DOMAIN NAME SYSTEM (DNS)

Computer Network Project Development

4

Target

In this project, DNS server will accept requests for IPv6 addresses and serves them either from its own cache or by querying servers higher up the hierarchy. Each transaction consists of at most four messages: one from the client to my server, one from my server to the upstream server, one from the upstream server to my server, and one from my sever to the client. The middle two can be sometimes skipped if my server cache some of the answers.

In a DNS system, the entry mapping a name to an IPv6 address is called a AAAA (or "quad A") record. Its "record type" is 28. DNS server will also keep a log of its activities. This is important for reasons such as detecting denial-of-service attacks, as well as allowing service upgrades to reflect usage patterns. For the log, a text version of the IPv6 addresses will be printed. IPv6 addresses are 128 bits long. They are represented in text as eight colon-separated strings of 16-bit numbers expressed in hexadecimal. As a shorthand, a string of consecutive 16-bit numbers that are all zero may be replaced by a single "::".

Background

The Domain Name System (DNS) provides, among other things, the mapping between human-meaningful hostnames like lms.unimelb.edu.au and the numeric IP addresses that indicate where packets should be sent. DNS consists of a hierarchy of servers, each knowing a portion of the complete mapping.

Non-blocking

DNS server can sometimes take a while for the server that was queried to give a response. To perform a recursive DNS lookup, many servers may need to be contacted (when starting from an empty cache, from a root DNS server down to an authoritative DNS server); any of them may be congested. Meanwhile, another request may have arrived, which the server cannot respond to if it was blocked by the completion of the first request.

Enabling the processing of new requests while waiting for prior requests to complete has been implemented using multi-threading. Using multithreading will require explicit locking of shared resources, such as the cache, whereas a single threaded implementation will not require.

Standard Function

Server can accept a DNS "AAAA" query over TCP on port 8053, and forward it to a server whose IPv4 address is the first command-line argument and whose port is the second command-line argument. Send the response back to the client who sent the request, over the same TCP connection. There will be a separate TCP connection for each query/response with the client. The server will log these events.

DNS message over TCP is slightly different from that over UDP: it has a two-byte header that specify the length (in bytes) of the message, not including the two-byte header itself. This means that you know the size of the message before you read it, and can malloc() enough space for it.

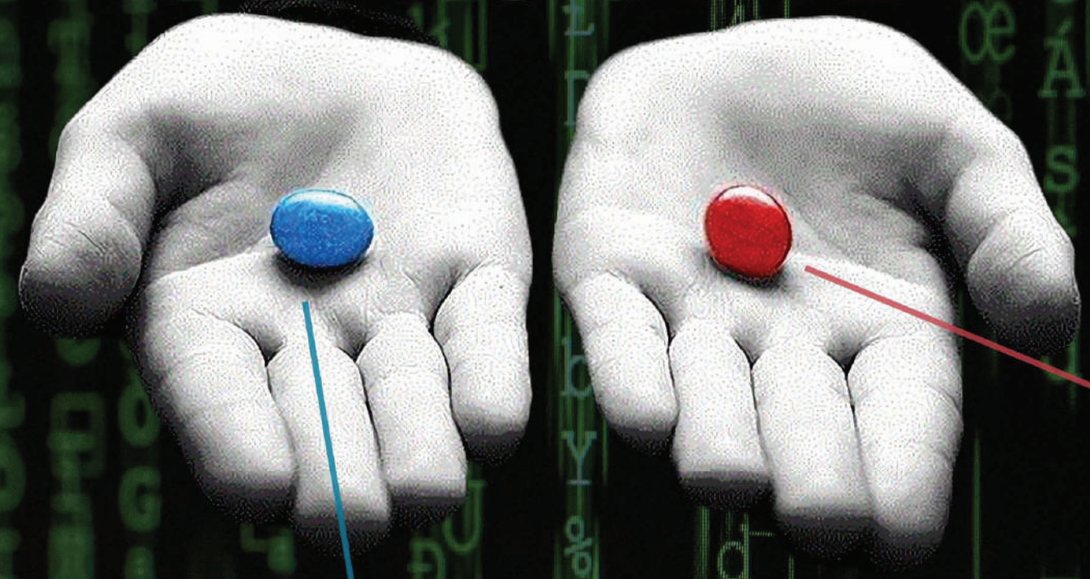
Moreover, the program will be ready to accept another query as soon as it has processed the previous query and response. The server will not shut down by itself. SIGINT (like CTRL-C) will be used to terminate your server between test cases.

