Optimal Replica Servers Placement for Content Delivery Networks (Invited)

Xiang Li[1], Xuejiao Zhao[1], Sanjay K. Bose[2], Bowen Chen[1], Mingyi Gao[1], Gangxiang Shen[1]*
[1]*School of Electronic and Information Engineering, Soochow University, Suzhou, Jiangsu Province, P. R. China*
[2]*Department of EEE, IIT Guwahati, India*
**Correspondence email: shengx@suda.edu.cn*

Abstract: A node-arc Mixed Integer Linear Programing (MILP) model is formulated for replica server placements and resource allocation in Content Delivery Networks (CNDs) which use anycast routing and multi-path transmissions and an efficient *server-list-growth (SLG) algorithm* for this is also proposed. Simulations indicate that the proposed approaches can balance the content delivery load and provide low transmission latency.
OCIS codes: (060.4250) Networks; (060.4256) Network, network optimization

1. Introduction

Cisco predicts that CDNs will carry about 55% of the global IP traffic in 2018 [1]. It also predicted that the CDN traffic will grow at a compound annual growth rate (CAGR) of 34% from 2013 to 2018. This has led to increasing demands for efficient CDNs. CDNs deploy replica servers nearer to their users for faster access and lower request latency with load balancing between servers. Appropriate placement of these replica servers is very important [2]. We address this design problem both through an MILP model and an efficient heuristic algorithm.

Fig. 1 illustrates an example of a CDN, with one origin server, multiple replica servers, and many end users distributed over the physical topology. The origin server could be a data center hosted by the internet service provider (ISP) which functions as the original source of all the contents. The replica servers are distributed around the network to be closer to the users so as to provide faster access. We assume that a user can be served by one of the multiple servers. This is referred to as the "anycast" strategy and is expected to provide both fast content delivery and reduce the load on the origin server. The content delivery capacity of each server is limited and so is the transmission capacity of each link. In Fig. 1, the links in different colors denote their traffic load intensities, which are divided into three levels, red for heavy load, yellow for medium load, and green for light load. In this example, link 5-7 is under heavy load and may not have enough capacity to transmit content from the original server at node 5 to the users attached to node 7. Thus, through deploying replica servers 4 and 8 and asking contents from them, we can reduce the traffic load on the origin server as well as the content traffic on link 5-7.

For a CDN, it would be crucial to decide as optimally as possible, the placement of the replica servers and the resource allocation for both the servers and the network links, so that the content delivery load at each server is balanced and the average content delivery latency of each user request can be minimized. To solve this problem, we develop both an MILP model and an efficient SLG algorithm. The simulation results indicate that the SLG algorithm is efficient and performs close to the MILP model. The effectiveness of the solution is also verified by comparing with a Monte Carlo simulation to show that the content delivery latency calculated by the SLG algorithm is very close to that actually obtained by the simulation.

2. MILP Model for Server Placement and Resource Allocation

With a CDN as shown in Fig. 1, based on the anycast mode, all the user requests can be served either by the origin server or by one of the replica servers. We assume that all the replica servers have a full cache of the origin server and that the content request intensity of each user node is known in advance. Note that the user node is defined as a network node that aggregates all the user content requests in the associated region. Assuming that the content delivery capacity of each server and the transmission capacity of each link are limited, the objective of the design is to minimize the content delivery load at each server and the average content delivery latency in the overall network by properly placing the replica servers at the appropriate nodes and allocating resources efficiently. For this, we develop an MILP model as follows.

Sets and parameters: A physical network topology $\boldsymbol{G(N,E)}$, where $\boldsymbol{N}$ is the set of nodes and $\boldsymbol{E}$ is the set of (bi-directional) links, $\boldsymbol{N_i}$ is the set of neighboring nodes of node i, t_u is the aggregated user content demand at node u, and L_{ij} is the physical distance of link (i, j). For quality of service (QoS), D denotes the maximal allowed latency for content delivery from a server to a user. M is the maximum transmission capacity of each link in the network. A is the total number of servers deployed in the network. Δ is a big value. c_{fiber} is the light speed in the optical fiber. O denotes the index of the original server.

