

Some Approaches to Solve a Web Replica Location Problem in MPLS Networks

Krzysztof Walkowiak

*Chair of Systems and Computer Networks, Wrocław University of Technology, Wrocław, Poland
walkowiak@zssk.pwr.wroc.pl*

Abstract: Internet and World Wide Web is now a popular mean of interaction among people, institutions, and corporations from all over the world. In order to overcome problems of Internet growth techniques of caching and replication are developed. In this work we address problems of designing the replica system in MPLS environment. MPLS is a new, evolving connection oriented technique for IP networks. We present and discuss issues of Web replica systems. We formulate a new optimisation problem of replica location and propose some heuristic algorithms to solve this problem.

Key words: WWW replica, MPLS, algorithm

1. INTRODUCTION

Internet is now a common mean of communication among people and institutions from all over the world. The World Wide Web is the mostly popular Internet set of applications. Main reasons of the Web's deployment are: ease of use, the availability of standard tools for creating web documents, software for navigating the Web, quick publication of information. Nonetheless we must point out some problems related to the WWW. The Web underlying infrastructure was not planned to support the demands now being placed on the Internet. The Web is resource intensive and consumes a lot of bandwidth. In order to fulfil these expectations enterprises and service providers are deploying faster WANs and larger servers. However, this solution is almost always economically impracticable,

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35616-7_23](https://doi.org/10.1007/978-0-387-35616-7_23)

W. Cellary et al. (eds.), *Internet Technologies, Applications and Societal Impact*

© IFIP International Federation for Information Processing 2002

costs grow high as compared with the short/medium term benefits; demand is also growing. When users once again overrun the capacity of the infrastructure the process repeats itself.

The most commonly solution applied to overcome mentioned problems is caching of the WWW. Caching is a technique usually planned to bring parts of an overall data set closer to its processing site [1]. A Web cache is an application residing between Web servers and clients [2]. This technique uses the knowledge acquired by several analysis on servers access logs and by looking into Web users behaviour, both individually and as members of an organization, to reduce latency experienced by end users when trying to fetch some documents through their Web-browser. An exhaustive list of bibliography on WWW caching can be found in [3].

Replication can be considered as a kind of caching, however there are some differences. Replication assumes storing of an object at a place that cannot see the object, while caching is storing an object at place that sees the source object. It means, that a cache notices both hit and miss requests, a replica on the contrary notices only hits, since requests to replicated server arrives only if that server is believed to have a replica of the requested object. In the presented sense replica is sometimes called push cache [4]. Replication is viewed also a caching system with only one Web server generating WWW documents, while standard caching must serve a great number of Web servers [5].

In this work we address problems of design of replica system in MPLS (Multiprotocol Label Switching) environment. MPLS is a new, evolving connection oriented technique for the IP networks. Most of previous work in the area of replication and caching considers pure IP networks. However MPLS is becoming a popular solution for backbone networks.

The paper is organized as follows. Section 2 presents essential information on Web server replication issues. In section 3 we formulate a new optimisation problem of replica location in MPLS environment. Section 4 contains heuristic algorithms solving the replica location problem. Last section concludes this work.

2. REPLICATION OF WEB SERVERS

Replication system has many interesting characteristics. Replication reduces user perceived latency of frequently requested data since it eliminates the overhead of cache validation messages. Moreover, highly popular Web documents (electronic papers, stock market information) are delivered to replica sites in advance, and reduce WWW servers load. The replication system copes with failures of the network or Web servers, since

data is replicated in different locations in the network. The replication is under full control of one service provider and such problems like copyright issues, hit metering, data consistency are simpler than for caching. In general, replication overcomes some problems encountered in caching, for instance replication handles non-cacheable dynamic or personalized objects, offers easier management and better scalability [4].

Replicas are located in a static or dynamic method. In the static replica placement system administrator, according to observed access and traffic characteristics decides where replicas should be located. Dynamic replica placement assumes, that the system monitors access to various servers and adapts replica set to the requirements [4].

