

Assignment 2: CS3530

1. Emulating an HTTP Server (Basic Topology)

The topology contains 2 hosts - h1 and h2, both of which act as a client and a server respectively connected to the same switch s1.

a. PCAP traces at H1 for all three types of requests

Following are three snapshots of the pcap traces for all the types of requests:
(1 request/response for each of GET, DELETE, PUT)

Note that the individual pcap files are available in the folder basic/pcaps.

GET request trace at H1

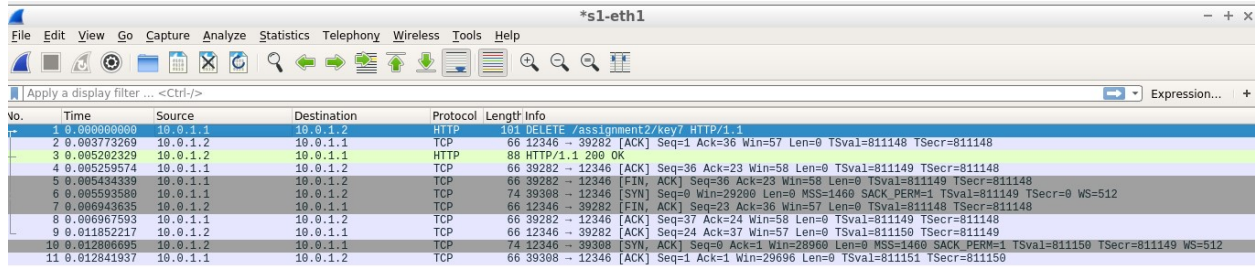
The screenshot displays the Wireshark interface for a PCAP file named h1_GET.pcap. The packet list shows a GET request (196 bytes) and its response (12346 bytes). The packet details pane shows the request structure: Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Hypertext Transfer Protocol. The packet bytes pane shows the raw data of the request, including the GET /a ssignmen t2?reque st=key1 HTTP/1.1 line.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.1.1	10.0.1.2	HTTP	196	GET /assignment2?request=key1 HTTP/1.1
2	0.004746161	10.0.1.2	10.0.1.1	TCP	66	12346 → 39186 [ACK] Seq=1 Ack=41 Win=57 Len=0 TSval=721742 TSecr=721742
3	0.006209966	10.0.1.2	10.0.1.1	HTTP	93	HTTP/1.1 200 OK
4	0.006363355	10.0.1.1	10.0.1.2	TCP	66	39186 → 12346 [ACK] Seq=41 Ack=28 Win=58 Len=0 TSval=721743 TSecr=721743
5	0.006681435	10.0.1.1	10.0.1.2	TCP	66	39186 → 12346 [FIN, ACK] Seq=41 Ack=28 Win=58 Len=0 TSval=721744 TSecr=721743
6	0.006819355	10.0.1.1	10.0.1.2	TCP	74	39214 → 12346 [SYN] Seq=0 Win=29260 Len=0 MSS=1460 SACK_PERM=1 TSval=721744 TSecr=0 WS=512
7	0.009716449	10.0.1.2	10.0.1.1	TCP	66	12346 → 39186 [FIN, ACK] Seq=28 Ack=41 Win=57 Len=0 TSval=721743 TSecr=721742
8	0.009739182	10.0.1.1	10.0.1.2	TCP	66	39186 → 12346 [ACK] Seq=42 Ack=29 Win=58 Len=0 TSval=721744 TSecr=721743
9	0.016296518	10.0.1.2	10.0.1.1	TCP	66	12346 → 39186 [ACK] Seq=29 Ack=42 Win=57 Len=0 TSval=721745 TSecr=721744
10	0.017433479	10.0.1.2	10.0.1.1	TCP	74	12346 → 39214 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=721745 TSecr=721744 WS=512
11	0.017466598	10.0.1.1	10.0.1.2	TCP	66	39214 → 12346 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=721746 TSecr=721745

Frame 1: 196 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface 0
Ethernet II, Src: 00:00:00:00:01:01 (00:00:00:00:01:01), Dst: 00:00:00:00:01:02 (00:00:00:00:01:02)
Internet Protocol Version 4, Src: 10.0.1.1, Dst: 10.0.1.2
Transmission Control Protocol, Src Port: 39186, Dst Port: 12346, Seq: 1, Ack: 1, Len: 40
Hypertext Transfer Protocol

```
0000  00 00 00 00 01 02 00 00 00 00 01 01 08 00 45 00  .....E..
0010  00 5c 13 6f 40 00 40 06 11 2b 0a 00 01 01 0a 00  ..\..@:..+....
0020  01 02 99 12 30 3a e9 88 fd 9e 9f e6 1b 72 80 18  ...0:.....r...
0030  00 3a ac ac 00 00 01 01 08 0a 00 0b 03 4e 00 0a  ...a...N.....
0040  ee f0 47 45 54 20 2f 61 73 73 69 67 6e 6d 65 6e  ..GET /a ssignmen
0050  74 32 3f 72 65 71 75 65 73 74 3d 6b 65 79 31 20  t2?reque st=key1
0060  48 54 54 50 2f 31 2e 31 0a 0a  HTTP/1.1 ..
```

