Computer Networks
Jacobs University Bremen
Dragi Kamov

Course: CO20-320301
May 10th, 2019

Homework 6

(The following homework was discussed with Brian Sherif, and Fezi Manana)

Problem 6.1

Solution:

a) Trace with "dig" can be done using the following command:

```
$ dig +trace grader.eecs.jacobs-university.de AAAA
```

Where AAAA records is specific for IPv6 records. And we get the following results:

```
; <<>> DiG 9.10.6 <<>> +trace grader.eecs.jacobs-university.de AAAA
;; global options: +cmd
.           377629  IN  NS  d.root-servers.net.
.           377629  IN  NS  k.root-servers.net.
.           377629  IN  NS  e.root-servers.net.
.           377629  IN  NS  f.root-servers.net.
.           377629  IN  NS  h.root-servers.net.
.           377629  IN  NS  g.root-servers.net.
.           377629  IN  NS  c.root-servers.net.
.           377629  IN  NS  l.root-servers.net.
.           377629  IN  NS  m.root-servers.net.
.           377629  IN  NS  i.root-servers.net.
.           377629  IN  NS  j.root-servers.net.
.           377629  IN  NS  b.root-servers.net.
.           377629  IN  NS  a.root-servers.net.
.           414429  IN  RRSIG   NS 8 0 518400 20190522050000 20190509040000 25266 . yB
;; Received 1097 bytes from 10.70.0.20#53(10.70.0.20) in 45 ms

de.         172800  IN  NS  a.nic.de.
de.         172800  IN  NS  f.nic.de.
de.         172800  IN  NS  l.de.net.
de.         172800  IN  NS  n.de.net.
de.         172800  IN  NS  s.de.net.
de.         172800  IN  NS  z.nic.de.
de.         86400   IN  DS  39227 8 2 AAB73083B9EF70E4A5E94769A418AC12E887FC3C0875EF20
de.         86400   IN  RRSIG   DS 8 1 86400 20190523050000 20190510040000 25266 . VEf
;; Received 738 bytes from 198.97.190.53#53(h.root-servers.net) in 142 ms

jacobs-university.de.   86400   IN  NS  dns.iu-bremen.de.
jacobs-university.de.   86400   IN  NS  www.jacobs-utils.de.
H319DM5GC3EDEK691VQBHEHOT7VGGJ2B.de. 7200 IN NSEC3 1 1 15 BA5EBA11 H31EGRUDRBMFSM3HAQ6
NS SOA RRSIG DNSKEY NSEC3PARAM
H319DM5GC3EDEK691VQBHEHOT7VGGJ2B.de. 7200 IN RRSIG NSEC3 8 2 7200 20190517160105 20190
SFAC58VBFNB14JPD7N2H5MOLE1O2L213.de. 7200 IN NSEC3 1 1 15 BA5EBA11 SFAEPI3LFEOB8NDDLF1
A RRSIG
SFAC58VBFNB14JPD7N2H5MOLE1O2L213.de. 7200 IN RRSIG NSEC3 8 2 7200 20190517160105 20190
;; Received 611 bytes from 195.243.137.26#53(s.de.net) in 19 ms

grader.eecs.jacobs-university.de. 3600 IN CNAME cantaloupe.eecs.jacobs-university.de.
cantaloupe.eecs.jacobs-university.de. 3600 IN AAAA 2001:638:709:3000::29
eecs.jacobs-university.de. 3600 IN  NS  dns.jacobs-university.de.
eecs.jacobs-university.de. 3600 IN  NS  ns.eecs.jacobs-university.de.
eecs.jacobs-university.de. 3600 IN  NS  ns1.ibr.cs.tu-bs.de.
;; Received 890 bytes from 212.201.44.22#53(dns.iu-bremen.de) in 2 ms
```

The steps of the following lookup are:

1. DNS query gets sent to the DNS recursive resolver provided by DHCP on our network.
2. DNS resolver on our network returns 13 root servers. **g.root-servers.net** is used with the query of **de**
3. 6 nameservers responsible for **de.** domains are received, **s.de.net** is then selected for the next subdomain **jacobs-university.de**
4. 2 nameserver are received **www.jacobs-utils.de** and **dns.iu-bremen.de** the selected nameserver is **www.jacobs-utils.de**
5. 3 nameservers are received, from which **dns.jacobs-university.de** is used to query the AAAA (IPv6 address)
   It is found that **grader.eecs.jacobs-university.de** contains a CNAME record that leads to **cantaloupe.eecs.jacobs-university.de** in which we receive the following information **3600 IN AAAA 2001:638:709:3000::29**

(Method and explanation by Yiping Deng)

b) The format of a SRV resource record looks like the following:

service	Service + Protocol + Domain
TTL (Time to live)	Specifies how long this RR (Resource Record) may be kept in the cache
IN	Internet
SRV	The string "SRV"
priority	if several identical services are offered, the lowest priority takes precedence (the higher priority services serve as replacements in case of failure)
mass	within the same priority, the probability of choosing a service should be relative to the weight (for a service with weight 3 and one with weight 2, 60% should be used on average for the purpose of load distribution)
port	TCP or UDP port number
server	Server providing this service (it must not be an IP address or an alias, ie a domain with a CNAME RR)

an example of a SRV resource is:
_ldap._tcp.example.net. 3600 IN SRV 0 0 389 phoenix.example.net. 2
It is used to identify servers that host specific services with a specific protocols. It also offers Load Balancing, where traffic can be divided onto servers that are sharing the same services depending on their priority and weight.

c) No, it is not a good idea to use as the cons outweigh the pros.

Advantages: The load balancing provided through SRV will lead to security issues. Example: Redirecting traffic to another domain or port can be abused using denial of service attacks.
DNS spoofing can also be achieved as we can redirect to different servers of port numbers.

Disadvantages: Better control over web services, including the ability to have load balancing and redirecting traffic according to priority and weight number.

d) EDNS0 is defined in RFC 2671
It is the extension of the size of the parameters in the DNS protocol. This provides the ability to advertise more information and further enhance security and capabilities.
This allows DNS clients to advertise 4096 bytes of UDP packets and Extended RCODE used together with the original RCODE contains a version field indicating the lowest implemented level.

e) Using "dig" for **google.com** we get the following:

IPv4 addresses	Description	A	AAAA
1.1.1.1	Cloudflare's DNS resolver	172.217.18.174	2a00:1450:4001:817::200e
8.8.8.8	Google's DNS resolver	172.217.18.110	2a00:1450:4001:809::200e
9.9.9.9	Quad9's DNS resolver	172.217.22.46	2a00:1450:4001:821::200e

The same for **amazon.com** we get the following:

IPv4 addresses	Description	A	AAAA
1.1.1.1	Cloudflare's DNS resolver	205.251.242.103 176.32.98.166 176.32.103.205	N/A
8.8.8.8	Google's DNS resolver	205.251.242.103 176.32.103.205 176.32.98.166	N/A
9.9.9.9	Quad9's DNS resolver	205.251.242.103 176.32.98.166 176.32.103.205	N/A

The "dig" command is the following:

```
$ dig @[IPv4 address] [server] [A/AAAA]
```

, an example would look like:

```
$ dig @1.1.1.1 google.com AAAA
```

From the output we can notice that **amazon.com** doesn't support IPv6 and that for IPv4 it has 3 different servers that respond. While **google.com** on the other hand supports IPv6, but has only one server responce from each.

Problem 6.2

Solution:

a) The goal of multicast DNS (mDNS) is to be able to implement the function of the DNS without central DNS servers. mDNS is nothing more than a description of how the client must proceed when DNS requests to multicast - address to send and how a group of computers thus bypasses so that the request is answered correctly and without increased load on the network. And it is defined in RFC6762.

b) DNS-based Service Discovery, or DNS-SD for short, is a mechanism which allows th client that is looking for a type of service on a domain where he is looking for it, using standard DNS queries. This nechanism is defined in RFC6763.

References:

http://www.rfc-editor.org/rfc/rfc6762.txt
http://www.rfc-editor.org/rfc/rfc6763.txt