Proceedings of Engineering and Technology Innovation, vol. 1, 2015, pp. 40 - 44

An ACO-based Routing Algorithm for Multimedia CDN Networks

Jian-Bo Chen *, Yui-Lin Wang

Department of Information and Telecommunications Engineering, Ming Chuan University, Taoyuan County, Taiwan.

Received 19 March 2015; received in revised form 20 April 2015; accepted 01 May 2015

Abstract

Multimedia Content Delivery Networks is used to distribute multimedia contents from origin server to many replica servers, which can reduce the loads and can improve the availability. The client request must be redirected to the most appropriate one among these replica servers. In this paper, we proposed Ant Colony Optimization (ACO) based routing algorithm to solve this problem. The ants in ACO-based routing algorithm will release pheromone when they go through the path. The routing decision is based on the cumulative pheromone of the path. The ACO-based routing algorithm not only can find the most appropriate path but also can suit for dynamic topology. When the topology changes frequently, the ACO-based routing algorithm will always find a optimized path. The simulation results show that the ACO-based routing algorithm can achieve higher performance than the other mechanisms.

Keywords: content delivery network, ant colony optimization

1. Introduction

Recently, the streaming multimedia contents play an important role in the Internet. The popular multimedia web sites, such as YouTube, have millions of multimedia contents could be accessible from the Internet. These web sites usually do not put all the multimedia contents in a single server or a single place to serve the clients. Instead, they distributed these multimedia contents into some replica servers in other places around the world. In this kind of network architecture, the loading of the single server can be shared by other servers. The architecture is also known as Content Delivery Networks (CDN) architecture. The CDN architecture distributes the multimedia contents from the origin server to many replica servers, which can reduce the loading and can improve the availability. When a client issues a request to specific multimedia content, the CDN architecture will redirect this request to the most appropriate replica server. Then this replica server serves the client request instead of the origin server, which can achieve better performance in the CDN networks. The bandwidth usage can be minimized, the CPU loading of servers can be reduced, and the connection number of specific link can be decreased. In other words, the CDN architecture is especially suitable for multimedia content distribution and can reduce the user perceive latency.

The CDN architecture consists of several parts: origin server, replica servers, accounting system, distribution system, and request routing system [1-6]. The distribution system is used to distribute the multimedia contents from origin server to other replica servers. The accounting system is in charge of the usage of multimedia contents. In this paper, our focus is on the request routing system. The request routing system [7] includes two sub-system, server location system and request routing system. The server location sub-system is used to find the most appropriate replica server. When there are several replica servers contain the requested multimedia content, how to find the most appropriate server is the challenge of server location sub-system. Once the most appropriate replica server is determined, how to redirect the client request to this replica server is another challenge of request routing sub-system.

In this paper, we will use Ant Colony Optimization based Routing Algorithm (ARA) to solve the request routing problem. In the server location sub-system, the most appropriate replica server can be located according to the Pheromone. The more Pheromone the path contains, the better replica server can be determined. In the request routing sub-system, when the client passes through a specific path, the Pheromone will be retained in this path. All the Pheromone will be accumulated in this path. Based on this ARA algorithm, the most appropriate replica server will always be selected when the path contains more and more Pheromone.

The ARA algorithm not only can find the best path but also can adapt with the dynamic routing environment. When the network topology changes or the link congests frequently, the ARA algorithm can dynamically adjust its routing policy based on the amount of pheromone. The ARA algorithm sends a lot of packets as artificial ants to each replica server and stores the amount of pheromone for each path. The routing is decided by the pheromone function and a random value to generate a probability value. The replica servers send back the artificial ants with their own status. When the artificial ants go back to the origin server, the status information of each replica server will be updated. In the streaming multimedia CDN architecture, when a client connects to a replica server and streams a multimedia content, the bandwidth of the path, the loading of the server and the connection number of the server will be changed. If the capacity, such as bandwidth availability, cannot afford to the client request, the ARA algorithm will automatically chose another path to find out the other appropriate replica server. Therefore, the ARA algorithm can enhance the accuracy of server locate and can improve the availability of servers.

This paper is organized as follows. We will introduce the CDN architecture in Section 2 and introduce the ACO in Section 3. The ARA algorithm will be described in Section 4. Our proposed method will be introduced in Section 5. The simulation results are shown in Section 6. Section 7 is the conclusions.

