar} = \sum_{a \in P} \sum_{k \in \Pi_p} \delta_{pa}^k x_p^k Q_p \quad \forall a \in A \quad (2)$$

$$\sum_{k \in \Pi_p} x_p^k = 1 \quad \forall p \in P \quad (3)$$

$$0 \leq f_{ar} \leq c_a \quad \forall a \in A \quad (4)$$

$$\sum_{k \in \Pi_p} x_p^k d(\pi_p^k) = \sum_{k \in \Pi_{\tau(p)}} x_{\tau(p)}^k o(\pi_{\tau(p)}^k) \quad \forall p \in P \quad (5)$$

$$x_p^k \in \{0,1\} \quad \forall p \in P; k \in \Pi_p \quad (6)$$

$$X_r = \left(\bigcup_{p, k: x_p^k = 1} \{x_p^k\} \right) \quad (7)$$

Variable r denotes the index of set X_r including information on currently selected routes given by variables x equal to 1. The goal is to minimize the network delay [2], [7]. Condition (3) states that each connection can use only one route. (4) is a capacity constraint. Equation (5) guarantees that two routes associated with the same anycast demand connect the same pair of nodes. Constraint (6) ensures that decision variables are binary ones. Note that the constraints set doesn't include a constraint guaranteeing that upstream (downstream) connection ends (starts) in a node with a replica server. This follows from the construct of set Π_p that contains only feasible candidate routes, i.e. each route connects the client node and a replica node. The AFA problem given by (1-7) is NP-complete. This follows from the fact that the unicast non-bifurcated flow assignment is NP-complete [8]. AFA is 0/1 optimization problem. Complexity of AFA problem is a result of huge solution space.

Accordingly to [3] we transform the AFA1 problem into an equivalent formulation called AFA2, which is better suited for a Lagrangean relaxation procedure. Since the objective function (1) is not decreasing with f_a , we can replace equality (2) with an inequality leading to the following problem

$$\varpi = \min_{x_r} \frac{1}{T} \sum_{a \in A} \frac{f_{ar}}{(c_a - f_{ar})} \quad (8)$$

subject to (3-7) and

$$\sum_{p \in P} \sum_{k \in \Pi_p} \delta_{pa}^k x_p^k Q_p \leq f_a \quad \forall a \in A \quad (9)$$

3 Lagrangian Relaxation

The major idea of the Lagrangian Relaxation (LR) decomposition algorithm is to consider the dual problem of by relaxing constraints to obtain a simpler subproblem iteratively to drive towards the optimal solution of the original problem. Therefore, a suitable constraint set is chosen to be relaxed. In our case we use the approach proposed in [3] and relax the constraint (9) using a vector of positive Lagrangian multipliers $\lambda_a, a \in A$. Consequently, the following Lagrangian relaxation of problem AFA2, called LR_AFA is formulated as

$$\varphi(\lambda) = \min \left(\frac{1}{T} \sum_{a \in A} \frac{f_{ar}}{(c_a - f_{ar})} + \sum_{a \in A} \lambda_a \left(\sum_{p \in P} \sum_{k \in \Pi_p} \delta_{pa}^k x_p^k Q_p - f_{ar} \right) \right) \quad (10)$$

subject to (3-6).

The set of feasible solutions for the problem AFA is a subset of the set of feasible solutions for the Lagrangian relaxation of AFA. Since we assume that

$$0 \leq \lambda_a \quad \forall a \in A \quad (11)$$

in any feasible solution of AFA, the term $\sum_{a \in A} \lambda_a \left(\sum_{p \in P} \sum_{k \in \Pi_p} \delta_{pa}^k x_p^k Q_p - f_{ar} \right)$ is non-

positive and thus, the value of the objective function (10) never exceeds the value of the objective function in problem AFA. Thus, accordingly to Lagrangian relaxation theory, whenever problem AFA has a feasible solution, $\varphi(\lambda) \leq \varpi$. Consequently, for each vector of multipliers λ , $\varphi(\lambda)$ is a lower bound of delay function. The best lower bound can be obtained for the vector of multipliers λ^* for which $\varphi(\lambda^*) = \max_{\lambda} \varphi(\lambda)$.

The objective function of Lagrangian problem can be rewritten as

$$\varphi(\lambda) = \min \left\{ \frac{1}{T} \sum_{a \in A} \frac{f_{ar}}{(c_a - f_{ar})} - \lambda_a f_{ar} \right\} + \left\{ \sum_{a \in A} \lambda_a \left(\sum_{p \in P} \sum_{k \in \Pi_p} \delta_{pa}^k x_p^k Q_p \right) \right\} \quad (12)$$

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

Subproblem 1

$$\varphi_1(\lambda) = \min \left\{ \frac{1}{T} \sum_{a \in A} \frac{f_{ar}}{(c_a - f_{ar})} - \lambda_a f_{ar} \right\} \quad (13)$$

subject to (4).

