

# CDN Project

Ian Fund

1656736

## 1 Introduction

This system solves the problem of finding the shortest path to deliver content to the user. A distance vector algorithm is used to find the shortest path between the proxy and the file the user is looking for.

## 2 Design and Implementation

My project consists of four components: 1) CDN 2) servers for each node in CDN 3) proxy 4) browser for proxy to connect to.

Each server and CDN node is started with arguments given in README. The CDN creates a routing table based on a configuration file. Each node then shares its routing table with each node it is linked with. Every time a table is shared the node will update its table using a distance vector algorithm. Over time, the shortest path to each node is found, along with which hop it needs to make next to get to its destination in the shortest time.

The proxy makes a request by connecting to a single node in the network and requesting a file. The request is in the form of *http : // < proxyIP > : < nodeport > / < destination > .html*  
for example: *http : //localhost : 8088/africa.html*

The node receives the request and then checks it's cache and server (in that order) to see if it has the file. If it does have the file it writes it to the browser. If it does NOT have the file it will use the destination and its routing table to find the shortest path to the destination. Along the route, each node will check it's cache and server for the file. When the file it found it will return it to the original requester and write it to the browser.

Whenever a request is made the proper delay time used to simulate delay. Also, each time a request is sent or received the time , destination, and sent/received is written to a log file.

The feature I implemented is pre-loading. Whenever the server is started,

each node will load file in its server to it's cache to allow for faster load times.

### 3 Evaluation

The performance of the two caching styles were measured with the following constraints: The request was sent over three nodes and each node had a delay of two seconds to each other node. Each was measured five times

Cache style one (caching the file along each hop): 13.7614 , 4.0153 , 4.0158, 4.0149, 4.0132

Cache style two (caching file on the next hop that doesn't have it) 13.9259 ,10.2542 , 4.0110 ,4.0172 ,4.0194

Each style converged to roughly four seconds after having the file in its own cache. Style two took longer because it took more requests before the file was in its cache.

### 4 Limitations

Only html was implemented for this project. I did not use the standardized web address, I misread it early on and developed everything around a different address type. The cache folders are not emptied automatically. When closing the CDN nodes you have to press control+C several times to close.