*Corresponding author. Email: jbchen@mail.mcu.edu.tw

2. CDN Architecture

The CDN network is a content caching mechanism with low cost and high reliability. It can increase the efficiency of data transmission and reduce the loading of the origin server. The CDN can also provide service to clients around the world. The client request is served by the closest replica server which contains the requested multimedia contents. In general, the architecture of CDN consists of several parts, including client, replica servers, origin server, accounting system, distribution system and request routing system. The system architecture of CDN is shown in Fig. 1.

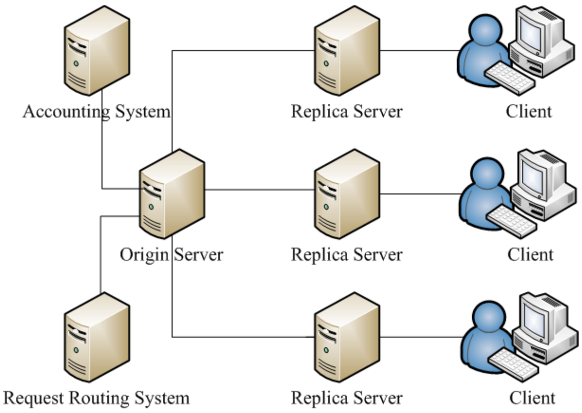

Fig. 1 The CDN system architecture

The request routing system has two important functions, server location and request routing. The function of server location is to determine the most appropriate replica server after receiving a request from client. The most appropriate replica server must have enough ability to serve the client and must have the multimedia content the client request. There are two mechanisms to obtain the capacity of the server [8]. One is server push mechanism [9, 10], which is an active way to collect the capacity of the server. The replica servers send their server capacities to the agent that manages all replica servers at a regular interval. The agents will summarize all the capacities and store them to the database. The other method is agent probe [11], which is a passive mechanism that an agent uses to query the capacities of servers at regular interval.

The processes of CDN include several steps. Figure 2 shows the procedures which is described as follows.

Step 1: The origin server sends an entry of multimedia content information to CDN by request routing system.
Step 2: The origin server transfers the multimedia content to distribution system. Then the distribution system decides which replica servers will store multimedia content.
Step 3: The distribution system transfers the multimedia content to the replica servers.
Step 4: The client sends a request to the request routing system.
Step 5: The request routing system determines the suitable replica server, and redirects the request to the specific replica server.
Step 6: The replica server makes a connection to the client and starts the streaming service.

The design requirements and design constraints are summarized based on the characteristics of the mechanism.

2.1. Design Requirements

(1) It has one degree of freedom, and therefore it has one input.

(2) It is a planar mechanism with four links or more.

(3) It has at least one cam pair to generate a non-uniform output speed.

(4) It has at least one gear pair to change the uniform output speed.

(5) It has a ground link to support or constrain other links.

(6) It has one output link.

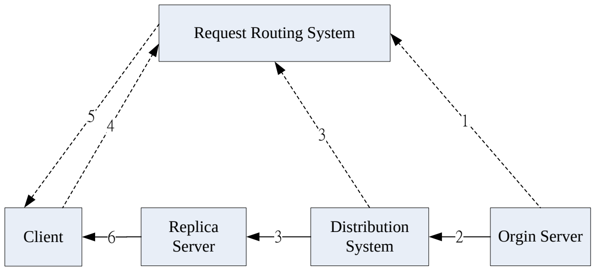

Fig. 2 The CDN operation processes

3. ACO Algorithm

The ACO is a swarm intelligent algorithm [12] that simulates the processes of ants exploring environment and foraging for food. It is used to solve the optimization problem of static allocation. This algorithm was proposed by M. Dorigo in 1992[13].

The ACO uses the pheromone between ants to communicate with each other. When ants reach a node, this refers to that the pheromone level of the node is more than the other node. The node contains higher pheromone level; the ants have higher probability to select it. The ACO uses the feature of positive feedback [14] to solve the optimization problem. It also uses a random function to make the result variable. The characteristics of ACO algorithm includes: (1) Ants tend to choose the path with higher pheromones. (2) For shorter paths, the pheromones cumulative speed is more quickly. (3) Ants communicate with each other indirect by release and stack the pheromones.