DELETE request trace at H1



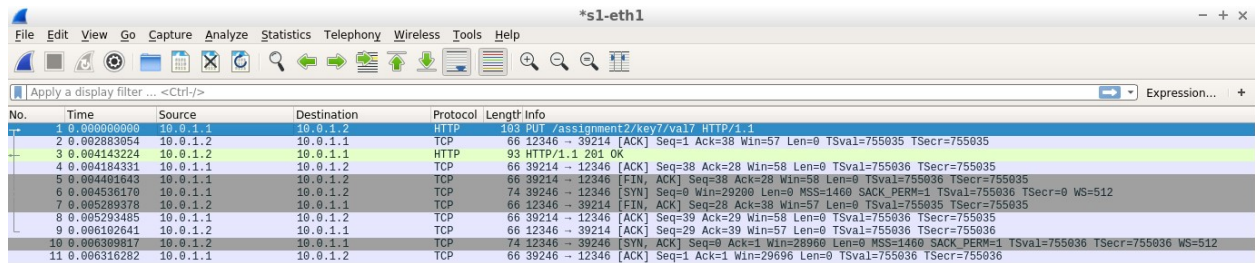
The screenshot shows a Wireshark capture on interface *s1-eth1. The packet list table is as follows:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.1.1	10.0.1.2	HTTP	101	DELETE /assignment2/key7 HTTP/1.1
2	0.003773269	10.0.1.2	10.0.1.1	TCP	66	12346 → 39282 [ACK] Seq=1 Ack=36 Win=57 Len=0 TSval=811148 TSecr=811148
3	0.005202329	10.0.1.2	10.0.1.1	HTTP	88	HTTP/1.1 200 OK
4	0.005259574	10.0.1.1	10.0.1.2	TCP	66	39282 → 12346 [ACK] Seq=36 Ack=23 Win=58 Len=0 TSval=811149 TSecr=811148
5	0.005434339	10.0.1.1	10.0.1.2	TCP	66	39282 → 12346 [FIN, ACK] Seq=36 Ack=23 Win=58 Len=0 TSval=811149 TSecr=811148
6	0.005593589	10.0.1.1	10.0.1.2	TCP	74	39308 → 12346 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=811149 TSecr=0 WS=512
7	0.006943635	10.0.1.2	10.0.1.1	TCP	66	12346 → 39282 [FIN, ACK] Seq=23 Ack=36 Win=57 Len=0 TSval=811148 TSecr=811148
8	0.006967593	10.0.1.1	10.0.1.2	TCP	66	39282 → 12346 [ACK] Seq=37 Ack=24 Win=58 Len=0 TSval=811149 TSecr=811148
9	0.011852217	10.0.1.2	10.0.1.1	TCP	66	12346 → 39282 [ACK] Seq=24 Ack=37 Win=57 Len=0 TSval=811150 TSecr=811149
10	0.012806695	10.0.1.2	10.0.1.1	TCP	74	12346 → 39308 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=811150 TSecr=811149 WS=512
11	0.012841937	10.0.1.1	10.0.1.2	TCP	66	39308 → 12346 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=811151 TSecr=811150

▶ Frame 1: 101 bytes on wire (808 bits), 101 bytes captured (808 bits) on interface 0
▶ Ethernet II, Src: 00:00:00:00:01:01 (00:00:00:00:01:01), Dst: 00:00:00:00:01:02 (00:00:00:00:01:02)
▶ Internet Protocol Version 4, Src: 10.0.1.1, Dst: 10.0.1.2
▶ Transmission Control Protocol, Src Port: 39282, Dst Port: 12346, Seq: 1, Ack: 1, Len: 35
▶ Hypertext Transfer Protocol

```
0000 00 00 00 00 01 02 00 00 00 00 01 01 08 00 45 00 .....E
0010 00 57 b6 78 40 00 40 06 74 26 0a 00 01 01 0a 00 Wx@ @ t&....
0020 01 02 99 72 30 3a 94 b9 4f 25 fb 0b d6 7b 80 18 ...r0:...0%...{..
0030 00 3a 1c 58 00 00 01 01 08 0a 00 0c 60 8c 00 00 :X.....
0040 ff 52 44 45 4c 45 54 45 20 2f 61 73 73 69 67 6e RDELETE /assign
0050 6d 65 6e 74 32 2f 6b 65 79 37 20 48 54 50 2f ment2/ke y7 HTTP/
0060 31 2e 31 0a 0a 1.1..
```

PUT request trace at H1



The screenshot shows a Wireshark capture on interface *s1-eth1. The packet list table is as follows:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.1.1	10.0.1.2	HTTP	103	PUT /assignment2/key7/val7 HTTP/1.1
2	0.002883954	10.0.1.2	10.0.1.1	TCP	66	12346 → 39214 [ACK] Seq=1 Ack=38 Win=57 Len=0 TSval=755035 TSecr=755035
3	0.004143224	10.0.1.2	10.0.1.1	HTTP	93	HTTP/1.1 201 OK
4	0.004184331	10.0.1.1	10.0.1.2	TCP	66	39214 → 12346 [ACK] Seq=38 Ack=28 Win=58 Len=0 TSval=755036 TSecr=755035
5	0.004401643	10.0.1.1	10.0.1.2	TCP	66	39214 → 12346 [FIN, ACK] Seq=38 Ack=28 Win=58 Len=0 TSval=755036 TSecr=755035
6	0.004536170	10.0.1.1	10.0.1.2	TCP	74	39246 → 12346 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=755036 TSecr=0 WS=512
7	0.005269378	10.0.1.2	10.0.1.1	TCP	66	12346 → 39214 [FIN, ACK] Seq=28 Ack=38 Win=57 Len=0 TSval=755036 TSecr=755035
8	0.005293465	10.0.1.1	10.0.1.2	TCP	66	39214 → 12346 [ACK] Seq=39 Ack=29 Win=58 Len=0 TSval=755036 TSecr=755035
9	0.006102641	10.0.1.2	10.0.1.1	TCP	66	12346 → 39214 [ACK] Seq=29 Ack=39 Win=57 Len=0 TSval=755036 TSecr=755036
10	0.006399817	10.0.1.1	10.0.1.1	TCP	74	12346 → 39246 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=755036 TSecr=755036 WS=512
11	0.006316282	10.0.1.1	10.0.1.2	TCP	66	39246 → 12346 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=755036 TSecr=755036

