LAB 2

COMP-9331

Tian Huang

2025-10-05

Exercise 3:

Question 1:

Status Code: 200

Status Phrase: OK

Question 2:

Yes, there is Date: Tue, 23 Sep 2003 05:29:50 GMT

Last Modified: Tue, 23 Sep 2003 05:29:00 GMT

Last modified shows the file last modified on server, that is about file content. Date is about the time of response of this request. For example, it looks like I edit the file on server in 052900 on server, and send to you on 052950.

Question 3:

It is persistent connection. It comes from “Connection: Keep-Alive” and “Keep-Alive: timeout=10, max=100”

Question 4:

Content-Length: 75 bytes

Question 5:

<html>\n

Congratulations. You've downloaded the file lab2-1.html!\n

</html>\n

Exercise 4:

Question 1:

No, there is no “IF-MODIFIED-SIENCE” line in first GET request

Question 2:

Yes, the response contains “Last-Modified: Tue, 23 Sep 2003 05:35:00 GMT”

It also contains Etag:”1bfef-173-8f4ae900”

Question 3:

Yes, there is. The second GET request has “IF-Modified-Since: Tue, 23 Sep 2003 05:35:00 GMT”

It also has “If-None-Match:”1bfef-173-8f4ae900”” which fit the first GET request received Etag

Question 4:

The server return status code 304 and phrase “Not Modified”

The server does not return the file, because it was not modified after Tue, 23 Sep 2003 05:35:00 GMT and the Etag is same, which shows the file is same. It will use local copy instead of downloading it, because they are the same

Question 5:

The Etag is 1bfef-173-8f4ae900, which is same as the first GET request’s response. It is used to check if the file has been changed. The same file version has same Etag. When the Etag is same, it means the files are same, no need to download it again, you can use the local version.

Exercise 5:

Code:

```
import argparse
import random
import socket
import time
from statistics import mean

def now_ms():
    return int(time.time() * 1000)

parser = argparse.ArgumentParser(description="UDP Ping Client")
parser.add_argument("host", type=str, help="server host")
parser.add_argument("port", type=int, help="server UDP port")
args = parser.parse_args()
server_addr = (args.host, args.port)
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.settimeout(0.6)

seq = random.randint(40000, 50000)

# count result
rtts_ms = []
results = []
first_send_ms = None
end_time_ms = None
for i in range(15):
    send_timestamp_ms = now_ms()
    if first_send_ms is None:
        first_send_ms = send_timestamp_ms
    message = f"PING {seq} {send_timestamp_ms}\r\n"
    data = message.encode("utf-8")
    # receive
    try:
        sock.sendto(data, server_addr)
        recv_data, recv_addr = sock.recvfrom(2048)
        recv_time_ms = now_ms()
        rtt = recv_time_ms - send_timestamp_ms
        clean = recv_data.rstrip(b"\x00").decode("utf-8", errors="replace").strip()
        rtts_ms.append(rtt)
        results.append(f"PING to {args.host}, seq={seq}, rtt={rtt} ms")
    # lost
    except socket.timeout:
```

```
            results.append(f"PING to {args.host}, seq={seq}, rtt=timeout")
        end_time_ms = now_ms()
        seq += 1
sock.close()
for line in results:
        print(line)
received = len(rtts_ms)
lost = 15 - received
packet_loss_pct = (lost / 15) * 100.0
if received >= 1:
        min_rtt = min(rtts_ms)
        max_rtt = max(rtts_ms)
        avg_rtt = int(round(mean(rtts_ms)))
else:
        min_rtt = max_rtt = avg_rtt = None
if first_send_ms is not None and end_time_ms is not None:
        total_tx_ms = end_time_ms - first_send_ms
else:
        total_tx_ms = 0
if received >= 2:
        diffs = [abs(rtts_ms[i] - rtts_ms[i - 1]) for i in range(1, received)]
        jitter = sum(diffs) / (received - 1)
        jitter = int(round(jitter))
else:
        jitter = None
print(f"\nPacket loss: {int(round(packet_loss_pct))}%")
if received >= 1:
        print(f"Minimum RTT: {min_rtt} ms, Maximum RTT: {max_rtt} ms, Average RTT:
{avg_rtt} ms")
else:
        print("Minimum RTT: N/A, Maximum RTT: N/A, Average RTT: N/A")
print(f"Total transmission time: {total_tx_ms} ms")
if jitter is not None:
        print(f"Jitter: {jitter} ms")
else:
        print("Jitter: N/A")
```

Client Result:

z5641211@vx08:~/Desktop/UNSW-COMP-9331/lab2$ python3 PingClient.py 127.0.0.1 9000
PING to 127.0.0.1, seq=46419, rtt=19 ms

PING to 127.0.0.1, seq=46420, rtt=timeout
PING to 127.0.0.1, seq=46421, rtt=6 ms
PING to 127.0.0.1, seq=46422, rtt=timeout
PING to 127.0.0.1, seq=46423, rtt=119 ms
PING to 127.0.0.1, seq=46424, rtt=27 ms
PING to 127.0.0.1, seq=46425, rtt=timeout
PING to 127.0.0.1, seq=46426, rtt=99 ms
PING to 127.0.0.1, seq=46427, rtt=180 ms
PING to 127.0.0.1, seq=46428, rtt=timeout
PING to 127.0.0.1, seq=46429, rtt=146 ms
PING to 127.0.0.1, seq=46430, rtt=92 ms
PING to 127.0.0.1, seq=46431, rtt=157 ms
PING to 127.0.0.1, seq=46432, rtt=99 ms
PING to 127.0.0.1, seq=46433, rtt=20 ms

Packet loss: 27%
Minimum RTT: 6 ms, Maximum RTT: 180 ms, Average RTT: 88 ms
Total transmission time: 3367 ms
Jitter: 66 ms

Server Result:

z5641211@vx08:~/Desktop/UNSW-COMP-9331/lab2$ javac PingServer.java
z5641211@vx08:~/Desktop/UNSW-COMP-9331/lab2$ java PingServer 9000
Received from 127.0.0.1: PING 46419 1759787383906
    Reply sent.
Received from 127.0.0.1: PING 46420 1759787383925
    Reply not sent.
Received from 127.0.0.1: PING 46421 1759787384525
    Reply sent.
Received from 127.0.0.1: PING 46422 1759787384531
    Reply not sent.
Received from 127.0.0.1: PING 46423 1759787385132
    Reply sent.
Received from 127.0.0.1: PING 46424 1759787385251
    Reply sent.
Received from 127.0.0.1: PING 46425 1759787385278
    Reply not sent.

Received from 127.0.0.1: PING 46426 1759787385879
    Reply sent.
Received from 127.0.0.1: PING 46427 1759787385978
    Reply sent.
Received from 127.0.0.1: PING 46428 1759787386158
    Reply not sent.
Received from 127.0.0.1: PING 46429 1759787386759
    Reply sent.
Received from 127.0.0.1: PING 46430 1759787386905
    Reply sent.
Received from 127.0.0.1: PING 46431 1759787386997
    Reply sent.
Received from 127.0.0.1: PING 46432 1759787387154
    Reply sent.
Received from 127.0.0.1: PING 46433 1759787387253
    Reply sent.

All related files can be visited at :

https://github.com/tanglehunter00/UNSW-COMP-9331/tree/main/lab2