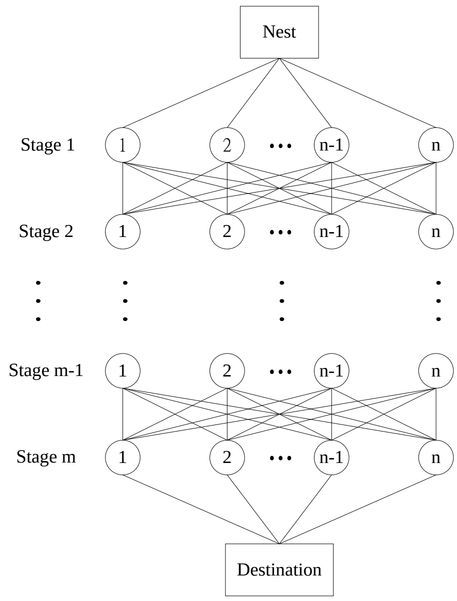

Fig 3. The searching graph of ACO

Figure 3 shows the searching graph of ACO. There are m stages from the nest of ants to destination. Each stage will choose a value from n values by the pheromone function. The pheromone function will refer to the result of previous iterations and a random value to compute a probability value and decide the result of stage. After several iterations, the processes output a set of results with m values.

4. ACO Routing Algorithm

The ant colony optimization routing algorithm (ARA) [15] is an enhanced algorithm from the ant colony optimization algorithm used for routing process. Assume that the source node is treated as the nest of ants, and the destination node is treated as the food.

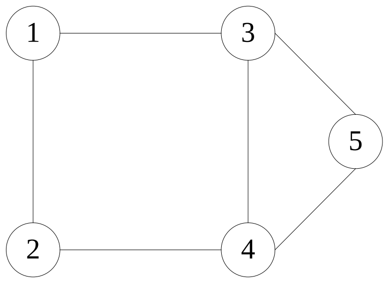

Fig 4. The example of network topology

The simple network topology is shown in Fig. 4. In this example, we assume that node 1 is the nest of the ants and node 5 is the destination node. The ARA algorithm is the processes of how the ants find the path from nest to destination. For instance, when an ant goes from node 1 to node 3, the pheromone in this link will be updated. If more ants pass through the link, then higher pheromone level will be accumulated on this link. As a result, more ants will select this link as the most appropriate link. Table 1 shows an example of the pheromone level for node 3. Each node maintains a table of pheromone level to the destination nodes. This is also known as the routing table. In the routing table of node 3, if the destination is node 2, there are three neighbors which are node 1, node 4, and node 5. The pheromone levels for these three neighbors are 0.4, 0.5, and 0.3. When the ant wants to select next node, it will choose the node with highest pheromone level as the next node. In this example, it is node 4.

Table 1 The pheromone table of node 3

		Neighbor Node		
		1	4	5
Destination Node	1	0.5	0.3	0.2
	2	0.4	0.5	0.3
	4	0.2	0.5	0.3
	5	0.2	0.3	0.5

5. Proposed Method

This paper is focused on the ACO-based routing algorithm for multimedia CDN network. The major objective is to enhance the accuracy of server location and to reduce the computation time in route decision. This algorithm also can solve the redirection failure problem when the network is congested [16]. In CDN network, the request routing system is to select the most appropriate replica server and to redirect this client request to the replica server. Assume that the replica server is chosen, but during the redirection process, the client cannot connect to the replica server. We call this kind of situation as the redirection failure.

We use the positive feedback from ARA algorithm to enhance the accuracy of server location calculation, and ARA algorithm maintains the updated data to increase the availability of servers. The process of ARA algorithm includes six components [17-19]. Figure 5 shows the flow diagram of our proposed method. The processes are Construction of the Network Topology, Initialization, Tour Construction, Data response and Pheromone Update.

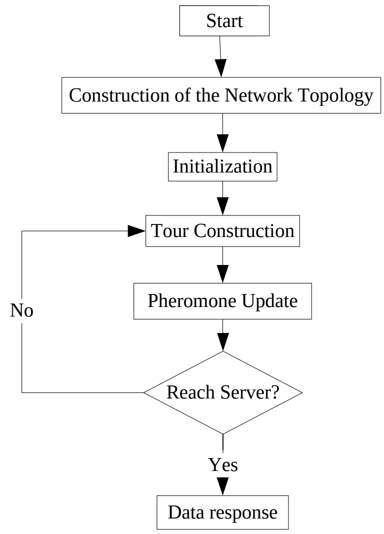

Fig. 5 The Flow Diagram of Propose Method

The first step is to construct the network topology. Before executing the ARA algorithm, the network topology, denote as G (n,e), must be constructed, where n is the set of nodes including origin server and replica servers, and e is the set of links between each node.

