DNS Client Implementation Report

Technical Explanation:

Creating the DNS packet:
I assigned a random 16-bit identifier for the Transaction ID. I set the flags to 0x0100 as its standard. Set question count to 1, answer count, authority records, and additional records are all set to 0. I constructed the header with struct.pack.

For the question part I split the domain name on dots, then I took the length of each part and prefixed it to it as a byte format. Lastly I joined each part together and added a null byte at the end of it. EX. "www.example.com" becomes `[3]www[7]example[3]com[0]`

In the send_query method I constructed the query by joining the header and question. A UDP socket was used to send the query to a public DNS resolver and get a response.

Parsing DNS responses:
I extracted the 6 header fields with struct.unpack. I took note of the answer count to use later. I made an offset pointer and set it to 12 to skip over the header, since the header is 12 bytes in length.

I also skipped the question section by following the length bytes until reaching a null byte, taking compression in consideration by checking for bytes starting with 11 i.e. 0xC0 and incrementing the offset by 2 when detected. Lastly I increment the offset by 5 to skip the null byte, qtype, and qclass.

In the answer section I skip the name field, also taking compression into account. I extracted record type, class, Time to Live, rdlength (resource data length). I used the offset : offset+(size of fields in bytes “10”).

Based on record type, extract and format data:

- A: Convert bytes to dotted decimal IP address
- CNAME/NS: Extract domain names with helper method _extract_name
- MX: Extract preference value and mail server name
- AAAA: Format IPv6 address
- TXT: Extract text data

DNS Compression Handling:

```
if (response[offset] & 0xC0) == 0xC0:
    pointer = ((response[offset] & 0x3F) << 8) | response[offset + 1]
```

This code:

- Checks if the top 2 bits are set to 0xC0 (11000000).
- Mask out those bits with 0x3F (00111111) to get the top 6 bits of the pointer.
- Shifts left by 8 bits to make room for lower 8 bits.
- Combines with the next byte using byte OR to form the complete 14-bit offset

when the first two bits of a byte are set to 0xC0, pointer is the calculated offset

```
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| 1  1|            OFFSET                       |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

HTTP Request:

After getting the IP address from the DNS Client, Establish connection with a TCP socket, then make a HTTP request with the IP address. Send the request with the host header since the IP address might host multiple websites. Get repose and print its status and reason.

External Resources Cited:

Domain Names Implementation and Specification

Used for understanding DNS packet structure and encoding requirements.

Results Explanation:

```
Domain Name     Record Type     Record Data     RTT (ms)
---------------------------------------------------------------------
www.google.com  A       142.250.189.196 14.07
HTTPS Response for www.google.com: 200 OK 61.46 ms
www.wikipedia.org       CNAME   dyna.wikimedia.org      24.66
www.wikipedia.org       A       198.35.26.96    24.66
HTTPS Response for www.wikipedia.org: 301 Moved Permanently 16.55 ms
```