▶ Frame 1: 103 bytes on wire (824 bits), 103 bytes captured (824 bits) on interface 0
▶ Ethernet II, Src: 00:00:00:00:01:01 (00:00:00:00:01:01), Dst: 00:00:00:00:01:02 (00:00:00:00:01:02)
▶ Internet Protocol Version 4, Src: 10.0.1.1, Dst: 10.0.1.2
▶ Transmission Control Protocol, Src Port: 39214, Dst Port: 12346, Seq: 1, Ack: 1, Len: 37
▶ Hypertext Transfer Protocol

```
0000 00 00 00 00 01 02 00 00 00 00 01 01 08 00 45 00 .....E
0010 00 59 40 43 40 00 40 06 dc 59 0a 00 01 01 0a 00 -YH@ @ -Y.....
0020 01 02 99 2e 30 3a 1b c6 4b 12 0d 78 94 3a 80 18 ...0:...K-x:..
0030 00 3a 2d 2d 00 00 01 01 08 0a 00 0b 85 5b 00 18 :.....[...
0040 03 51 50 55 54 20 2f 61 73 73 69 67 6e 6d 65 6e QPUT /a ssignmen
0050 74 32 2f 6b 65 79 37 2f 76 61 6c 37 20 48 54 54 t2/key7/ val7 HTT
0060 50 2f 31 2e 31 0a 0a P/1.1..
```

b. End to End times

From H1, GET all 6 keys 3 times each and note down the end-to-end time taken to finish the GET request. That is, capture time before issuing a request and after receiving the response and report the difference in the table below.

Key	Request 1 (first time)	Request 2 (second time)	Request 3 (third time)	Average Time
key1	0.005174862	0.005140435	0.004806779	0.005040692
key2	0.001287872	0.003804495	0.004518082	0.003203483
key3	0.005198547	0.004181163	0.0045346	0.0046381033
key4	0.004011415	0.003938796	0.004262324	0.0040708450
key5	0.004838614	0.004963489	0.004757225	0.0048531093
key6	0.004875395	0.004828673	0.003964222	0.0045560966

2. Web Cache Development (Star Topology)

The topology contains 3 hosts - h1, h2 and h3, acting as a client, cache and a server respectively and are connected to the same switch in a star fashion.

a. PCAP traces for GET request of same key (key1)

i. When the key is not present in the cache (H2)

When the client requests the cache for key1's value initially, the key-value pair is not present in the cache. The cache then forwards the request to the server, and the server replies with the value, which the cache will then send to the client. The value of that pair will also be added to the cache's database (this can be tracked in the output of the terminal while running the requests).

PCAP traces snapshot at h1 (client) during the first GET request and response

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.1.1	10.0.1.2	HTTP	106	GET /assignment2?request=key1 HTTP/1.1
2	0.004509578	10.0.1.2	10.0.1.1	TCP	66	12345 → 46418 [ACK] Seq=1 Ack=41 Win=57 Len=0 TSval=2719003 TSecr=2719002
3	0.011209492	10.0.1.2	10.0.1.1	HTTP	93	HTTP/1.1 200 OK
4	0.011224567	10.0.1.1	10.0.1.2	TCP	66	46418 → 12345 [ACK] Seq=41 Ack=28 Win=58 Len=0 TSval=2719005 TSecr=2719005
5	0.011227101	10.0.1.2	10.0.1.1	TCP	66	12345 → 46418 [FIN, ACK] Seq=28 Ack=41 Win=57 Len=0 TSval=2719005 TSecr=2719002
6	0.011265516	10.0.1.1	10.0.1.2	TCP	66	46418 → 12345 [FIN, ACK] Seq=41 Ack=29 Win=58 Len=0 TSval=2719005 TSecr=2719005
7	0.011294494	10.0.1.1	10.0.1.2	TCP	74	46534 → 12345 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2719005 TSecr=0 WS=512
8	0.011939106	10.0.1.2	10.0.1.1	TCP	66	12345 → 46418 [ACK] Seq=29 Ack=42 Win=57 Len=0 TSval=2719005 TSecr=2719005
9	0.012987912	10.0.1.2	10.0.1.1	TCP	74	12345 → 46534 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=2719005 TSecr=2719005 WS=512
10	0.012996188	10.0.1.1	10.0.1.2	TCP	66	46534 → 12345 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=2719005 TSecr=2719005

Frame 1: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface 0
 Ethernet II, Src: 00:00:00:00:01:01 (00:00:00:00:01:01), Dst: 00:00:00:00:01:02 (00:00:00:00:01:02)
 Internet Protocol Version 4, Src: 10.0.1.1, Dst: 10.0.1.2
 Transmission Control Protocol, Src Port: 46418, Dst Port: 12345, Seq: 1, Ack: 1, Len: 40
 Hypertext Transfer Protocol

```

0000  00 00 00 00 01 02 00 00 00 00 01 01 00 00 45 00  .....E
0010  00 5c 34 00 40 00 40 06 f0 99 0a 00 01 01 0a 00  \4.0.0
0020  01 02 b5 52 30 39 67 5e 08 e5 c0 ff 5b d0 80 18  ..R09g^
0030  00 3a df b3 00 00 01 01 08 0a 00 29 7d 1a 00 1c  :.....}}
0040  3b 1b 47 45 54 20 2f 61 73 73 69 67 6e 60 65 6e  ;GET /a ssignmen
0050  74 32 3f 72 65 71 75 65 73 74 3d 60 65 79 31 20  t2?reque st=key1
0060  48 54 54 50 2f 31 2e 31 0a 0a  .....HTTP/1.1

```

Here we can see the first request/response pair, for when the key is not present in the cache. This is the GET request from the client side to the cache. The overall end-2-end time taken for this request is around 0.012 seconds.