After the step of network topology construction, each node must be initialized. That is, setting the initial pheromone levels and weights on each node. The initial pheromone levels and weights are determined by the management policy and corresponding situation. Before this process, the ARA algorithm sends lots of ants to construct the tours and to cumulate the pheromone level on each node. These ants select the forwarding path randomly. The ARA algorithm uses the pheromone level and random value to generate a probability value. When these ants select the next nodes, each node will update its own pheromone table. The pheromone table records the pheromone level of each link that connects to this node. When these ants pass through a node and select a link, the pheromone level of this link will be updated and stored into the pheromone table. However, the pheromone level will count to infinite, so the pheromone levels are decreased over time. When an ant reaches to the replica server, the replica server will send back an ant to the origin server. The back ant carries the current status information of the replica server. The back path is decided by the pheromone level so that this ant passes through the router directly, no calculation are needed. When the back ants reach to the origin server, the origin server updates the information what they carry. To keep the information newest, our proposed algorithm will be run every few minute.

The pheromone is the most important mechanism in ant colony optimization algorithm. It allows the ants able to communicate with each other while they arrive in different time. The prior ants store the results in nodes, and the posterior ants consult the results to make decision. In the beginning of algorithm execution, the distribution of pheromone is scattered. After some iteration, the better path has more pheromone and the pheromone will continually increase. When pheromone level in a path is high enough, almost all ants will select this path and this path is become primary path. In this time, we call that the network is converged. Although almost all ants select the primary path, there are still some ants selecting the different path because the primary path cannot be sure that it is always being the best. These ants try another path and find other feasible path as alternative path. When the primary path is more and more congested, the alternate path will replace the primary path. If a node or path failed, the pheromone is reset to zero immediately. This means that no ants can go through this path. Therefore, the ARA algorithm can adapt to topology change in a short time.

6. Simulation Results

The network topology used in our simulation is shown in Fig. 6. In this topology, one origin server, five replica servers and four routers are included. The replica servers contain some multimedia contents. In this simulation, we duplicate the multimedia contents to replica server randomly. The bandwidth between routers and replica servers is 100Mbps, the bandwidth between routers is 1Gbps, and the bandwidth between origin server and router is 100Mbps. The bit rate and service time for multimedia contents is shown in Table 2. For example, the bit rate (transmission rate) of Content2 is 768Kbps and the service time (content length) is 700 seconds. The origin server sends 20 ants to each replica server every 5 minutes. The update information for each replica server includes a timestamp field. When these ants come back from replica server to origin server, the origin server will update the newest data by the timestamp field. To avoid routing loop, the ant visits all nodes will be dropped. To keep the better path with higher pheromone level, and to ensure another path with lower pheromone level, when an ant passes a node, it will increase 2 pheromone levels and decrease 1 pheromone every 30 second. The client request is distributed by Erlangs distribution. The 10% Erlangs means that the packet loss is 10%. Each kind of network load is tested 10 times and the execution time for each test is 10 minutes.

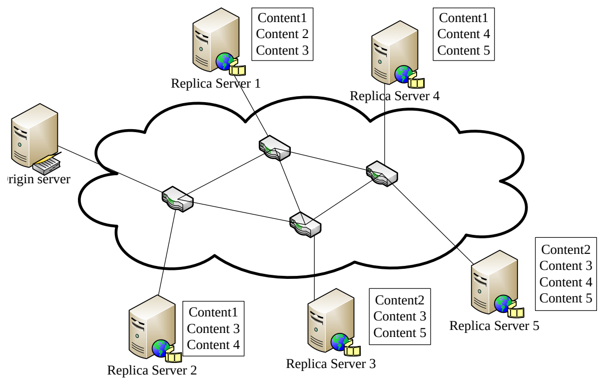

Fig 6. The simulation network topology

Table 2 The information of multimedia contents

	Bit Rate (Kbps)	Service Time (Second)
Content 1	1024	400
Content 2	768	700
Content 3	1024	800
Content 4	512	300
Content 5	768	200

We use SMPL simulation tool in this simulation. SMPL is a set of C language functions. The hit rates of redirection with different schemes are compared in Fig. 7. We compare our proposed ARA algorithm with other two methods, routing table polling and network probing. The results show that our proposed ARA algorithm has better performance in a congested and dynamic network.