Variables: R_v is a binary variable that takes the value of 1 if there is a replica server placed at node v and 0 otherwise. λ^{vu} is the total amount of bandwidth for content delivery from server node *v* to user node *u*. λ_{ij}^{vu} is the amount of bandwidth for content delivery from server node *v* to user node *u* that traverses link (*i*, *j*). ρ_{ij}^{vu} is a binary variable which takes the value of 1 if there is any content from server node *v* to user node *u* traversing link (*i*, *j*) and 0 otherwise. λ_{ij} is the total amount of bandwidth for content delivery on link (*i*, *j*). C is the maximum content delivery capacity at each server.

Objective: $\boldsymbol{minimize}\ C + \alpha \sum_{u,v,i\in \boldsymbol{N}, j\in \boldsymbol{N_i}} \left(\lambda_{ij}^{vu} \cdot L_{ij}/c_{fiber}\right)$ (1)

Here, the first term minimizes the maximum content delivery capacity of the servers including the origin and replica servers and the second term minimizes the content delivery latency which is weighted by the content load. α is a weight factor, which is set to be small so as to ensure a higher priority for the first objective.

Subject to:

$$\sum_{j\in \boldsymbol{N_i}} \lambda_{ij}^{vu} - \sum_{j\in \boldsymbol{N_i}} \lambda_{ji}^{vu} = \begin{cases} -\lambda^{vu}, & i = u \\ \lambda^{vu}, & i = v \\ 0, & otherwise \end{cases} \quad \forall i, u, v \in \boldsymbol{N}: u \neq v \quad (2)$$

$$\lambda_{ij}^{vu} = \begin{cases} \lambda_{ji}^{vu} = \lambda^{vu}, & i = j = v \\ 0, & otherwise \end{cases} \quad \forall i, j, u, v \in \boldsymbol{N}: u = v \quad (3)$$

$$\lambda^{vu} \leq R_v \cdot \Delta \qquad \forall u \in \boldsymbol{N}, v \in \boldsymbol{N}: v \neq O \quad (4)$$

$$\sum_{v\in \boldsymbol{N}} \lambda^{vu} \geq t_u \qquad \forall u \in \boldsymbol{N} \quad (5)$$

$$\sum_{u\in \boldsymbol{N}} \lambda^{vu} \leq C \qquad \forall v \in \boldsymbol{N} \quad (6)$$

$$\sum_{u\in \boldsymbol{N}} \sum_{v\in \boldsymbol{N}: u\neq v} \lambda_{ij}^{vu} \leq M \qquad \forall i \in \boldsymbol{N}, \forall j \in \boldsymbol{N_i} \quad (7)$$

$$\sum_{i\in \boldsymbol{N}} \sum_{j\in \boldsymbol{N_i}} \lambda_{ij}^{vu} \cdot L_{ij}/c_{fiber} \leq D \quad \forall i, u, v \in \boldsymbol{N}: u \neq v \quad (8)$$

$$\Delta \cdot \rho_{ij}^{vu} \geq \lambda_{ij}^{vu} \qquad \forall i, u, v \in \boldsymbol{N}: u \neq v, j \in \boldsymbol{N_i} \quad (9)$$

$$\rho_{ij}^{vu} \leq \lambda_{ij}^{vu} \qquad \forall i, u, v \in \boldsymbol{N}: u \neq v, j \in \boldsymbol{N_i} \quad (10)$$

