Computer Networks HW#6

Submitted By: Neeha Hammad

Problem 6.1

(a)

hammad@hammad-Lenovo-V330-15IKB:~$ **dig grader.eecs.jacobs-university.de AAAA**

```
; <<>> DiG 9.11.3-1ubuntu1.11-Ubuntu <<>> grader.eecs.jacobs-university.de AAAA
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6920
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;grader.eecs.jacobs-university.de. IN    AAAA

;; ANSWER SECTION:
grader.eecs.jacobs-university.de. 3600 IN CNAME    cantaloupe.eecs.jacobs-university.de.
cantaloupe.eecs.jacobs-university.de. 3599 IN AAAA 2001:638:709:3000::29
cantaloupe.eecs.jacobs-university.de. 3599 IN AAAA 2001:638:709:300::29

;; Query time: 2 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sat Apr 18 11:13:42 GMT 2020
;; MSG SIZE  rcvd: 142
```

Unlike examples online, my trace command doesn't give a detailed output due to a firewall issue.

ha@hammad-Lenovo-V330-15IKB:~$**dig +trace grader.eecs.jacobs-university.de AAAA**

```
; <<>> DiG 9.11.3-1ubuntu1.11-Ubuntu <<>> +trace grader.eecs.jacobs-university.de AAAA
;; global options: +cmd
;; Received 51 bytes from 127.0.0.53#53(127.0.0.53) in 0 ms
```

An AAAA resource record contains an IPv6 address encoded in 16 bytes in network byte order. So here, we're resolving the website to IPv6. Dig reminds us of our initial question in the QUESTION SECTION. Then in the ANSWER SECTION, we're receiving the addresses of the grader website (001:638:709:3000::29). The CNAME resource record contains a character string preceded by the length of the string encoded in the first byte.

First, we, as the DNS client, query the recursive resolver (127.0.0.53) for grader.eecs.jacobs-university.de. The recursive resolver queries the root name server for grader.eecs.jacobs-university.de and asks for AAAA records. The root name server refers the recursive resolver to the .de Top-Level Domain (TLD) authoritative server. The recursive resolver then queries the .de TLD authoritative server for the grader website.

Next, it receives 1.2.3.4 as the answer. The recursive resolver caches the answer for the duration of the time-to-live (TTL) specified on the record, and returns it back to us.

We could also use this separately to check the nameservers (NS):

dig grader.eecs.jacobs-university.de NS

(b) SRV or service records are used for service discovery. An SRV record typically defines a symbolic name and the transport protocol used as part of the domain name. It defines the priority, weight, port, and target for the service in the record content.

It is defined in RFC 2782.

Format of SRV RR, whose DNS type code is 33:

_Service._Proto.Name TTL Class SRV Priority Weight Port Target

- Proto (symbolic name of the desired protocol, with an underscore)
- Name (the domain this RR refers to)
- TTL (Time to Live)
- Class (Protocol-specific namespace. SRV records occur in the IN Class.
- Port (the port on this target host of this service).
- Target (the domain name of the target host).

- Priority (the priority of this target host: A client must try to contact the target host with the lowest-numbered priority it can reach. Target hosts with the same priority should be tried in an order defined by the weight field. The range is 0-65535. This is a 16 bit unsigned integer in network byte order.

- Weight (a server selection mechanism). It specifies a relative weight for entries with the same priority. Larger weights have a higher priority of being selected.

Example:

_h323cs._tcp.vc.example.com. 86400 IN SRV 10 10 1720 px01.vc.example.com.

(c) HTTP is not one of the protocols that require SRV records in order to determine where the service is located. Generally, there's no browser that would check for SRV records so we can't achieve much by adding them. If we find a way around it then there might be a few advantages but they come with severe security threats.

Disadvantages:

- HTTP has to preserve backwards compatibility with the current mechanisms. Anyone using SRV for HTTP would still have to provide an alternate mechanism anyways. Of course, no one wants to maintain two ways of doing the same thing.
- Since SRV allows redirection of traffic to any port/server along with bypassing firewalls, this can be a huge security threat. Manually blacklisting certain ports or servers becomes almost impossible.