PCAP traces snapshot at h2 (cache) during the first GET request and response

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.1.1	10.0.1.2	HTTP	106	GET /assignment?request=key1 HTTP/1.1
2	0.000063831	10.0.1.2	10.0.1.1	TCP	66	12345 → 46418 [ACK] Seq=1 Ack=41 Win=57 Len=0 TSval=2719003 TSecr=2719002
3	0.000461477	10.0.1.2	10.0.1.3	TCP	74	42886 → 12345 [SYN] Seq=0 Win=29208 Len=0 MSS=1460 SACK_PERM=1 TSval=2719003 TSecr=0 WS=512
4	0.003751421	10.0.1.3	10.0.1.2	TCP	74	12345 → 42886 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=2719003 TSecr=2719003 WS=512
5	0.003787666	10.0.1.2	10.0.1.3	TCP	66	42886 → 12345 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=2719003 TSecr=2719003
6	0.005180366	10.0.1.2	10.0.1.3	HTTP	106	GET /assignment?request=key1 HTTP/1.1
7	0.005894491	10.0.1.3	10.0.1.2	TCP	66	12345 → 42886 [ACK] Seq=1 Ack=41 Win=29184 Len=0 TSval=2719004 TSecr=2719004
8	0.006043640	10.0.1.3	10.0.1.2	HTTP	93	HTTP/1.1 200 OK
9	0.006191808	10.0.1.3	10.0.1.2	TCP	66	12345 → 42886 [FIN, ACK] Seq=28 Ack=41 Win=29184 Len=0 TSval=2719004 TSecr=2719004
10	0.006191813	10.0.1.2	10.0.1.3	TCP	66	42886 → 12345 [ACK] Seq=41 Ack=28 Win=29696 Len=0 TSval=2719004 TSecr=2719004
11	0.006379719	10.0.1.2	10.0.1.3	TCP	66	42886 → 12345 [FIN, ACK] Seq=41 Ack=29 Win=29696 Len=0 TSval=2719004 TSecr=2719004
12	0.006743308	10.0.1.3	10.0.1.2	TCP	66	12345 → 42886 [ACK] Seq=29 Ack=42 Win=29184 Len=0 TSval=2719004 TSecr=2719004
13	0.008875902	10.0.1.2	10.0.1.1	HTTP	93	HTTP/1.1 200 OK
14	0.008885716	10.0.1.2	10.0.1.1	TCP	66	12345 → 46418 [FIN, ACK] Seq=28 Ack=41 Win=57 Len=0 TSval=2719005 TSecr=2719002
15	0.009786239	10.0.1.1	10.0.1.2	TCP	66	46418 → 12345 [ACK] Seq=41 Ack=28 Win=58 Len=0 TSval=2719005 TSecr=2719005
16	0.009798333	10.0.1.1	10.0.1.2	TCP	66	46418 → 12345 [FIN, ACK] Seq=41 Ack=29 Win=58 Len=0 TSval=2719005 TSecr=2719005
17	0.009801538	10.0.1.2	10.0.1.1	TCP	66	12345 → 46418 [ACK] Seq=29 Ack=42 Win=57 Len=0 TSval=2719005 TSecr=2719005
18	0.009814573	10.0.1.1	10.0.1.2	TCP	74	46534 → 12345 [SYN] Seq=0 Win=29208 Len=0 MSS=1460 SACK_PERM=1 TSval=2719005 TSecr=0 WS=512
19	0.009819091	10.0.1.2	10.0.1.1	TCP	74	12345 → 46534 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=2719005 TSecr=2719005 WS=512
20	0.010305759	10.0.1.1	10.0.1.2	TCP	66	46534 → 12345 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=2719005 TSecr=2719005

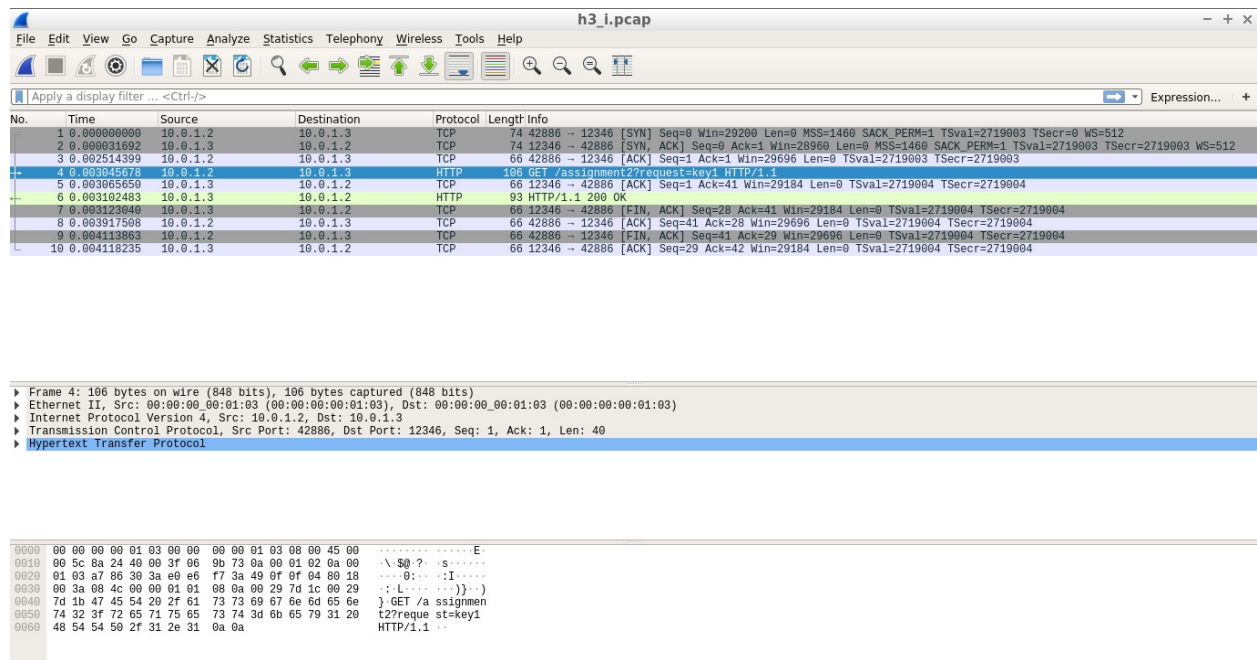
▶ Frame 1: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface 0
 ▶ Ethernet II, Src: 00:00:00:00:01:02 (00:00:00:00:01:02), Dst: 00:00:00:00:01:02 (00:00:00:00:01:02)
 ▶ Internet Protocol Version 4, Src: 10.0.1.1, Dst: 10.0.1.2
 ▶ Transmission Control Protocol, Src Port: 46418, Dst Port: 12345, Seq: 1, Ack: 1, Len: 40
 ▶ Hypertext Transfer Protocol

```

0000  00 00 00 00 01 02 00 00 00 00 01 02 08 00 45 00  .....E
0010  00 5c 34 00 40 00 3f 06 f1 99 0a 00 01 01 0a 00  \4.0.? .....
0020  01 02 b5 52 30 39 67 5e 08 e5 c0 ff 5b d0 80 18  ...R09g^....[...
0030  00 3a df b3 00 00 01 01 08 0a 00 29 7d 1a 00 1c  :.....)]...
0040  3b 1b 47 45 54 20 2f 61 73 73 69 67 6e 6d 65 6e  ;GET /a ssignmen
0050  74 32 3f 72 65 71 75 65 73 74 3d 6b 65 79 31 20  t?reque st=key1
0060  48 54 54 50 2f 31 2e 31 0a 0a  HTTP/1.1 ..
  