For the context of this paper we are interested in static replica placement. The main problem of static method is to find effective algorithms for replica location. Some previous authors have developed such algorithms. According to [6] the first work on this subject is [7]. Li *et al.* formulate a problem of proxies location in a tree topology with the objective function of cost associated with the selection of proxies and propose dynamic programming algorithm. The cost can be calculated as the overall latency if the link distance is associated with the cost function. Authors of [5] consider the cache location problem for transparent caches. The objective function is the cost of demands using a cache in selected location. The general problem is NP-complete, therefore Krishnan *et al.* analyse only regular topologies: homogenous line, general line and ring. Qiu *et al.* formulate in [6] the placement of web server replicas problem as an uncapacitated k -median problem connected with the facility location problem. They restrict the maximum number of replicas, but did not restrict the number of requests served by each replica. The objective function is to minimize the total cost of all requests. A greedy algorithm and a super optimal algorithm are developed. Guha *et al.* consider in [8] a generalization of the standard facility problem and introduce the requirement of fault-tolerant mechanisms. Every demand point is served by a number of facilities instead of just one. The cost is a weighted combination of facilities locations. An algorithm using the filtering technique and fractional demands is provided. Authors of [9] present a simple and natural greedy algorithm for the metric uncapacitated facility location problem and k -median problem. Arya *et al.* analyse in [10] a local search heuristics for facility location and k -median problems. The main operation of the proposed algorithm is swap, which consists in closing one facility and opening another, clients of the closed facility are assigned to other facilities. In [11] improved combinatorial approximation algorithms for the uncapacitated facility location and k -median problems are proposed and discussed.

Most of presented above work focus on well-known optimisation problems: the k -facility location problem and the k -median problem. The former problem consists in assignment of clients to k facilities that are to be located in network nodes. The objective is to minimize the total cost including the connection cost of each client and the facility cost. The k -median problem generally differs from the facility problem in one thing: there is no cost for opening facilities. The main problem of both discussed problems is location of k facilities, i.e. selection of k nodes of the network for hosting a facility. The next step, assignment of individual clients to a particular replica is much simpler, since one can select the closest replica in terms of connection cost. Capacity constraints of network links are not considered. However in capacitated version of facility location problem there is a capacity constraint on load served by each facility.

We propose a different approach that is much closer to problems encountered in real computer networks. In this work we formulate a new replica location problem. The main difference with previous work is that we take into account capacity constraints on each link of the network. In many cases networks are congested, therefore the capacity resources must be used in an effective manner. Furthermore we consider an MPLS network that is a connection oriented network, i.e. the flow is modelled as a non-bifurcated multicommodity flow. Most of the work in the field of replica placement considers IP networks using multicommodity bifurcated flow. For a good survey on the topic of MPLS please refer to [12,13,14,15,16,17].

3. THE PROBLEM OF WEB REPLICA PLACEMENT IN MPLS ENVIRONMENT

Optimisation of the Web replica placement is a difficult task. In many real life cases, replica or proxies are placed in fairly obvious nodes, e.g. Internet service provider gateway [7]. However, in order to optimise network parameters some algorithms must be applied to provide optimal or sub-optimal solutions. In this section we model the MPLS network flow as non-bifurcated multicommodity flow. However results of this work can be also applied in connection-less networks. Most of previous work in the subject of replica placement presented in section 2 [5,6,8] considers the problem of cache or replica location in a network without taking into account capacity constraint and use IP bifurcated multicommodity flow. As an objective function generally the cost function is used. We propose to extend the replica location problem and include to the model capacity constraint as well as non-bifurcated multicommodity flow.