Advantages:

- SRV allows easy service discovery.
- SRV allows us to specify a different port for the service. SRV removes the restriction of using only port 80, which is often blocked by ISPs. Using SRV could be useful if the user wants to run a service on a different machine.
- We can have multiple systems for the same (or the same system and multiple instances on a different port). In simpler words, a single name can be resolved into multiple hosts. We can then order them using weights and priority. Even if there's an issue with one of them, the next one can be used.
- SRVs can also help bypass several firewalls and network restrictions.

(d) EDNS0 (Extension mechanism for DNS) is used for expanding the size of several parameters of the Domain Name System protocol, which had size restrictions that the Internet engineering community deemed too limited for increasing functionality of the protocol. For example, it provides extra data space for flags and return codes. It is currently defined in RFC 6891.

Coming to OPT:

- **Class:** Requestor's UDP payload size. It's field type is u_int16_t. It indicates the maximum size of the sender's UDP payload.

- **TTL:** This 32 bit field is divided into 3 fields:

- Extended Response Code: Adds 8 bits to the 4-bit R-Code in the DNS message header to provide 12 bits total.
- EDNS Version Number
- Extended Header Flags: DO (which indicate the usage of DNSSEC) & Z (which are usually set to 0 and are designed for future usage).

(e)

```
hammad@hammad-Lenovo-V330-15IKB:~$ dig @1.1.1.1 instagram.com AAAA

; <<>> DiG 9.11.3-1ubuntu1.11-Ubuntu <<>> @1.1.1.1 instagram.com AAAA
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14317
;; flags: qr rd ra; QUERY: 1, ANSWER: 8, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1452
;; QUESTION SECTION:
;instagram.com.                 IN    AAAA

;; ANSWER SECTION:
instagram.com.          60    IN    AAAA    2406:da00:ff00::23a9:2ad1
instagram.com.          60    IN    AAAA    2406:da00:ff00::34c9:6a60
instagram.com.          60    IN    AAAA    2406:da00:ff00::3401:6e34
instagram.com.          60    IN    AAAA    2406:da00:ff00::22ec:8517
instagram.com.          60    IN    AAAA    2406:da00:ff00::23ad:9cc7
instagram.com.          60    IN    AAAA    2406:da00:ff00::23a8:5bba
instagram.com.          60    IN    AAAA    2406:da00:ff00::22ec:b613
instagram.com.          60    IN    AAAA    2406:da00:ff00::34c9:1620

;; Query time: 10 msec
;; SERVER: 1.1.1.1#53(1.1.1.1)
;; WHEN: Sat Apr 18 17:22:18 GMT 2020
;; MSG SIZE  rcvd: 279
```

```
hammad@hammad-Lenovo-V330-15IKB:~$ dig @1.1.1.1 instagram.com A

; <<>> DiG 9.11.3-1ubuntu1.11-Ubuntu <<>> @1.1.1.1 instagram.com A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51804
;; flags: qr rd ra; QUERY: 1, ANSWER: 8, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1452
;; QUESTION SECTION:
;instagram.com.                 IN    A

;; ANSWER SECTION:
instagram.com.          23    IN    A    3.214.32.78
instagram.com.          23    IN    A    3.214.138.217
instagram.com.          23    IN    A    3.216.235.82
instagram.com.          23    IN    A    3.229.97.56
instagram.com.          23    IN    A    3.210.81.177
instagram.com.          23    IN    A    3.218.3.143
instagram.com.          23    IN    A    3.224.3.80
instagram.com.          23    IN    A    3.223.46.130

;; Query time: 35 msec
;; SERVER: 1.1.1.1#53(1.1.1.1)
;; WHEN: Sat Apr 18 23:00:04 GMT 2020
;; MSG SIZE  rcvd: 183



hammad@hammad-Lenovo-V330-15IKB:~$ dig @1.1.1.1 amazon.com AAAA
; <<>> DiG 9.11.3-1ubuntu1.11-Ubuntu <<>> @1.1.1.1 amazon.com AAAA
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25606
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1452
```