Subproblem 2

$$\varphi_2(\lambda) = \min \left\{ \sum_{a \in A} \lambda_a \left(\sum_{p \in P} \sum_{k \in \Pi_p} \delta_{pa}^k x_p^k Q_p \right) \right\} \quad (14)$$

subject to (3) and (5-6).

According to formulation of objective function we can separate subproblem 1 into $m=|A|$ subproblems and each of such subproblems is solved by

$$f_{ar} = \begin{cases} (c_a - \sqrt{c_a / (T \cdot \lambda_a)}) & \text{when } \lambda_a > 1/(T \cdot c_a) \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

Subproblem 2 can be separated into $q=|P|$ subproblems, one for each connection

$$\varphi_2^p = \min \left\{ \sum_{k \in \Pi_p} x_p^k \alpha_p^k \right\} \quad (16)$$

subject to (3), (5-6) and

$$\alpha_p^k = \sum_{a \in A} \lambda_a \delta_{pa}^k Q_p \quad (17)$$

α_p^k denotes the length of route $k \in \Pi_p$. The minimum for each connection p is attained when for the shortest route calculated under a metric $\lambda_a Q_p$. It means that the subproblem 2 consists of a number of shortest path problems. Comparing to the unicast version of considered problem addressed in [3], in the anycast flow problem to find a shortest path we must take into account all content servers. Another modification is the coupling between upstream and downstream connections given by (5). Therefore, upstream and downstream connections 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 constraint (5) and have the smallest sum of lengths. Such a procedure guarantees optimal solution of subproblem 2. If the set of candidate routes includes only a subset of all possible routes, we must find a pair of routes for which length is the smallest.

4 Subgradient Search

Given multiplier's vector λ we can solve the Lagrangean relaxation problem by solving two subproblems as presented above. Let $x_p^k(\lambda)$ and $f_{ar}(\lambda)$ be an optimal solution

of Lagrangean relaxation for a fixed vector of multipliers λ . The corresponding sub-gradient of the dual function (10) at λ is given by

$$\gamma_a(\lambda) = \left(\sum_{p \in P} \sum_{k \in \Pi_p} \delta_{pa}^k x_p^k(\lambda) Q_p \right) - f_{ar}(\lambda) \quad \forall a \in A \quad (18)$$

The multipliers are updated as follows

$$\lambda_a^{i+1} = \max(1, \lambda_a^i + t_i \gamma_a^i) \quad \forall a \in A \quad (19)$$

The step-size, t_i , can be given as [3], [7]

$$t_i = \rho(\bar{\varphi} - \varphi(\lambda^i)) / \|\gamma^i\|^2 \quad (20)$$

$\bar{\varphi}$ denotes an 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 $0 \leq \rho \leq 2$ [7].

Lagrangean Relaxation Heuristic (LRH) Algorithm

Step 0. Select an initial λ^0 , Set $\rho := 2$, $\rho_{\max} := 5$, $\rho_{\min} := 0.005$, $i_{\max} := 100$, $i := 0$, $\rho_{\text{iter}} := 0$, $\varphi^{\text{best}} := -\infty$, $\lambda^* := \lambda^0$. Using a heuristic, compute an overestimate $\bar{\varphi}$ of $\varphi(\lambda^*)$ or set $\bar{\varphi}$ to an arbitrary large value.

Step 1. Set $i := i + 1$, $\rho_{\text{iter}} := \rho_{\text{iter}} + 1$. Given λ^i , solve the LR_AFA as decoupled subproblems to obtain $\phi(\lambda^i)$, x^i , f^i .

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

Step 2b. Use x^i, f^i to compute feasible primal objective ϖ of the problem AFA2 using ANBFD[9] algorithm. If $\varpi < \varphi^{\text{best}}$ then $\varphi^{\text{best}} := \varpi$, $x^{\text{best}} := x^i$, $f^{\text{best}} := f^i$, $\bar{\varphi} := \varphi^{\text{best}}$.

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

Step 3. Use decoupled solutions x^i, f^i to compute: subgradient $\gamma(\lambda^i)$ (refer to (18)); step size t_i (refer to (19)); multipliers λ^{i+1} (refer to (20)).

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

5 Results

We now describe our simulation setup and scenarios. Algorithms LRH and ANBFD were coded in C++. Next, we run a number of tests to evaluate effectiveness of this algorithm for various simulation scenarios. The network on which we conduct our experiment consists of 36 nodes and 144 directed links [9]. We run experiments for various scenarios of replica location and number of replicas. According to observations of real networks and user behavior, more data is received by clients than is sent to replicas. Therefore, we assume that upstream bandwidth is 0.1 of the downstream bandwidth. The total demand is calculated as a sum of all downstream and all upstream bandwidth requirements. We assume that there are 10 anycast demands for each node in the network. Therefore, the total number of anycast demands is $10n = 360$. Bandwidth requirements of demands are generated randomly.