```

In the above trace, there are 2 request/response pairs. This shows the request from the client (No 1 - request from the client), then the cache requesting that key from the server (No 6 - request from cache to the server), getting the response from the server with the key-value pair (No 8 - response from server to cache), and finally the cache replying to the client with the value (No 13 - response from cache to client). The trace for this GET request is long because the cache connects to the server, gets the value, copies the value to its own database and then sends the requested value to the client. Essentially 2 requests/responses are happening.

PCAP traces snapshot at h3 (server) during the first GET request and response



The image shows a Wireshark interface with a PCAP file named 'h3_i.pcap'. The packet list pane shows several packets, with packet 4 selected. The packet details pane shows the structure of the selected packet, and the packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.1.2	10.0.1.3	TCP	74	42886 → 12346 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2719003 TSecr=0 WS=512
2	0.000031692	10.0.1.3	10.0.1.2	TCP	74	12346 → 42886 [SYN, ACK] Seq=9 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=2719003 TSecr=2719003 WS=512
3	0.002514399	10.0.1.2	10.0.1.3	TCP	66	42886 → 12346 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=2719003 TSecr=2719003
4	0.003045678	10.0.1.2	10.0.1.3	HTTP	106	GET /assignment2?request=key1 HTTP/1.1
5	0.003065650	10.0.1.3	10.0.1.2	TCP	66	12346 → 42886 [ACK] Seq=1 Ack=41 Win=29184 Len=0 TSval=2719004 TSecr=2719004
6	0.003102483	10.0.1.3	10.0.1.2	HTTP	93	HTTP/1.1 200 OK
7	0.003129048	10.0.1.3	10.0.1.2	TCP	66	12346 → 42886 [FIN, ACK] Seq=28 Ack=41 Win=29184 Len=0 TSval=2719004 TSecr=2719004
8	0.003911508	10.0.1.2	10.0.1.3	TCP	66	42886 → 12346 [ACK] Seq=41 Ack=28 Win=29696 Len=0 TSval=2719004 TSecr=2719004
9	0.004113863	10.0.1.2	10.0.1.3	TCP	66	42886 → 12346 [FIN, ACK] Seq=41 Ack=29 Win=29696 Len=0 TSval=2719004 TSecr=2719004
10	0.004118235	10.0.1.3	10.0.1.2	TCP	66	12346 → 42886 [ACK] Seq=29 Ack=42 Win=29184 Len=0 TSval=2719004 TSecr=2719004

Frame 4: 106 bytes on wire (848 bits), 106 bytes captured (848 bits)
▶ Ethernet II, Src: 00:00:00:00:01:03 (00:00:00:00:01:03), Dst: 00:00:00:00:01:03 (00:00:00:00:01:03)
▶ Internet Protocol Version 4, Src: 10.0.1.2, Dst: 10.0.1.3
▶ Transmission Control Protocol, Src Port: 42886, Dst Port: 12346, Seq: 1, Ack: 1, Len: 40
▶ Hypertext Transfer Protocol