To mathematically represent the problem, we introduce the following notations. Network is modelled as a directed graph $G=(N,L,C)$ where N is a set consisting of n nodes, L is a set of l links and C is a vector of link capacity, where c_i denotes capacity of link i . Set R denotes r replicas to be placed in network nodes. Let y_i denote a binary decision variable which is equal to one if a replica is located in the node i and is equal to zero otherwise. Let Y_r be a set of variables y_i equal to one. The set Y_r is called a selection. Each selection determines the unique assignment of replicas to network nodes.

In MPLS network packets can be aggregated using FECs (Forwarding Equivalence Classes). We assume, that flow from the replica to a set of clients connected to one node can be aggregated to one LSP. Moreover, we make an assumption, that traffic between clients and replicas is generally asymmetric, since more data is received than is sent to replicas. Therefore many clients connected to one node we model as a one client with aggregated traffic. Let P be a set of p such clients that must be assigned to replicas. Q_i denotes estimated bandwidth requirement for client i . For each client i we are given a set $\Pi_i = \{\pi_i^k : k = 1, \dots, l_i\}$ containing proposals of routes from various replicas to the node of client i . In order to reduce size of the problem only subsets Π_i of all possible routes can be considered. Such subsets can be constructed according to various criteria associated with routing protocols or the network structure. Let x_i^k be a binary decision variable equal to one if the client i uses the route π_i^k and is equal to zero otherwise. Let X_r denote the set of variables x_i^k equal to one. The set X_r is called a selection. Each selection determines the unique set of routes between clients and replicas. A binary variable a_{ij}^k is equal to one if the j -th link belongs the route π_i^k and is equal to zero otherwise. Let $u_j(\pi_i^k)$ be a binary function that equals to one if the source node of the route π_i^k is node j . With each link i we associate a cost d_i that can be also interpreted as link length or latency.

The optimisation model of replica location can be formulated as follows

$$\min_{X_r, Y_r} D(X_r, Y_r) = \sum_{i=1}^l d_i f_{ir} \quad (1)$$

subject to

$$f_{jr} = \sum_{i=1}^p \sum_{k=1}^{l_i} a_{ij}^k x_i^k Q_i \quad \text{for each } j = 1, \dots, l \quad (2)$$

$$\sum_{i=1}^l y_i = r \quad (3)$$

$$\sum_{k=1}^{l_i} x_i^k = 1 \quad \text{for each } i = 1, \dots, p \quad (4)$$

$$f_{jr} \leq c_j \quad \text{for each } j = 1, \dots, l \quad (5)$$

$$\sum_{j=1}^n \sum_{k=1}^{l_i} x_i^k y_j u_j (\pi_i^k) = 1 \quad \text{for each } i = 1, \dots, p \quad (6)$$

$$y_i \in \{0,1\} \quad \text{for each } i = 1, \dots, l \quad (7)$$

$$x_i^k \in \{0,1\} \quad \text{for each } i = 1, \dots, p; k = 1, \dots, l_i \quad (8)$$

The objective function (1) is a cost of transmitting the data from replicas to clients. As mentioned above, according to definition of d_i , the objective function may represent cost, network latency or other metric. Note, that if for all links d_i is equal to 1, the function (1) denotes the overall network flow. Condition (2) is a definition of a link flow. Constraint (3) determines that the number of established replicas must be equal to the defined number of replicas. Constraint (4) ensures that each client uses only one route. Condition (5) is a capacity constraint. Constraint (6) guarantees that each selected route starts in a node with replica. Finally constraints (7-8) ensure that decision variables are binary ones.

Note, that in model (1-8) we do not constrain the amount of service that can be provided at any replica. According to [6], it is a reasonable assumption, since increasing the number of replica sites is much more difficult than increasing the capacity of a replica. The number of replicas is frequently given a priori due to cost and administrative reasons, while the capacity constraint can be overcome by inserting more machines. Moreover we disregard the cost of placing replicas because in many cases the replication traffic and cost of replicas managing can be ignored. We also suppose, that the replication flow connected with delivering the information from the source Web server to the replicas can be neglected, since it is much lower than the flow of replicas clients.