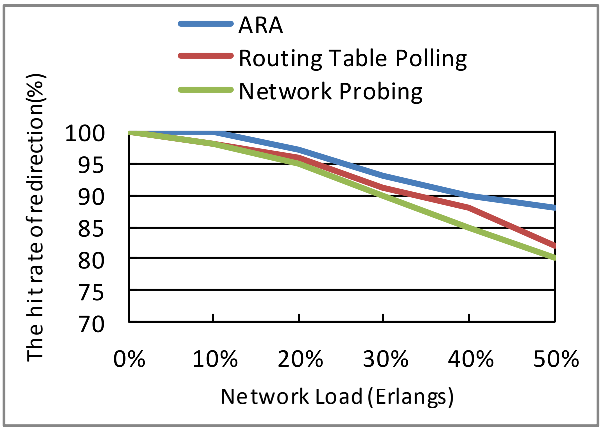

Fig. 7 Comparison of redirection hit rate

7. Conclusion

In this paper, we propose an ACO-based routing algorithm for multimedia CDN network to enhance the accuracy of server location and the availability of CDN network. In this algorithm, numerous artificial ants are sent to replica servers. The status information of replica servers is then sent back to the origin server to build a database for request routing system. We also propose a pheromone function for ants to select the next node in each decision. The ACO-based routing algorithm also has the ability for load balancing that will reduce the overall drop rate. Our simulation results show that the algorithm can work in a congested network or dynamic network

References

[1] Gang Peng, CDN: Content distribution network, Technical Report TR-125, Experimental Computer Systems Lab, Stony Brook University, 2003.

[2] Novella Bartolini, Emiliano Casalicchio, and Salvatore Tucci, “A walk through content delivery networks,” Lecture Notes in Computer Science, vol. 2965, pp. 1-25, 2004.

[3] Turrini E., “An architecture for content distribution internetworking,” Dept. of Computer Science, Univ. of Bologna, Italy, March 2004.

[4] Pathan M. and Buyya R., “A taxonomy and survey of content delivery networks,” Univ. of Melbourne, Dept. of Computer Science and Software Engineering, Technical Report GRIDS-TR-2007-4, Australia, February 2007.

[5] M. Green, B. Cain, G. Tomlinson, and S. Thomas, "CDN peering architectural overview," Network Working Group, Internet-Draft, November 2000.
[6] M. Day, B. Cain, G. Tomlinson, and P. Rzewski, "A model for content internetworking," Network Working Group, Category: Informational, February 2003.
[7] Md. Humayun Kabir, Eric G. Manning, and Gholamali C. Shoja, "Request-routing trends and techniques in content distribution network," Parallel, Networking, Distributed Applications Laboratory, Dept. of Computer Science, Univ. of Victoria, Canada, 2008.
[8] James D. Guyton and Michael F. Schwartz, "Locating nearby copies of replicated internet servers," Proc. of Applications, Technologies, Architectures, and Protocols for Computer Communication, pp. 288-298, August 1995.
[9] David R. Boggs, Internet broadcasting, Ph.D. Thesis, Technical Report CSL-83-3, Xerox Palo Alto Research Center, October 1993.
[10] Craig Partridge, Trevor Mendez, and Walter Milliken, Host Anycasting Service, IETF RFC 1546, November 1993.
[11] E. W. M. Wong and S. Chan, "Modeling of video-on-demand networks with server selection," Proc. Global Telecommunications, IEEE, Nov. 1998, pp. 8-12.
[12] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Swarm intelligence – from natural to artificial systems," Santa Fe Institute Studies in the Sciences of Complexity, vol. 14, pp. 163-164, 2002.
[13] Dorigo, M. and Gambardella L.M., "Ant colony system: a cooperative learning approach to the traveling salesman problem," IEEE Trans. Evolutionary Computation, vol. 1, pp. 53-66, 1997.
[14] Dorigo, M., Maniezzo, V. and Colomi, A., "Positive feedback as a search strategy," Technical Report, no. 91-016, Politecnico di Milano, Italy, 1991.
[15] Mesut G¨unes, Udo Sorges, and Imed Bouazizi, "ARA - the ant-colony based routing algorithm for MANETs," Proc. Parallel Processing Workshops, Vancouver, BC, Canada, August 2002.
[16] Gustavo Sousa Pavani and Helio Waldman, "Traffic engineering and restoration in optical packet switching networks by means of ant colony optimization," Proc. Broadband Communications, Networks and Systems, San Jose, California, October 2006.
[17] Eslam Al Maghayreh, "Salam Abu Al-Haija, Faisal Alkhateeb, and Shadi Aljawarneh, bees ants based routing algorithm," Proc. Intelligent Systems, Modelling and Simulation, Liverpool, UK, January 2010.
[18] Dongming Zhao, Liang Luo, and Kai Zhang, "An improved ant colony optimization for the communication network routing problem," Proc. Bio-Inspired Computing, Beijing, China, October 2009.
[19] Alireza Abbasy, and Seyed Hamid Hosseini, "Ant colony optimization-based approach to optimal reactive power dispatch: a comparison of various ant systems," Proc. IEEE PES PowerAfrica, Johannesburg, South Africa, July 2007.