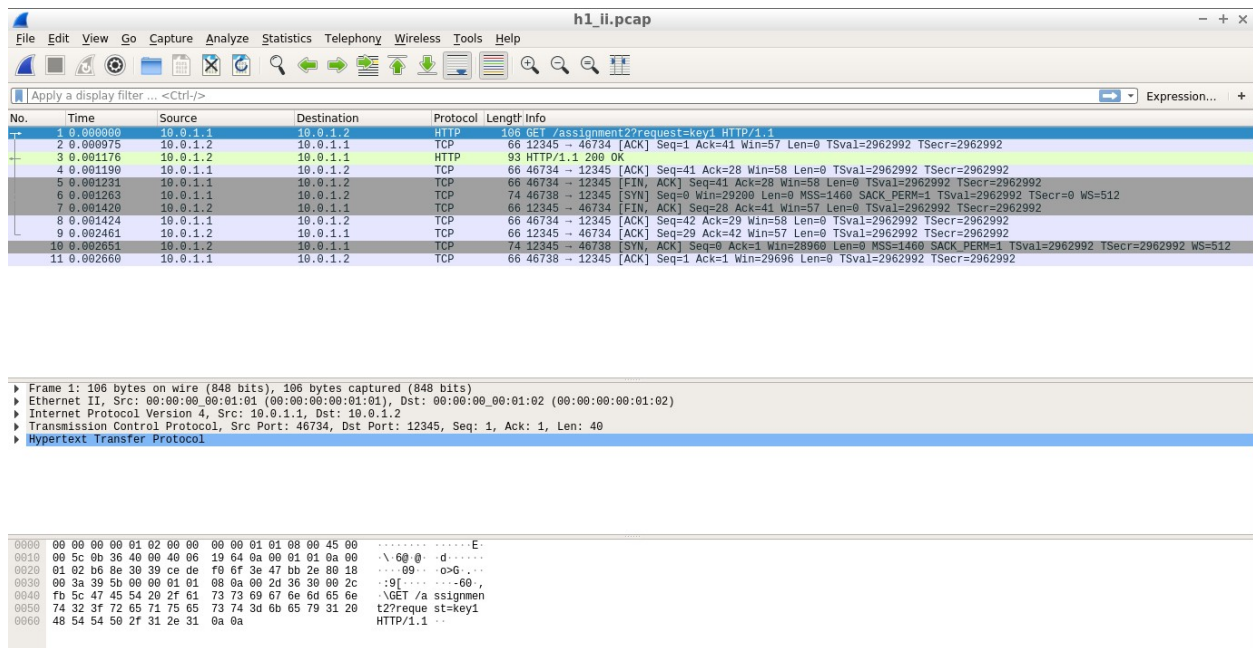
```
0000 00 00 00 00 01 03 00 00 00 00 01 03 08 00 45 00 .....E
0010 00 5c 8a 24 40 00 3f 06 9b 73 0a 00 01 02 0a 00 \.S0?..s.....
0020 61 03 a7 06 30 3a e0 e6 f7 3a 49 0f 0f 04 00 18 ...0:..:I....
0030 00 3a 06 4c 00 00 01 01 08 0a 00 29 7d 1c 00 29 ..L.....:..
0040 7d 1b 47 45 54 20 2f 61 73 73 69 67 6e 6d 65 6e }.GET /a ssignmen
0050 74 32 3f 72 65 71 75 65 73 74 3d 6b 65 79 31 20 t2?reque st=key1
0060 48 54 54 50 2f 31 2e 31 0a 0a HTTP/1.1 ..
```

This trace is only that of the cache requesting the server for the same key value pair that was requested by the client to the cache. Interaction with the server is only present in this first GET request. This is also that of a simple client-server connection, where ordinary server-like behavior is observed.

ii. When the key is present in the cache (H2)

Now, after the first request we know for sure that the key value pair is now stored in the caches database. So, when the client makes a request for this key at this point, we can assume it will be an ordinary GET request behavior at the client and cache side - the cache will simply reply with the requested value.

PCAP traces snapshot at h1 (client) during the second GET request and response



The image shows a Wireshark interface for a PCAP file named h1_ii.pcap. The packet list on the left shows 11 packets. Packet 10 is the second GET request, and packet 11 is the response. The packet details pane shows the structure of the first packet (106 bytes on wire, 106 bytes captured). The packet bytes pane shows the raw data of the first packet, which is a GET request for /assignment2?request=key1.

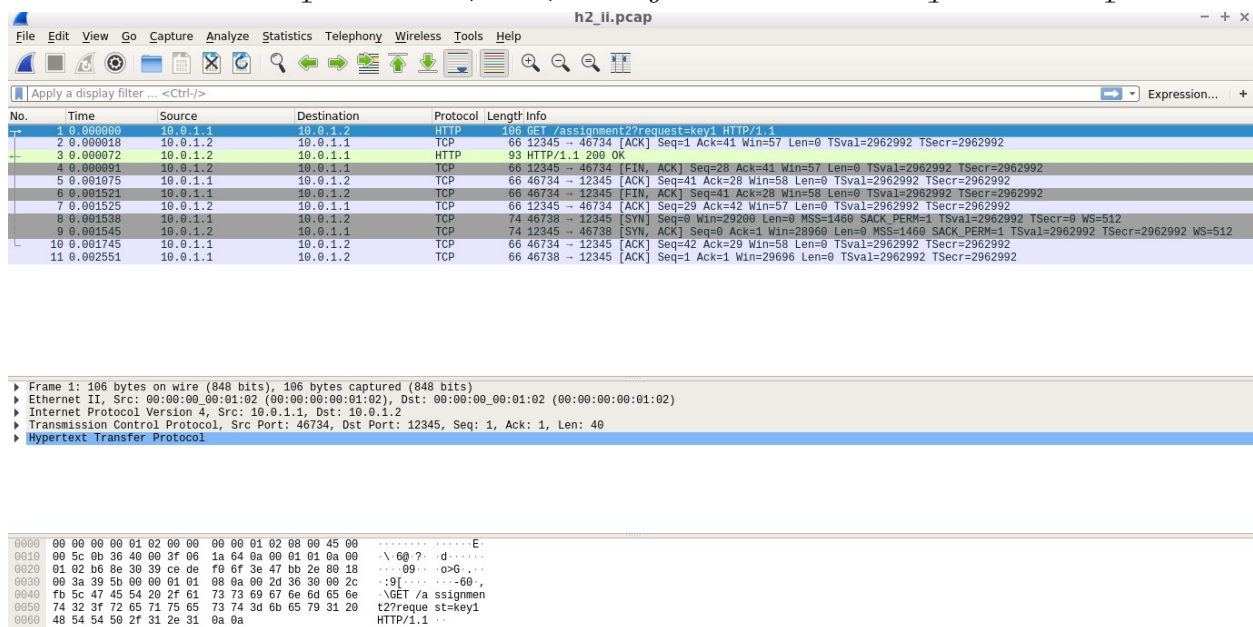
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.1.1	10.0.1.2	HTTP	106	GET /assignment2?request=key1 HTTP/1.1
2	0.000075	10.0.1.2	10.0.1.1	TCP	66	12345 → 46734 [ACK] Seq=1 Ack=41 Win=57 Len=0 TSval=2962992 TSecr=2962992
3	0.000176	10.0.1.2	10.0.1.1	HTTP	93	HTTP/1.1 200 OK
4	0.001190	10.0.1.1	10.0.1.2	TCP	66	46734 → 12345 [ACK] Seq=41 Ack=28 Win=58 Len=0 TSval=2962992 TSecr=2962992
5	0.001231	10.0.1.1	10.0.1.2	TCP	66	46734 → 12345 [FIN, ACK] Seq=41 Ack=28 Win=58 Len=0 TSval=2962992 TSecr=2962992
6	0.001263	10.0.1.1	10.0.1.2	TCP	74	46738 → 12345 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2962992 TSecr=0 WS=512
7	0.001428	10.0.1.2	10.0.1.1	TCP	66	12345 → 46734 [FIN, ACK] Seq=28 Ack=41 Win=57 Len=0 TSval=2962992 TSecr=2962992
8	0.001424	10.0.1.1	10.0.1.2	TCP	66	46734 → 12345 [ACK] Seq=42 Ack=29 Win=58 Len=0 TSval=2962992 TSecr=2962992
9	0.002461	10.0.1.2	10.0.1.1	TCP	66	12345 → 46734 [ACK] Seq=29 Ack=42 Win=57 Len=0 TSval=2962992 TSecr=2962992
10	0.002651	10.0.1.2	10.0.1.1	TCP	74	12345 → 46738 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=2962992 TSecr=2962992 WS=512
11	0.002660	10.0.1.1	10.0.1.2	TCP	66	46738 → 12345 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=2962992 TSecr=2962992

