CN 3530: Assignment 1 - Star Topology

Name: Prashanth Sriram S **Date:** 14/10/2022
Roll: CS20BTECH11039

(Client and Server for GET/PUT/DELETE Requests - Basic Topology's Report is in the basic folder)

II. Web Cache:

a) Implementation Details: For the cache, to not let the cache size be unbounded, an LRU replacement is used and at a time maximum 5(arbitrarily chosen for the assignment) key value pair can be stored in the cache and for the next pair to be cached, the least recently used pair is removed and the new pair is added, More details in the comments of the code

b) Pcap Reports: After coding for client, cache and server, wireshark was run to capture pcap traces for h1, h2 and h3 for first a GET request for a key (that was inserted using a PUT request before) and then another GET request for the same key again. All three pcap files are submitted along with the report.

c) Response Times:

Then, From H1, GET request for 6 Keys are made, each key 3 times. The response time are as follows (Measured using wireshark):

Key	First time (sec)	Second Time (sec)	Third Time (sec)
Key1	0.004989131	0.001398384	0.002820462
Key2	0.00701165	0.0019808	0.00223889
Key3	0.010016123	0.002261283	0.002090418
Key4	0.010015121	0.001937496	0.002275067
Key5	0.009289922	0.004320585	0.002631847
Key6	0.00856099	0.002763958	0.005435925
Average	0.008313822833	0.002443751	0.002915434833
Avg in ms	8.313822833	2.443751	2.915434833

(Note: Socket connection establishment times are not taken into account in the above data, since wireshark measures the time since request for the response to arrive and the request is sent only after the socket connection is established.)

Observation:

While time for second and third requests are almost the same, they both take significantly lesser time than the first request

Justification:

For the first request, the cache server(h2) doesn't have that key in its cache and has to request the main server(h3) to get the value for the key and then cache it and then send the response to the client with the value. But for the subsequent requests, since the keys are present in the cache, h2 doesn't request h3 and instead sends the cached value to h1. So, in the second and third requests, there is no time spent on sending a request to h3, h3 processing it and h3 sending back a response, unlike in the case of the first request.

ACKNOWLEDGEMENTS:

1. **https://medium.com/@narengowda/designing-the-caching-system-e42c6938df6a For how to design/implement an LRU cache in the h2 cache**
2. **Mozilla MDN Docs: https://developer.mozilla.org/en-US/docs/Web/HTTP/Messages to see the format of GET, PUT, DELETE Requests and the response codes.**
3. **Python docs on socket programming: https://docs.python.org/3/howto/sockets.html**
4. **Computer Networking : A top down approach by Kurose Ross**
5. **The boiler plate code present for the assignment already**

PLAGIARISM STATEMENT

We certify that this assignment/report is our own work, based on our personal study and/or research and that we have acknowledged all material and sources used in its preparation, whether they be books, articles, packages, datasets, reports, lecture notes, and any other kind of document, electronic or personal communication. We also certify that this assignment/report has not previously been submitted for assessment/project in any other course lab, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that we have not copied in part or whole or otherwise plagiarized the work of other students and/or persons. We pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, We understand my responsibility to report honor violations by other students if we become aware of it.

Name: Prashanth Sriram S
Date: 14/10/2022
Signature: Prashanth Sriram S