Socket Programming: Domain Name System (DNS)

Student 1: Christine Li | 916857224 | A03
Student 2: Minh-Tu Nguyen | 917003682 | A03

A. Implement DNS Client:

1. Describe in detail the DNS request and response header format in your implementation.

   The DNS request contains the following sections: header and question. Each section has its own components. The header section contains the following: ID, QR, OPCODE, AA, TC, RD, RA, Z, RCODE, QDCOUNT, ANCOUNT, NSCOUNT, and ARCOUNT. The flag content (QR, OPCODE, AA, TC, RD, RA, Z, RCODE) is changed to a string and extended to fit its respective number of bits. It is then altered into a hexadecimal format to follow the 2 byte restriction. ID, QDCOUNT, ANCOUNT, NSCOUNT, and ARCOUNT are also changed into a string and transformed to hexadecimal. All header values are then put into one single string of bits (called request) in the following order: ID + Flags + QDCOUNT + ANCOUNT + NSCOUNT + ARCOUNT.

   The question section contains the following: QNAME, QTYPE, and QCLASS. QNAME is the given domain name represented in hexadecimal format. The domain name is separated into different components by the period. For example, tmz.com would be separated into tmz and com. Then the loop finds the length of each address component, transforms the address component to a hexadecimal format, adds these values to the request string of bits, and repeats. Once all components have been added, the terminating value “00” is added to the request string to indicate the end of QNAME. QTYPE and QCLASS are then changed into hexadecimal format and added to the request string.

   The answer, authority, and additional sections do not contain any content (as ANCOUNT, NSCOUNT, and ARCOUNT all equal 0) and are not included in the request.

   The request string then serves as the message that is sent to the public DNS resolver and a response is received. The response is a string of bits that needs to be parsed. The DNS response contains the following sections: header, question, resource records (answer, authority, and additional).

   For the header section, the first 24 bits of the response corresponds to the header components, i.e. ID is the first 4 bits, flags are the next 4 bits, QDCOUNT is the next 4 bits, etc. The flag bits are reverted to a binary format in which the bits correspond to a specific flag. The value of QDCOUNT determines how entries are in the question section, ANCOUNT determines how many resource records (rr) are in the answer

section, NSCOUNT determines how many rr are in the authority records section, ARCOUNT determines how many rr re in the additional records section.

Then the question section is parsed. Since QNAME occurs right after the header section, the length of the first component of the domain name is found. Then the first part of the domain name is parsed, altered into ascii form, and added to the domain name string along with a period (since the period separates the domain components). Then this process is repeated until we reach the terminating value. QTYPE and QCLASS are then parsed.

The resource records are parsed. The first resource record section is answer (followed by authority and additional). Note: As each resource record follows the same format, all sections are parsed the same way. Only the answer section will be described here. The first four bits correspond to the name in which it is in a compressed format. The next four are for type followed by class. Then TTL is the next 8 bits which determines how long this result is stored in the cache. The next four bits correspond to the RDLength which indicates how many bytes the RDData (IP address) is. Then the RDLength determines how many bytes to parse through and each byte corresponds to one segment of the IP address that is separated by periods. This parsing is repeated for all resource records.

Now all data from the response is extracted.

2. Compute the RTT between your DNS client to each of the public DNS resolvers. Do you notice any meaningful differences across different DNS resolvers? Explain.

   RTT is computed by subtracting the time when client first sends a message to the DNS resolver by the time when the client receives a message from the resolver. The value is then multiplied by 1000 to get the result in milliseconds.

   USA RTT: 47.085 milliseconds
   Canada RTT: 112.376 milliseconds
   Iran RTT: Error with connection (times out)

   Iran RTT was not able to be obtained due to timing out with no response acquired. The USA RTT is much faster than the Canada RTT as Canada takes more than double the amount of time to return a response in comparison to the USA. Distance between the DNS client and the DNS server (resolver) affects the round trip time. It will return a response slower the farther apart the client and the server is. As such, Canada's public DNS resolver is farther than the USA DNS resolver from our current location, leading to Canada's RTT to be higher than the USA's.

3. Compute the RTT between your HTTP client to the HTTP server of the resolved hostname.

   RTT is computed by subtracting the time when the HTTP client first sends the message to the HTTP server by the time when the client receives a message from the server. The value is then multiplied by 1000 to get the result in milliseconds.

   USA RTT: 248.985 milliseconds
   Canada RTT: 365.199 milliseconds
   Iran RTT: Error with connection (times out)

B. Implement DNS Server:

1. Compute the RTT from your local DNS server to each of the DNS servers including the root name server, the TLD name server, and the authoritative DNS server of tmz.com.

   RTT is computed by subtracting the time when the local DNS server first sends a message to the respective DNS server by the time when the client receives a message from the server. The value is then multiplied by 1000 to get the result in milliseconds.

   Root name server RTT: 29.725 milliseconds
   TLD name server RTT: 88.564 milliseconds
   Authoritative name server RTT: 42.6478 milliseconds

C. Implement DNS Server with Caching:

1. Report the time it takes to resolve each of these host names from your local DNS server.

   Time is computed by adding all RTT values from Root, TLD, and Authoritative name servers that were found for each of these host names.

   youtube.com: 175.688 milliseconds
   facebook.com: 154.055 milliseconds
   tmz.com: 169.634 milliseconds
   nytimes.com: 153.742 milliseconds
   cnn.com: 147.937 milliseconds

2. Report the TTL value in the DNS responses to each of these host names.

   The TTL is found by parsing the DNS responses, focusing on the TTL of the HTTP Server IP address. The TTL hex value is changed into decimal form.

youtube.com: 300 seconds
facebook.com: 300 seconds
tmz.com: 172800 seconds
nytimes.com: 300 seconds
cnn.com: 3600 seconds

3. Report the time it takes to resolve each of these host names by your DNS client from your local DNS server when it did implement the cache (and the answers are already in the cache).

   The time was calculated by subtracting the start time at the moment after the user prompts for the IP address and the end time when we retrieved the IP from the cache.

   youtube.com: 65.2471milliseconds
   facebook.com: 30.418 milliseconds
   tmz.com: 33.798 milliseconds
   nytimes.com: 108.153 milliseconds
   cnn.com: 31.851 milliseconds