$$\sum_{v\in \boldsymbol{N}: v\neq O} R_v \leq A - 1 \quad (11)$$

Constraints (2) and (3) jointly ensure flow conservation and (3) is for the case where the server node *v* and the user node *u* are collocated at the same node. Constraint (4) indicates that if there is any content delivered from node *v* (excluding the original server node) to node *u*, then *v* must be a replica server. Constraint (5) ensures that all the contents requested at user node *u* are fully served by all the servers. Constraint (6) ensures that all of the user contents delivered by the server at node *v* should not exceed its maximum delivery capacity. Constraint (7) ensures that the total bandwidth for content delivery over link (*i*, *j*) should not exceed its maximum capacity. For the QoS purpose, constraint (8) ensures that the content delivered from server node *v* to user node *u* should not exceed the maximum allowed latency. Constraints (9) and (10) jointly ensure that if there is any content delivered from server node *v* to user node *u* that traverses link (*i*, *j*), ρ_{ij}^{vu} should be 1; otherwise, it should be 0. Constraint (11) ensures that the total number of replica servers should not be greater than $A - 1$.

3. Heuristic Algorithm

Due to the computational complexity of the above MILP model, we also propose an efficient heuristic algorithm, called *server-list-growth (SLG) algorithm*, for replica server placement and network resource allocation as follows.

Step 1: Assume that the origin server s_0 has been fixed by the ISP. We add it to the server list S.

Step 2: Select another node s from the remaining nodes. Use Dijkstra's algorithm to find the shortest route from each user node *i* to each server node in list $S \cup \{s\}$, and find $d_i^s = \min_{k\in S\cup\{s\}} d_{i,k}$, where $d_{i,k}$ is the shortest distance from user node *i* to server node *k*.

Step 3: Repeat Step 2 to consider all the potential new server node s, then choose the one s^* that has a minimum of $\sum_{i\in \boldsymbol{N}} d_i^s$ as the server node to be placed. Add s^* to server node list S.

Step 4: Repeat Steps 2 and 3 until A-1 replica server nodes are placed.

In the above algorithm, the server list grows gradually and each growth ensures the maximum benefit in terms of the reduction of content delivery latency of the whole network. For performance comparison, we also consider a *hot-spot algorithm* [3] and a *zone-optimal algorithm*. For the hot-spot algorithm, the user content requests from each node is sorted according to their volumes from the highest to the lowest and then the first A-1 nodes are chosen as the locations for replica server placement. The key idea of the zone-optimal algorithm is to define a node *n* and all its neighbor nodes as a zone. Then we calculate the total amount of content requests for each zone, which is a sum of the content request of each node contained in the zone. We further sort these zones according to the amounts of their content requests from the highest to the lowest to obtain A-1 zones, and set all the central nodes of these zones as the replica server locations.

With the replica servers placed, we next describe the steps of delivering contents from servers to users.

Step 5: Get a user node u, and choose the nearest server from the server list placed as in Step 4 to satisfy its content request. In order to balance the content delivery load on the servers, we ensure that the maximal content delivery capacity of each server does not exceed $\sum_{u\in \boldsymbol{N}} t_u/A$. The nearest server will satisfy the content request of node u as much as it can; however, if the sum of the delivered content exceeds $\sum_{u\in \boldsymbol{N}} t_u/A$, then choose the next nearer servers

to satisfy the remaining content request.

Note that when each server serves the content requests, multi-paths between the server and a user would be employed to deliver the content if the first shortest route does not have sufficient bandwidth.

4. Simulations

We evaluated the performance of the MILP model and the heuristic algorithms based on the 14-node, 21-link NSFNET network. The location of the origin server is pre-decided, and the replica servers are placed based on different approaches. All the content requests from users are aggregated at the network node that the users are attached to. The number of aggregated content requests at each node is assumed to be random within a range from 100 to *X* units. The network link capacity is set to be 1000 units, implying that at the most 1000 units of content requests may be served simultaneously. The maximum allowed latency *D* of each requested content is set to be 50 ms. The software AMPL/Gurobi was used to find the optimal solution to the MILP model.