Frame 1: 106 bytes on wire (848 bits), 106 bytes captured (848 bits)
Ethernet II, Src: 00:00:00:00:01:01 (00:00:00:00:01:01), Dst: 00:00:00:00:01:02 (00:00:00:00:01:02)
Internet Protocol Version 4, Src: 10.0.1.1, Dst: 10.0.1.2
Transmission Control Protocol, Src Port: 46734, Dst Port: 12345, Seq: 1, Ack: 1, Len: 40
Hypertext Transfer Protocol

0000 00 00 00 00 01 02 00 00 00 00 01 01 08 00 45 00E
0010 00 5c 0b 36 40 00 40 06 1a 64 0a 00 01 01 0a 00 \.60.?.d.....
0020 01 02 b6 8e 30 39 ce de f0 6f 3e 47 bb 2e 08 1809....oG...
0030 00 3a 39 5b 00 00 01 01 08 0a 00 2d 36 30 00 2c :9[.....60.,
0040 fb 5c 47 45 54 20 2f 61 73 73 69 67 6e 6d 65 6e \GET /a ssignmen
0050 74 32 3f 72 65 71 75 65 73 74 3d 6b 65 79 31 20 t2?reque st=key1
0060 48 54 54 50 2f 31 2e 31 0a 0a HTTP/1.1 ..

Here we can see the second GET request for the same key. By this point, the key value pair is stored in the cache database. Now, the end-to-end time taken is around 0.001 seconds. So there is a significant reduction in the time taken to give the response to the client compared to the first request.

PCAP traces snapshot at h2 (cache) during the second GET request and response



The image shows a Wireshark interface for a PCAP file named h2_ii.pcap. The packet list on the left shows 11 packets. Packet 1 is the second GET request, and packet 2 is the response. The packet details pane shows the structure of the first packet (106 bytes on wire, 106 bytes captured). The packet bytes pane shows the raw data of the first packet, which is a GET request for /assignment2?request=key1.

