Computer Networks

5. Application layer (DNS)

Boni García
http://bonigarcia.github.io/
boni.garcia@urjc.es

Departamento de Teoría de la Señal y Comunicaciones y Sistemas Telemáticos y Computación
Escuela Técnica Superior de Ingeniería de Telecomunicación
Universidad Rey Juan Carlos

2019/2020

Table of contents

1. Introduction
2. Domain names
3. DNS servers
4. DNS database
5. Name resolution
6. DNS messages
7. Takeaways

Table of contents

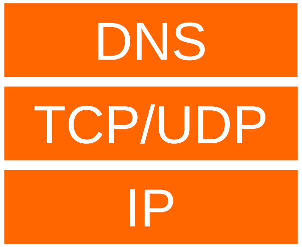
1. Introduction
   1. DNS motivation
   2. DNS architecture
   3. DNS service
2. Domain names
3. DNS servers
4. DNS database
5. Name resolution
6. DNS messages
7. Takeaways

1. Introduction - DNS motivation

- For final users (humans), it is easier to remember names than numbers
  - For instance, it is easier to remember google.es than 216.58.211.195
- DNS (Domain Name System) is an application protocol whose most important feature is to translate (*resolve*) readable names for humans (called **domain names**) into IP addresses and vice versa
  - DNS maps the domain google.es ⇄ 216.58.211.195
- DNS has been initially defined in the RFCs 1034 and 1035
- DNS is typically used by other applications (e.g. a web browser)

1. Introduction - DNS architecture

- DNS is a **client-server** application protocol
- The information handled by DNS is stored as a distributed **database** in a number of distributed **servers**
  - The most used DNS server is called BIND (Berkley Internet Name Domain), installed on UNIX or GNU/Linux systems
  - The default port in which servers listens to requests is 53
- DNS **clients** make requests to server to resolve domain names to IP address and vice versa
  - DNS clients are known as *resolvers* (implemented as library in the OS)
  - For example, the `host` command-line tool
  - Resolvers usually use UDP as transport layer
  - … except a special case (zone transfer) TCP is used



DNS

TCP/UDP

IP

1. Introduction - DNS service

- DNS provides 3 different services:

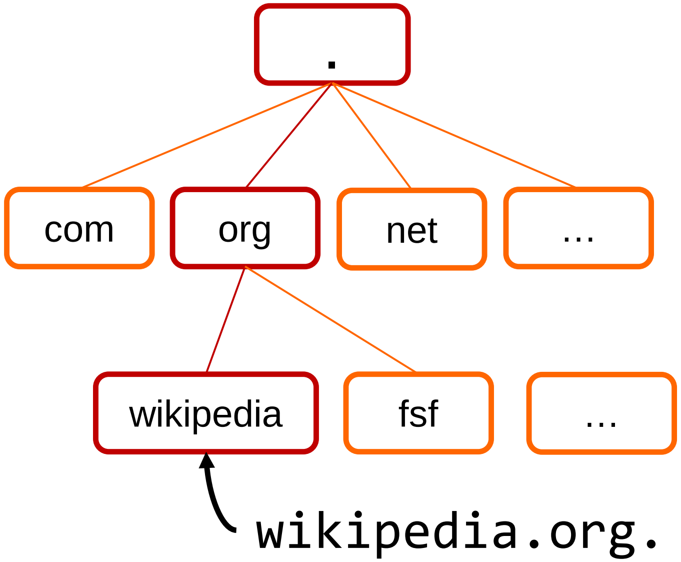
1. Name resolution:
   - Direct resolution: given a domain name, get the IP address
   - Reverse resolution: given an IP address, get the domain name
2. Alias. Pseudonym for domain names
   - For example, a domain called mydomain.com could have an alias www.mydomain.com (both domain names point to the same IP address)
3. Load distribution. DNS can be used to balance load to replicated servers (DNS Round Robin). This is useful for specially loaded servers (e.g. mail or web servers)
   - Round Robin is a method to select the elements in a group starting with the first element of the list until the last one in succession

Table of contents

1. Introduction
2. Domain names
   I. Structure
   II. Types of TLDs
   III. DNS bodies
   IV. IDN standard
3. DNS servers
4. DNS database
5. Name resolution
6. DNS messages
7. Takeaways

2. Domain names - Structure

- Domain names have a **hierarchical structure**:
  1. The top level of the hierarchy is called root and is represented by a dot (.)
  2. TLD (Top Level Domain): Identify the type of domain (.com, .org, ...)
  3. Domain: Unique name within the TLD
     - It can also contain sub-domains (for instance: es.wikipedia.org.)



- The FQDN (Fully Qualified Domain Name) consists of the concatenation of all the parts of a domain including the point
  - In the example before: `wikipedia.org.`

2. Domain names - Types of TLDs

- Country code (ccTLD). Used by a country or independent territory (2 letters): For example: .es, .us, .de, .fr, .uk, .jp, ...
  - Second level (SLD). Organizations within a country: .co.uk, .co.jp, ...
- Generic (gTLD). Used by a particular kind of organization. They have three or more letters. For example: .com (commercial), .org (initially non-profit organizations, today without limitation), .net (initially for network infrastructures, today without limitation) ...
- Sponsored (sTLD): There are rules to obtain for the domain. For example: .edu (educational purposes), .int (international organizations), ...
- Infrastructure. In this group there is a single TLD: .arpa. It is used in reverse resolution

https://www.iana.org/domains/root/db

2. Domain names - DNS bodies

- Root servers administration: **ICANN** (Internet Corporation for Assigned Names and Numbers)
  - www.icann.org
- TLD servers: **IANA** (Internet Assigned Numbers Authority)
  - www.iana.org
- Spanish domains: **red.es** (public entity dependent on the Ministry of Energy, Tourism, and Digital Agency)
  - www.dominios.es
  - The complete list of registry agents (called *registrars*) of domain .es can be visited in the URL:

    http://www.dominios.es/dominios/es/agentes-registradores/todos-los-agentes-registradores