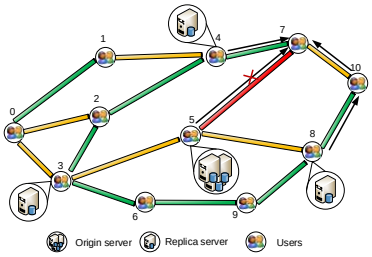

Fig. 1. Example of CDN

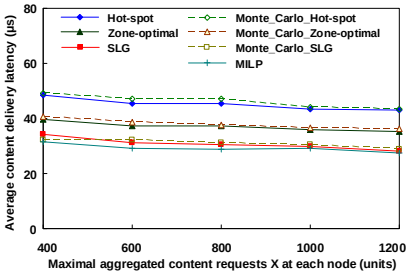

Fig. 2. Content delivery latency under different content request load

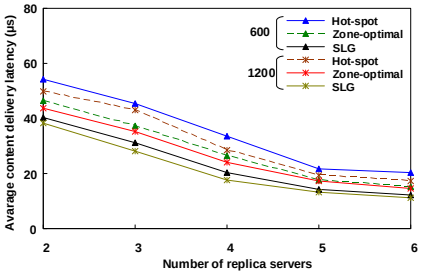

Fig. 3. Content delivery latency under different numbers of replica servers

Since we use the "anycast" strategy to achieve load balance between the servers, the results show that all the approaches including the MILP model and the heuristic algorithms can efficiently achieve the same content delivery load as $\sum_{u\in N} t_u/A$ on each server node. The performance difference among these approaches is only highlighted in the aspect of content delivery latency. Next we analyze this performance difference.

Assume that there are four servers in the network. Fig. 2 shows the average content delivery latency of the overall network for the different replica server placement approaches, which is calculated as $\sum_{u,v,i\epsilon N,j\in N_i}\left(\lambda_{ij}^{vu}\cdot L_{ij}/c_{fiber}\right)/\sum_{u\in N} t_u$. We can see that for the MILP model the average delivery latency keeps flat for different numbers of content requests at each user node. It is also found that the proposed SLG algorithm is very efficient to achieve delivery latency very close to that of the MILP model. It is also efficient to achieve latency about 30% lower than that of the hot-spot algorithm and more than 15% lower than that of the zone-optimal algorithm.

Similarly, Fig. 3 shows how the average content delivery latency changes with an increasing number of replica servers under different intensities of content requests (for *X*=600 and 1200 units). It is reasonable to see that the delivery latency becomes lower when more replica servers are placed. In addition, again we can see that the proposed SLG algorithm is efficient to show the lowest delivery latency among the three heuristic algorithms.

To verify the accuracy of the content delivery latency calculated by the different heuristic algorithms above, we also performed Monte Carlo simulations to generate content requests and simulate the content delivery process. We have simulated 10^6 such content requests and computed the delivery latencies for all the served requests. Fig. 2 also shows the average content delivery latencies obtained by the Monte Carlo simulation for all the heuristic algorithms. We can see that the results of the Monte Carlo simulation and the analytical calculation by the different heuristic algorithms match well, verifying the accuracy of the analytical calculations.

5. Conclusion

We proposed a replica servers placement strategy to distribute contents through replica servers so as to minimize the total content delivery latency and achieve load balance between servers for a CDN. An MILP model and an SLG heuristic algorithm were developed to choose the replica server locations. Simulation results show that the proposed SLG algorithm can achieve load balance between servers and can significantly reduce the average delivery latency of the whole network, performing almost as well as the MILP model.

Acknowledgment: This work was jointly supported by National Natural Science Foundation of China (NSFC) (61322109, 61172057), Natural Science Foundation of Jiangsu Province (BK20130003), and Science and Technology Support Plan of Jiangsu Province (BE2014855).

References

[1] Cisco visual networking index: forecast and methodology, 2013-2018, Jun. 2014.
[2] T. Wauters *et al.*, "Replica placement in ring based content…," *Computer Communications*, vol. 29, no. 16, pp. 3313-3326, Oct. 2006.
[3] L. Qiu *et al.*, "On the placement of web server replicas," in Proc. *INFOCOM 2001*.