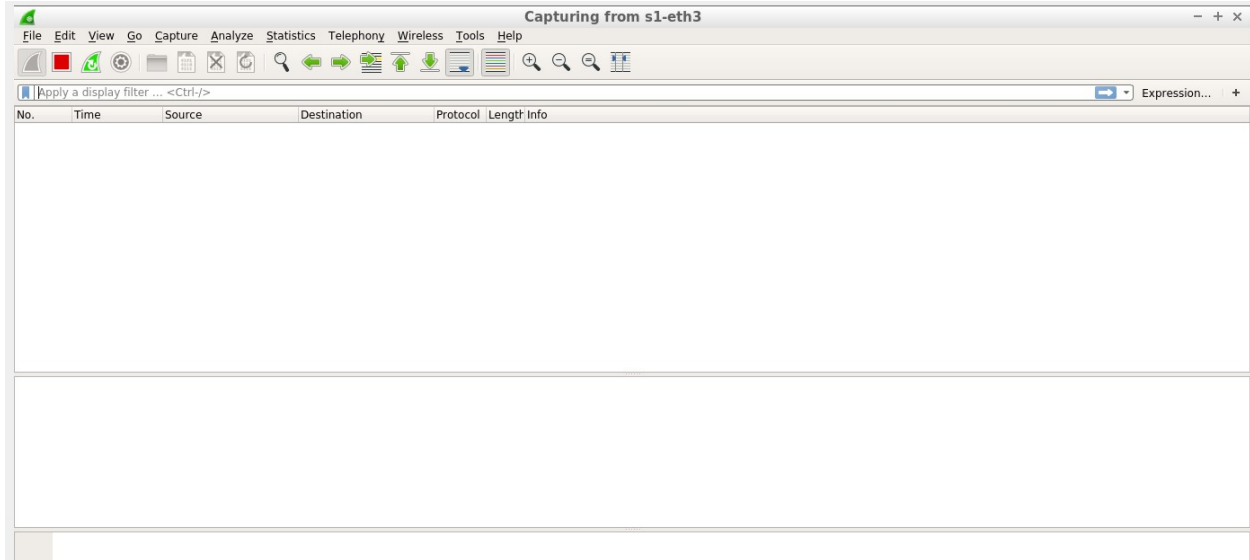
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.1.2	10.0.1.1	HTTP	106	GET /assignment2?request=key1 HTTP/1.1
2	0.000018	10.0.1.2	10.0.1.1	TCP	66	12345 → 46734 [ACK] Seq=1 Ack=41 Win=57 Len=0 TSval=2962992 TSecr=2962992
3	0.000072	10.0.1.2	10.0.1.1	HTTP	93	HTTP/1.1 200 OK
4	0.000091	10.0.1.2	10.0.1.1	TCP	66	12345 → 46734 [FIN, ACK] Seq=28 Ack=41 Win=57 Len=0 TSval=2962992 TSecr=2962992
5	0.001075	10.0.1.1	10.0.1.2	TCP	66	46734 → 12345 [ACK] Seq=41 Ack=28 Win=58 Len=0 TSval=2962992 TSecr=2962992
6	0.001521	10.0.1.2	10.0.1.1	TCP	66	12345 → 46734 [FIN, ACK] Seq=41 Ack=28 Win=58 Len=0 TSval=2962992 TSecr=2962992
7	0.001525	10.0.1.2	10.0.1.1	TCP	66	12345 → 46734 [ACK] Seq=29 Ack=42 Win=57 Len=0 TSval=2962992 TSecr=2962992
8	0.001538	10.0.1.1	10.0.1.2	TCP	74	46738 → 12345 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2962992 TSecr=0 WS=512
9	0.001545	10.0.1.2	10.0.1.1	TCP	74	12345 → 46738 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=2962992 TSecr=2962992 WS=512
10	0.001745	10.0.1.1	10.0.1.2	TCP	66	46734 → 12345 [ACK] Seq=42 Ack=29 Win=58 Len=0 TSval=2962992 TSecr=2962992
11	0.002551	10.0.1.1	10.0.1.2	TCP	66	46738 → 12345 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=2962992 TSecr=2962992

Frame 1: 106 bytes on wire (848 bits), 106 bytes captured (848 bits)
Ethernet II, Src: 00:00:00:00:01:02 (00:00:00:00:01:02), Dst: 00:00:00:00:01:02 (00:00:00:00:01:02)
Internet Protocol Version 4, Src: 10.0.1.1, Dst: 10.0.1.2
Transmission Control Protocol, Src Port: 46734, Dst Port: 12345, Seq: 1, Ack: 1, Len: 40
Hypertext Transfer Protocol

0000 00 00 00 00 01 02 00 00 00 00 01 02 08 00 45 00E
0010 00 5c 0b 36 40 00 3f 06 1a 64 0a 00 01 01 0a 00 \.60.?.d.....
0020 01 02 b6 8e 30 39 ce de f0 6f 3e 47 bb 2e 08 1809....oG...
0030 00 3a 39 5b 00 00 01 01 08 0a 00 2d 36 30 00 2c :9[.....60.,
0040 fb 5c 47 45 54 20 2f 61 73 73 69 67 6e 6d 65 6e \GET /a ssignmen
0050 74 32 3f 72 65 71 75 65 73 74 3d 6b 65 79 31 20 t2?reque st=key1
0060 48 54 54 50 2f 31 2e 31 0a 0a HTTP/1.1 ..

Here as well, we can see the second GET request at the cache side. Over here the cache acts like a simple server, the interaction being a simple client-server type connection (like in question 1 - basic topology).

PCAP traces snapshot at h3 (server) during the second GET request and response



As expected, there is no activity or interaction with the server in this case. Since the value is already present with the cache, the cache will be able to satisfy the client's request without involvement of the server.

Note that the individual pcap files are available in the folder star/pcaps.

b. End to End times

Send a total of 3 GET requests for each key. Note down the end-to-end time taken to finish GET requests at H1. That is, capture time before issuing a request and after the response and report the difference in the table below.

Key	Request 1 (first time)	Request 2 (second time)	Request 3 (third time)
key1	0.017914311	0.004772632	0.004921236
key2	0.021335847	0.005533870	0.004148218
key3	0.017454997	0.005572584	0.001033328
key4	0.013621024	0.004087662	0.004633021
key5	0.016182809	0.004299085	0.007892530
key6	0.017011895	0.004356772	0.003929093
Average Time	0.017253481	0.004770434	0.004426237

Comparison of the average time taken for Req1 (all keys), Req2 (all keys) and Req3 (all keys):

1. Differences observed:

From the table above, we can clearly see that on an average, the 1st request takes a significantly higher time than the subsequent 2 requests, for all keys. Almost more than twice the time is taken.

2. Justification for why there is a difference:

The first request takes more time, because the requested data (key-value pair) was initially not present in the cache, at the time of making the first request. So the cache had to request this pair from the server again, and also store a copy of this in its own database. The next 2 requests to the cache, requesting the same key value pair, will now get a much quicker response because the cache now stores a copy of that value with it. Hence, the time initially taken by the cache to request from the server again will have been cut down.

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: Padmini Palivela, Pushkal Mishra

Date: 26-09-2023

Signature: Padmini Palivela, Pushkal Mishra

Group Members:

Padmini Palivela - EE20BTECH11038

Pushkal Mishra - EE20BTECH11042