;; QUESTION SECTION:
;amazon.com. IN AAAA

;; AUTHORITY SECTION:
amazon.com. 32 IN SOA dns-external-master.amazon.com. root.amazon.com. 2010125071 180 60 3024000 60
;; Query time: 10 msec
;; SERVER: 1.1.1.1#53(1.1.1.1)
;; WHEN: Sat Apr 18 21:18:52 GMT 2020
;; MSG SIZE rcvd: 110

Website	DNS Server	A	AAAA
instagram.com	1.1.1.1	8 entries	8 entries
	8.8.8.8	8 entries	8 entries
	9.9.9.9	8 entries	8 entries
amazon.com	1.1.1.1	3 entries	0 entries
	8.8.8.8	3 entries	0 entries
	9.9.9.9	3 entries	0 entries

When we query with A, we get IPv4 addresses while with AAAA, we get IPv6 addresses. When checking for amazon.com, we don't get any ip addresses for AAAA and instead, get a SOA (Start of Authority) record. This holds administrative information about the DNS zone. We can conclude that amazon.com does not support IPv6.

For instagram.com, A & AAAA values were different for all three dns servers. The time to live (TTL) values were also different.

I also observed that A and AAAA values aren't always in the same order. It's possible to receive the same values in a different order when you retry the same commands.

Problem 6.2:

(a) A mDNS is used to resolve host names to IP addresses on small computer networks without local name servers. It allows us to perform DNS-like operations on the local link in the absence of any conventional Unicast DNS server. Basically, mDNS designates a portion of the DNS namespace to be free for local use. It eliminates the need of any additional configurations or fees.

It is defined in RFC 6762 (February 2013). Multicast DNS requires little or no lengthy set up configurations or administrations. More importantly, they can also work during any infrastructure failures. Moreover, multicast DNS can usually work without any extra infrastructure. This is not the case in regular DNS.

Few other key differences :
Multicast DNS…

* Allows larger UDP packets
* Uses UDP port 5353 instead of port 53
* Has no Start of Authority (SOA) records
* Uses only UTF-8 to encode resource record names
* Allows names up to 255 bytes along with a terminating zero byte
* Allows name compression in rdata for SRV and other record types
* Allows more than one question in a query message
* Doesn't require the question to be repeated in the response message
* Uses unsolicited responses to announce new records
* Uses DNS RR TTL 0 to indicate that a record has been deleted
* Monitors queries to perform Duplicate Question Suppression
* Monitors responses to perform Duplicate Answer Suppression...

(b) DNS-SD is a way of using standard DNS programming interfaces, servers, and packet formats to browse the network for services. It is defined in RFC6763 and is compatible with both, regular DNS and mDNS.

It allows clients to discover a named list of service instances using a query for a DNS PTR (pointer record which resolves an IP address to a host name) with a name of the form "<Service>.<Domain>". This returns a set of zero or more names. These names depict the DNS SRV/TXT record pairs.

In SRV records, the first label of the pair is an underscore character followed by the Application Protocol Name. The second label is either "_tcp" or "_udp". These SRV records can resolve to the domain name where the instance comes from.

TXT records hold additional information like configuration parameters. In order to connect to a certain device or find the domain/host name, clients can resolve A/AAAA records using standard DNS queries.

References:

https://ns1.com/blog/using-dig-trace
https://www.heise.de/netze/rfc/rfcs/rfc6762.shtml
https://support.dnsimple.com/articles/srv-record/
https://www.ietf.org/rfc/rfc2782.txt
https://serverfault.com/questions/31060/will-srv-records-ever-become-usefull
https://en.wikipedia.org/wiki/Extension_mechanisms_for_DNS
https://tools.ietf.org/html/rfc6762#section-19
http://distributedfog.com/discovery/dns-sd/
http://files.dns-sd.org/draft-cheshire-dnsext-dns-sd.txt
https://books.google.de/books?id=0TwtDwAAQBAJ&pg=PA45&lpg=PA45&dq=opt+class+field&source=bl&ots=j2JUrldj6H&sig=ACfU3U18oAraTlMoroLz9whFLsquilJ92w&hl=en&sa=X&ved=2ahUKEwj8pbngu_LoAhUMuaQKHWroA3kQ6AEwAXoECAgQKQ#v=onepage&q=opt%20class%20field&f=false