Fig. 1. Percentage difference between LRH and ANBFD for various replica location

In Table 1 we present results of algorithms LRH and ANBFD for the following four locations of replica servers: (5,23), (5,23,30), and (5,9,23,30). Fig. 1 shows the percentage difference in results of LRH and ANBFD obtained for various location of 4 replicas. If we compare LRH against ANBFD, we can notice that for relatively low loads both algorithms perform similarly. However, when the volume of all connections increases, LRH yields better results than ANBFD. This follows from the fact that LRH can search larger solution space, while ANBFD have a tendency of quick convergence to local minimum and does not have mechanisms to enlarge the number of tested combinations. Another, quite obvious, observation is that increasing the number of replicas decreases the network delay. In addition, if there are more replicas in the network, demands with higher bandwidth requirements can be satisfied.

Table 1. Performance of LRH and ANBFD for various replica location and demands

Total Demand	(5,23)		(5,23,30)		(5,9,23,30)	
	ANBFD	LRH	ANBFD	LRH	ANBFD	LRH
3933	2.4320	2.4320	1.8383	1.8380	1.3240	1.3240
4781	2.8480	2.8480	1.9786	1.9775	1.4140	1.4139
6125	4.1142	4.1033	2.3472	2.3458	1.5582	1.5582
6956	6.0474	6.0474	2.6890	2.6864	1.6384	1.6384
8359			3.3583	3.3503	1.8801	1.8801
9351			4.1990	4.1854	2.1807	2.1793
10400			6.0389	6.0248	2.5018	2.5013
11708			20.9596	20.7720	3.1085	3.0891
12588					3.6577	3.6331
13332					4.2587	4.2319

In Table 2 we report the decision time of both algorithms. Certainly, LRH is much slower than ANBFD, because in each of 100 iterations LRH must run ANBFD algorithm. However, execution time of ANBFD is not deterministic – it depends on situation in the network. Therefore, for (5,23), (5,23,30) and (5,9,23,30) cases, the execution time of LRH is about 94, 85 and 92 times longer than for ANBFD, respectively.

Table 2. Average decision time of LRH and ANBFD in seconds

Algorithm	(5,23)	(5,23,30)	(5,9,23,30)
ANBFD	0.883	1.188	0.844
LRH	83.035	101.369	77.680

5 Conclusions

In this paper, we have addressed the anycast flow assignment problem in connection-oriented networks. We have developed a new heuristic algorithm based on the Lagrangean-relaxation method combined with subgradient optimization to iteratively solve the problem by a heuristic ANBFD proposed in [9]. Based on experimental results, our conclusion is that LRH outperforms ANBFD, especially for congested networks. The main overhead of LRH is much higher computational cost. However the considered problem is offline and consequently the decision time isn't the most important criterion in our study.

Acknowledgements. This work was supported by a research project of the Polish State Committee for Scientific Research carried out in years 2005-2007.

References

1. Awerbuch, B., Brinkmann, A., Scheideler C.: Anycasting in adversarial systems: routing and admission control. *Lecture Notes in Computer Science, LNCS 2719* (2003), 1153-1168
2. Fratta, L., Gerla, M., Kleinrock, L.: The Flow Deviation Method: An Approach to Store-and-Forward Communication Network Design. *Networks Vol. 3* (1973), 97-133
3. Gavish, B., Huntler, S.: An Algorithm for Optimal Route Selection in SNA Networks. *IEEE Trans. Commun., Vol. COM-31, 10* (1983), pp. 1154-1160
4. Hao, F., Zegura, E., Ammar, M.: QoS routing for anycast communications: motivation and an architecture for DiffServ networks. *IEEE Communication Magazine*, 6 (2002), 48-56
5. Markowski, M., Kasprzak, A.: The web replica allocation and topology assignment problem in wide area networks: algorithms and computational results. *Lectures Notes in Computer Science, Vol. 3483* (2005), 772-781
6. Peng, G.: CDN: Content Distribution Network. sunysb.edu/tr/rpe13.ps.gz, (2003)
7. Pióro, M., Medhi, D.: *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufman Publishers (2004)
8. Rosen, E., Viswanathan, A., Callon, R.: Multiprotocol Label Switching Architecture. *RFC 3031* (2001)
9. Walkowiak, K.: Heuristic Algorithm for Anycast Flow Assignment in Connection-Oriented Networks. *Lectures Notes in Computer Science, Vol. 3516* (2005), 1092-1095