Computer Networks
Jacobs University Bremen
Romelda Blaceri

Course: CO20-320301
5 May, 2020

Homework 6

Problem 6.1

Solution:

a) The process of the domain name being resolved to an IPv6 address is as follows:

* First a DNS query is fired to the DNS recursive solver asking for AAAA records for the domain that we asked: `grader.eecs.jacobs-university.de`.

* Then, we have one server proceed with the query of de, which is the top level of the domain, and then continues to the next subdomain `jacobs-university.de` and so on until it finds exactly one match.

* Then the server finds out that the domain contains a CNAME record pointing to `cantaloupe.eecs.jacobs-university.de` and it receives an AAAA recording pointing to `2001:638:709:3000::29` which is the IPv6 address we were looking for.

Running `dig grader.eecs.jacobs-university.de AAAA` results in this output:

```
; <<>> DiG 9.11.5-P4-5.1ubuntu2.1-Ubuntu <<>> grader.eecs.jacobs-university.de AAAA
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41446
;; 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: 1 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Di Mai 05 11:19:04 CEST 2020
;; MSG SIZE  rcvd: 142
```

First thing that is noticed, is the saying that `EDNS` specification was used. Then we can see under the `QUESTION SECTION` that we asked for the domain `grader.eecs.jacobs-university.de.` in AAAA, and under the `ANSWER SECTION` what I explained in the last steps of the look up process above.

b) The SRV resource record is defined in RFC 2782 and a sample SRV record has the following form: `_service._proto.name.` TTL class SRV priority weight port target in which `service` refers to the symbolic name of the service that is desired, `proto` refers to the transport protocol of the service above (usually either TCP or UDP), `name` refers to the domain name for which the record is valid, and it ends in a dot, `TTL` refers to the standard DNS Time To Live field, `class` refers to the standard DNS class field (which is always IN), `priority` refers to the priority of the target host (lower value means more preferred),

`weight` is a relative weight for records with the same priority (higher value means higher chance of getting picked), `port` refers to the TCP or UDP port on which the service is to be found, and lastly, `target` refers to the the canonical hostname of the machine providing the service(ends in a dot).
An example would be this:

```
\_sip.\_tcp.example.com. 86400 IN SRV 10 20 5060 randomname.com.
```

The difference between the priority and weight field, is that a client will first try to target the host with the lowest priority and only in case it comes across 2 entries with same priority, then it chooses the one with the highest weight.

c) **Pros:**

* SRV resource record is also really useful on HTTP, as it allows websites to be served on any ports and not just port 80. It is especially useful to serve websites on home ISP networks as some ISP will block 80 to save upbound traffic. It is also able to bypass certain censorship on the network and firewalls.

* Web services are more configurable because of the HTTP SRV resource record which can redirect traffic to different domains under different port numbers.

* The design of SRV makes it possible to easily achieve client-side load balancing, which can save a lot of infrastructures cost for website owners.

* Since service discovery is the key to microservice architecture, SRV resource record is very useful as it can be used for service discovery for distributed systems.

Cons:

* The usage of redirect traffic to any server on any port can be abused and exploited to launch denial of service attack.

* Non-deterministic visit to the host to open a website can be also casued using SRV. It is hard to gain access information and statistics, as well as to pinpoint failure servers. Security issues can be caused as the internal infrastructures are exposed.

* Security vulnerability can be caused as when serving HTTP on any ports via SRV resource record, HTTP traffic might bypass firewalls.

d) With the growth that the internet is currently having, the introduction to IPv6 became neccessary for the long term health of the internet. In the same way it became necessary to have more bytes in DNS messages, and that is what EDNS0 (E stands from 'Extension' mechanisms) makes possible. It is defined in RFC6891 and allows for messages that surpass 512 bytes of size. It also provides extra data space for flags. The CLASS field refers to the requestor's UPD payload size and the TTL field is used to store extended RCODE and flags.

e) Below you can see the results for bing.com:

IPv4 DNS Address	A	AAAA
1.1.1.1	204.79.197.200	2620:1ec:c11::200
8.8.8.8	204.79.197.200	2620:1ec:c11::200
9.9.9.9	204.79.197.200	2620:1ec:c11::200

From the table above it can be seen that I got the same result from all 3 address.

Problem 6.2

Solution:

a) `mDNS` is defined in `FC6762` and is a protocol is able to look up a DNS resource record in absence of a conventional managed DNS server. As for the differences with respect to DNS protocol semantics, when an mDNS client needs to resolve a hostname, it sends an IP multicast query message that asks the host having that name to identify itself. The message is sent to the IPv4 address `224.0.0.251` or IPv6 address `ff02::fb`. That target machine will then multicasts a message that includes its IP address. All machines in that subnet can then use that information to update their mDNS caches. nDNS also supports continuous querying.

b) DNS-based service discovery specifies how DNS resource records are named and structured to facilitate service discovery and is defined in RFC6763. There are 2 types of record that can be used, TXT and SRV. The former has a lot of flexibility as it stores key-value pairs and is able to provide more information than just port number and IP address. SRV on the other side does not provide more info, and it has a fixed form of instance.servive.doman.

References:

https://en.wikipedia.org/wiki/SRV_record
https://en.wikipedia.org/wiki/Multicast_DNS