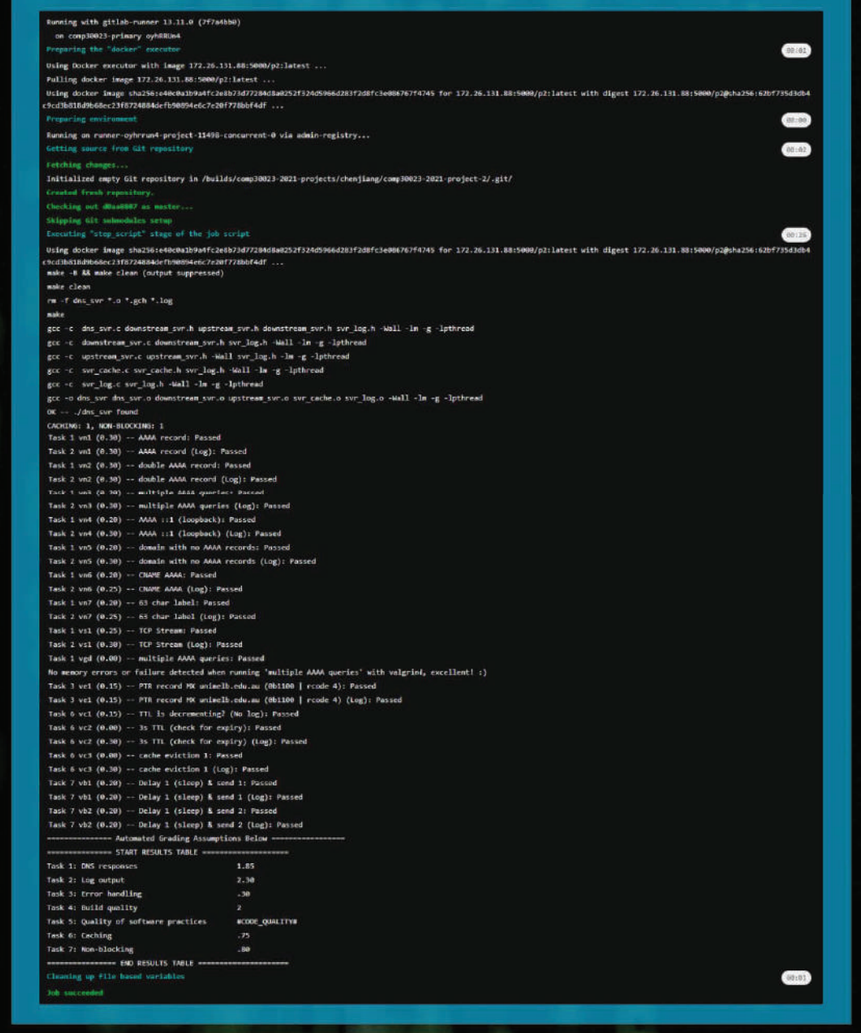
Cache

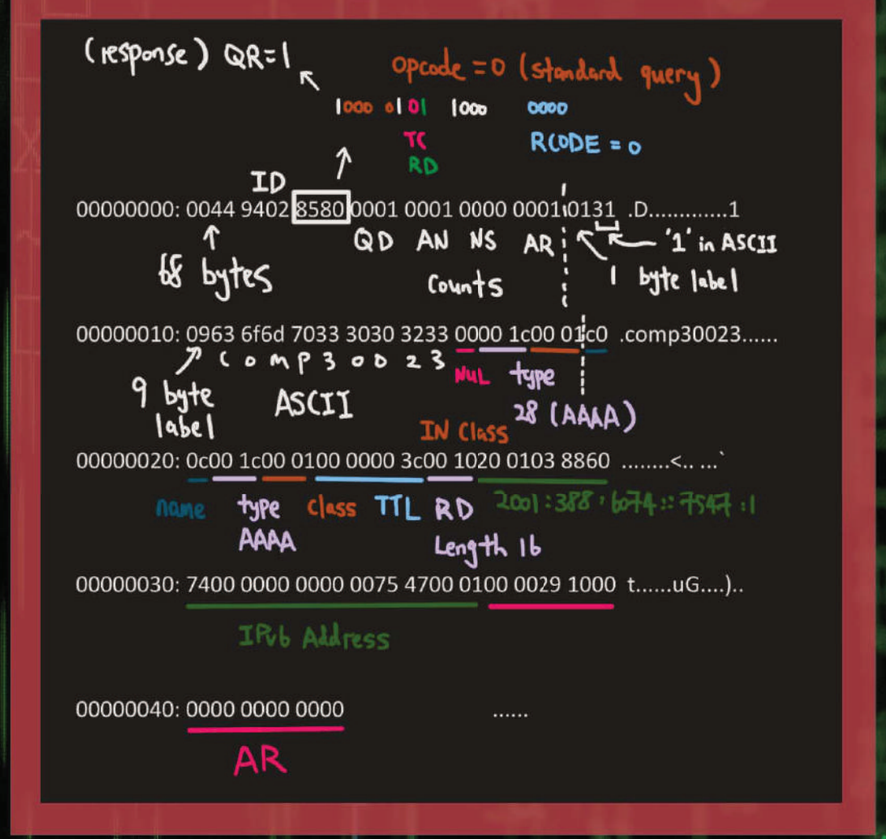
Cache keeps the five (5) most recent answers to queries. Answer directly, if possible, instead of querying the server again. The ID field of the reply will be changed for each query so that the client doesn't report an error. The time-to-live (TTL) field of the reply will also be decremented by the amount of time which has passed.

Moreover, cache entries can expire. A cache entry will not be used once it has expired and a new query should be sent to the recursive server. If a new answer is received when the cache contains an expired entry, the expired entry (or another expired entry) will be overwritten. If there is no expired entry, any cache eviction policy is allowed. In addition, cache dose not response that the query does not contain answers (i.e., the address was not found). If multiple answers are returned, only cache the first one.

OUTCOME







CODE: github.com/chenjiang0819/Process-Scheduling