Main goal of this work is to propose heuristic algorithms to be used for optimisation of replica location problem. One of the most important advantage of the replication system is providing reduction of the overall traffic in the network that improves network survivability. Therefore we assume that for all links cost d_i is equal to 1. It means that we want to minimise the overall network flow.

The modified problem looks as follows

$$\min_{X_r, Y_r} D(X_r, Y_r) = \sum_{i=1}^l f_{ir} \quad (9)$$

subject to (2-8).

4. A HEURISTIC ALGORITHM FOR PROBLEM OF WEB REPLICA LOCATION

In this section we propose and discuss three heuristic algorithms solving the problem (2-9). The first algorithm is called Two-Phase (Fig. 1). In the step 1 we evaluate each of n potential locations individually to decide its suitability for hosting a replica. Next we select r best locations and we place one replica in each of them. The step 2 consists in assignment of each client i to one replica according to selected criterion. We can use one of π_i^k routes, however the final solution must be feasible, i.e. the capacity constraint must be satisfied.

Algorithm Two-Phases

1. Evaluate all of the potential nodes for placing a replica, select r best nodes and locate there replicas.
2. For each client i evaluate all replicas, choose the best one and assign client i to this replica using one of the routes from sets Π_i .

Figure 1. An algorithm Two-Phases given in pseudo-code

Now we propose some methods for evaluation of potential sites to host a replica. The first proposition is to apply the hot spot method [14]. We sort the n potential sites according to the amount of traffic generated within their vicinity. We call this parameter a vicinity load. We can use different radius of vicinity, starting from 0 (the site serves only its own traffic) to the maximum distance between any pair of nodes in the network. Next we place r replicas in the best r nodes in terms of the vicinity load. However in this method we do not consider the capacity constraint that can limit the traffic served by a candidate site.

The second approach uses the maximal flow algorithm. For each potential site j we construct a new graph. We add a new node j' and $(n-1)$ new links connecting the j' node to all other nodes except the j node. All these new links have infinite capacity. Next we assume that the flow of all nodes except the j node is placed in the j' node. Finally we calculate the maximal

flow between nodes j and j' . We repeat this operation for n nodes. The best r nodes are selected to host a replica.

Now we introduce a notion of node capacity. We define this parameter as sum of capacity of all links leaving that node. The node capacity indicates how much traffic can leave that node. In the calculation of the node capacity we can take into account not only links leaving that node, but also links leaving neighbours of the considered node. Note that in the problem (2-9) we have two objectives: minimising the overall network flow and satisfying the capacity constraint. In some cases a potential site for a replica can have large capacity, however in that node's vicinity there is not too much load to be served. On the other hand, some nodes with small capacity can have many clients in the neighbourhood. Therefore we propose to use in the selection of potential sites a combination of two criteria: node capacity and vicinity load.

The next proposition for the evaluation of potential sites to host a replica uses the penalty function approach. In order to introduce the penalty function we define a new function as follows

$$pgt(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases} \quad (10)$$

Now we define the modified objective function with the incorporated penalty method

$$\min_{X_r, Y_r} D(X_r, Y_r) = \sum_{i=1}^l \left(f_{ir} + \left(pgt(f_{ir} - c_i) \right)^2 \right) \quad (11)$$

Note that we apply as a penalty function the square of the capacity exceeding in the considered link. It means that if flow of link i exceeds capacity of that link, we add to the objective function the square of the differences between flow and capacity. With the modified objective function we can restrict the set of constraints (2-8) and ignore the capacity constraint (5). For the relaxed problem we propose to use the following approach close to the hot spot method discussed above. First we assign to a considered node all clients. We use routes from the given set of routes. Next we calculate flow of each link and then we compute the cost of a node using the penalty function. Also other algorithms for the k -median problem with the objective function given by (11) may be applied to solve that problem. However we must modified these methods, since in our problem we use virtual connections.

