

Multisource and Multipath Content Transfer

R & D Project Presentation

Shruti Hiray
Guide: Prof. D. Manjunath

Department of Electrical Engineering
IIT Bombay

November 2018

Table of Contents

1 Introduction

2 Implementation

Introduction

- Devices can make use of multiple internet connections (WiFi, Ethernet, Cellular) by data transfer through multipath.
- Parallel download can increase overall throughput of file download.
- Files are divided into chunks which are stored in Content Server.
- Some of the file chunks are stored in cache of WiFi Access points.
- Controller at server directs clients to get the available chunks from local cache on Access points.
- Helps in decreasing the path length of data transfer for some chunks, leads to lesser download time and less network congestion.
- Remaining chunks are delivered to client from server.
- Client thus obtain chunks from multiple sources through multiple paths.
- Client finally orders the chunks and merges to single file.

Implementation

- Client sends a file request to server through web browser.
- Client browser plugin will filter requests that matches the required url.
- After request reaches the server, it decides which chunks are to be delivered by which WiCache AP and which by itself.
- The mapping <Chunk_Cache_Pair_List> is send to client.
- Plugin then sends request for each file chunk to appropriate WiCache AP or Server.
- After receiving all the chunks, plugin orders and merges to a single file and sends it to browser application.
- In this manner, client downloads file through multisource and multipath.

Content Request Algorithm

Client is connected to the cellular network and WiFi network

- Client initiates TCP connection to content server (through cellular network).
- Server always listens for TCP connection from clients.
- Client generates random 48-bit key (`Client_Key`).
- Client sends file request: `<File_Req> <File_Name> <Client_Key>`
- Server responds to file request:
`<File_Req_Res> <File_Name> <File_Size> <Chunk_Count>`
`<Chunk_Cache_Pair_List> <Connection_Key>`
- `<Connection_Key>` is a unique 32-bit key assigned by server for every file request
- `<Chunk_Cache_Pair_List>`: `<Chunk_Name, Cache_IP/Server>` which instructs client the location to fetch each chunk from.
- `<Chunk_Cache_Pair_List>` is determined by `BEST_PATH_FOR_CHUNK_ALGORITHM` at the server.

Content Delivery Algorithm

- Chunk will be requested from either Content Server or WiCache AP or both by the client using <Chunk_Cache_Pair_List>
- Content server informs WiCache AP about the incoming chunk request from the client: <Client_Request> <Chunk_Name> <Skip_Bytes> <Connection_Key> <Client_Key> <Neighbor_AP_IP>
Neighbor_AP_IP=false, if chunk need not be fetched from a neighbor AP
- Client requests for chunk: <Chunk_Request> <Chunk_Name> <Skip_Bytes> <Connection_Key> <Client_Key>
- Content server/WiCache AP sends chunk delivery information to client before delivering the chunk data: <Chunk_Delivery> <Chunk_Name> <Chunk_Size> <Skip_Bytes> <Connection_Key>