Finally, the random algorithm [6] can be applied in order to obtain r potential sites to host a replica. This method is however unconscious of

client workload and link capacity. To improve the performance we execute the random selection of replicas a number of times.

The second part of the algorithm Two-Phases consists in assignment of all clients to replicas. Once more, a few methods can be applied to solve this problem. Now we present and discuss some of them. Since all replicas are already located, for each client i we must select one route from the set Π_i to one of replicas. According to the approaches proposed above to locate replicas we can sort replicas in order of their costs or suitability.

The first method consists in determining for each client i the closest replica in terms of hop number using one of the routes from the Π_i set. We make a ranking of replicas according to the method used to locate replicas, and we start from the best one. We try to assign to the considered replica as much as possible of clients for which that replica is the closest. We use only routes from sets Π_i and remember about feasibility of the solution. We repeat this process for all replicas. In order to improve the solution we propose to do again the algorithm with another order of replicas.

The second algorithm for assignment of clients is very close to the previous one. Once again we make a ranking of replicas according to the method used to locate replicas, and we start from the best one. We assign to the considered replica as much as possible of clients. We use only routes from sets Π_i and remember about feasibility of the solution. Next we take another replica and repeat the process of assignment, but we can change assignment of already processed clients. We continue the iterations, until all replicas are processed and all clients are assigned.

The last approach uses a modification of the FD (Flow Deviation) algorithm [18] for non-bifurcated multicommodity flow proposed in [19] for assignment of virtual paths in survivable ATM networks. As a link metric we use the derivative of the penalty included objective function (11). The metric of link i is defined as follows

$$l_{ir} = 1 + 2 \text{pgt}(f_{ir} - c_i) \quad (11)$$

Notice, that if flow of the link i do not exceed capacity of that link, metric of link i is equal to 1. The key idea of the FD algorithm is to deviate to flow in repeated evaluations in order to obtain the best solution.

Algorithm Down-Top

1. Set $k:=1$.
2. Evaluate each of the remaining potential nodes for placing a replica, select the best one and locate there a new replica denoted as j .

3. Find all clients i for which the replica j is the closest. Next assign to the replica j as much as possible of these clients using routes from sets Π_i . Set $k:=k+1$.
4. Repeat steps 2 and 3 until $k=r$.

Figure 2. An algorithm Down-Top given in pseudo-code

Next algorithm called Top-Down (Fig. 2) is similar to the greedy method proposed in [14]. We select one replica at a time. In step 2 of the first iteration we evaluate each of n potential locations individually to decide its fitness for hosting a replica. Next, in step 3 we find all clients for which the new replica is the closest. In first iteration, it will be all clients. After that we try to assign to a new replica as much as possible clients using routes from sets Π_i taking into account the capacity constraint. In the second iteration we evaluate each of remaining $(n-1)$ potential locations individually to decide its fitness for hosting a replica together with the already located replica. We assume that clients are assigned to the nearest replica is such a way that the capacity constraint is satisfied. We continue the iterations until we have selected r replicas.

Next proposed algorithm called Top-Down (Figure 3) starts with selection of location of $k>r$ replicas. In this phase we can select replicas according to rankings presented above. Next we assign clients to nearest replicas in order to minimize the overall flow of the network. For each client i we choose one route π_i^k from the Π_i subject to capacity constraint. Since there are too many replicas, $(k-r)$ of replicas must be removed. In step 2 we remove one by one selected replica. We propose to choose replicas according to the load served by the replica and remove a replica with the smallest value of the load. Next we must reassign each client i of the removed replica to new replicas using one of routes of the Π_i . The distance between the client and the replica candidate can be used a criterion.

Algorithm Top-Down

1. Locate $k>r$ replicas in different network nodes. Assign each client i to one of k replicas using a route from set Π_i according to selected criterion.
2. Choose a replica j according to selected criterion and remove this replica. Assign one by one clients of replica j to one of $(k-1)$ remaining replicas. Set $k:=k-1$. Repeat step 2 until $k=r$.

Figure 3. An algorithm Top-Down given in pseudo-code

5. CONCLUDING REMARKS

In this work we have concentrated on the Web replica location problem. We focused on connection oriented environment based on the MPLS. We have described main issues of the replication. We have formulated a new optimisation problem of replica location and clients' assignment. We have proposed and discussed a wide range of heuristic algorithms to solve this problem. In future work we want to make serious experimental testes in order to evaluate proposed heuristics and apply these algorithms to evaluate various policies of replication system design. Furthermore we plan to develop exact algorithms solving presented problem.

REFERENCES

- [1] Baentsch M., Baum L., Molter G., Rothkugel S., Sturm P. World Wide Web Caching: The Application-Level View of the Internet. *IEEE Comm. Magazine*, June 1997, pp. 170-178.
- [2] Vakali A. An evolutionary scheme for Web Replication and Caching. 4th International Web Caching Workshop, San Diego, USA, March 31-April 2, 1999.
- [3] Wang J. A survey of web caching schemes for the internet. *ACM Computer Communication Review*, October 1999, pp. 36-46.
- [4] Rabinovich M., Issues in Web Content Replication. *Data Engineering Bulletin*, Invited paper, Vol. 21, No. 4, December 1998.
- [5] Krishnan P., Raz D., and Shavitt Y. The Cache Location Problem. *IEEE/ACM Trans. Networking*, Vol. 8, October 2000, pp.568-582.
- [6] Qiu L., Padmanabhan V., Voelker G. On the Placement of Web Server Replicas. *Proceedings of IEEE Infocom 2001*, Anchorage, AK, USA.
- [7] Li B., Golin M., Italiano G., Deng X., and Sohrawy K. On The Optimal Placement of Web Proxies in theInternet, in *Proc. of INFOCOM99*, New York, March 1999.
- [8] Guha S., Meyerson A., and Munagala K. Improved Approximation Algorithms for Fault-tolerant Facility Location. To appear in *ACM-SIAM Symposium on Discrete Algorithms*, 2001.
- [9] Jain K., Mahdin M, Saberi A. A new greedy approach for facility location problems. <http://www-math.mit.edu/~mahdian/newalg.ps>.
- [10] Arya V., Garg N., Khandekar R., Meyerson A. Local Search Heuritics for k-median and Facility Location Problems. <http://theory.stanford.edu/~kamesh/lsearch.ps>.
- [11] Charikar M., Guha S. Improved Combinatorial Algorithms for the Facility Location and k-Median Problems. http://theory.stanford.edu/~sudipto/mypapers/kemdian_2.ps.
- [12] Armitage G. MPLS: The Magic Behind the Myths. *IEEE Comm. Magazine*, January 2000, pp. 124-131.
- [13] Awduche D. MPLS and Traffic Engineering in IP Networks. *IEEE Comm. Magazine*, December 1999, pp. 42-47.
- [14] Chen T., Oh T. Reliable Services in MPLS. *IEEE Comm. Magazine*, December 1999, pp. 58-62.

- [15] Lawrence J. Designing Multiprotocol Label Switching Networks. IEEE Comm. Magazine, July 2001, pp. 134-142.
- [16] Li T. MPLS and the Evolving Internet Architecture. IEEE Comm. Magazine, December 1999, pp. 38-41.
- [17] Rosen E., Viswanathan A., Callon R. Multiprotocol Label Switching Architecture. RFC 3031.
- [18] Fratta L., Gerla M. and Kleinrock L. The Flow Deviation Method: An Approach to Store-and-Forward Communication Network Design. Networks, Vol. 3, pp. 97-133
- [19] Walkowiak K. Algorithms for assignment of virtual paths in survivable ATM networks. Ph.D. Thesis, Sci. Papers Division of Systems and Computer Networks of Wrocław Univ. Technol., Series: Report Preprinty 2/2000 (in Polish).