2. Domain names - IDN standard

- Initially, the domain names were alphanumeric strings (with '-' as the only allowed symbol)
- IDN (Internationalized Domain Name) is an extension to DNS that allows (since 2005) that a domain name contains non-ASCII characters (even emojis)
- Examples: http://canción.com/, http://pequeñin.com/, https://i❤.ws/
- In IDN, instead of redefining the existing DNS infrastructure, what is done with non-ASCII domain names is to convert it to an ASCII-based form called **Punycode** (RFC 3492)
- Example: españa.es = xn--espaa-rta.es
- Punycode online converter: http://punycode.es/
- In practice these domains are not very common

Table of contents

1. Introduction
2. Domain names
3. DNS servers
    I. Types
    II. Root servers
4. Name resolution
5. DNS database
6. DNS messages
7. Takeaways

3. DNS servers - Types

- Depending on the hierarchy, we distinguish between:
  - **Root** servers. There are 13 root servers (labeled from A to M) replicated throughout the world
    - These severs know all TLD servers
  - Top Level Domain (**TLD**) server. Server for each of the zones .com, .es, .net, etc.
    - These severs know all next level servers in their zones
  - Second-level domain servers
  - Third-level domain servers
  - …

3. DNS servers - Types

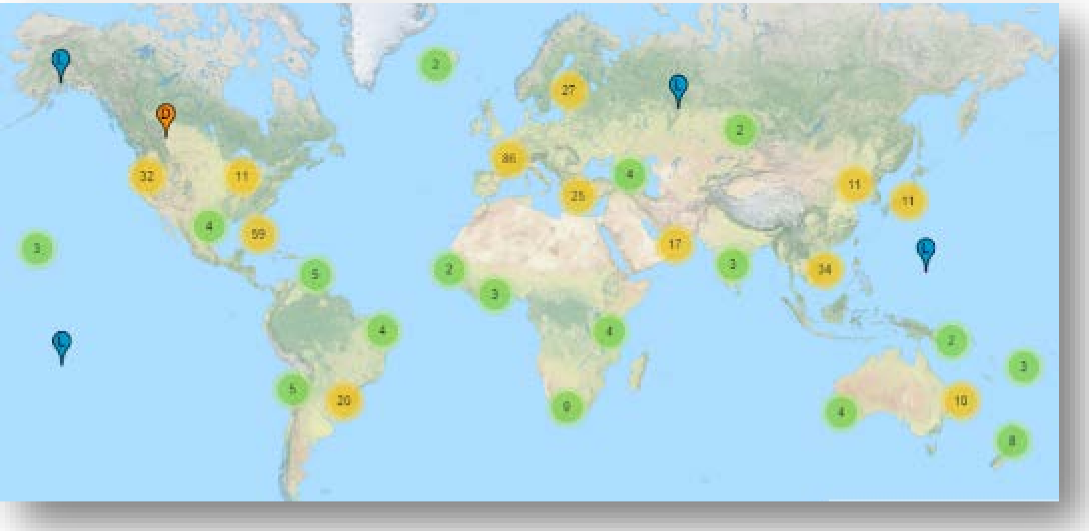
- Depending on the response provide by servers, we distinguish between:
  - Authoritative servers: These servers actually resolve the domain names in their area of authority. If not, it will return a list of servers to ask. There are two kinds:
  - Non-authoritative servers (also known as local servers): They are not able to perform name resolution by themselves and perform recursive requests (or use a cached value)

3. DNS servers - Types

- Depending on how the information is stored, we distinguish between:
  - Primary (master): Main copy of the zone information
  - Secondary (slave): Replica of the primary
- Zone transfer is the process by which the content of an authoritative server is copied from a primary (master) server to a secondary (slave) server
  - The messages exchanged in this process use TCP
  - A zone transfer happens in any of the following scenarios:
    - When there are changes in the main zone file
    - When starting the DNS service on the secondary server
    - When the expiration time is over

3. DNS servers - Root servers

- Root servers store a list of the domain names and IP addresses of all the TLD servers
- The nearest geographical DNS server is located
  - This type of traffic is called **anycast**, ant it is implemented thanks to BGP



http://www.root-servers.org/

3. DNS servers - Root servers

Host name	IP Address	Operator
a.root-servers.net	198.41.0.4	VeriSign, Inc.
b.root-servers.net	192.228.79.201	University of Southern California (ISI)
c.root-servers.net	192.33.4.12	Cogent Communications
d.root-servers.net	199.7.91.13	University of Maryland
e.root-servers.net	192.203.230.10	NASA (Ames Research Center)
f.root-servers.net	192.5.5.241	Internet Systems Consortium, Inc.
g.root-servers.net	192.112.36.4	US Department of Defense (NIC)
h.root-servers.net	128.63.2.53	US Army (Research Lab)
i.root-servers.net	192.36.148.17	Netnod
j.root-servers.net	192.58.128.30	VeriSign, Inc.
k.root-servers.net	193.0.14.129	RIPE NCC
l.root-servers.net	199.7.83.42	ICANN
m.root-servers.net	202.12.27.33	WIDE Project

Table of contents

1. Introduction
2. Domain names
3. DNS servers
4. DNS database
5. Name resolution
6. DNS messages
7. Takeaways

4. DNS database

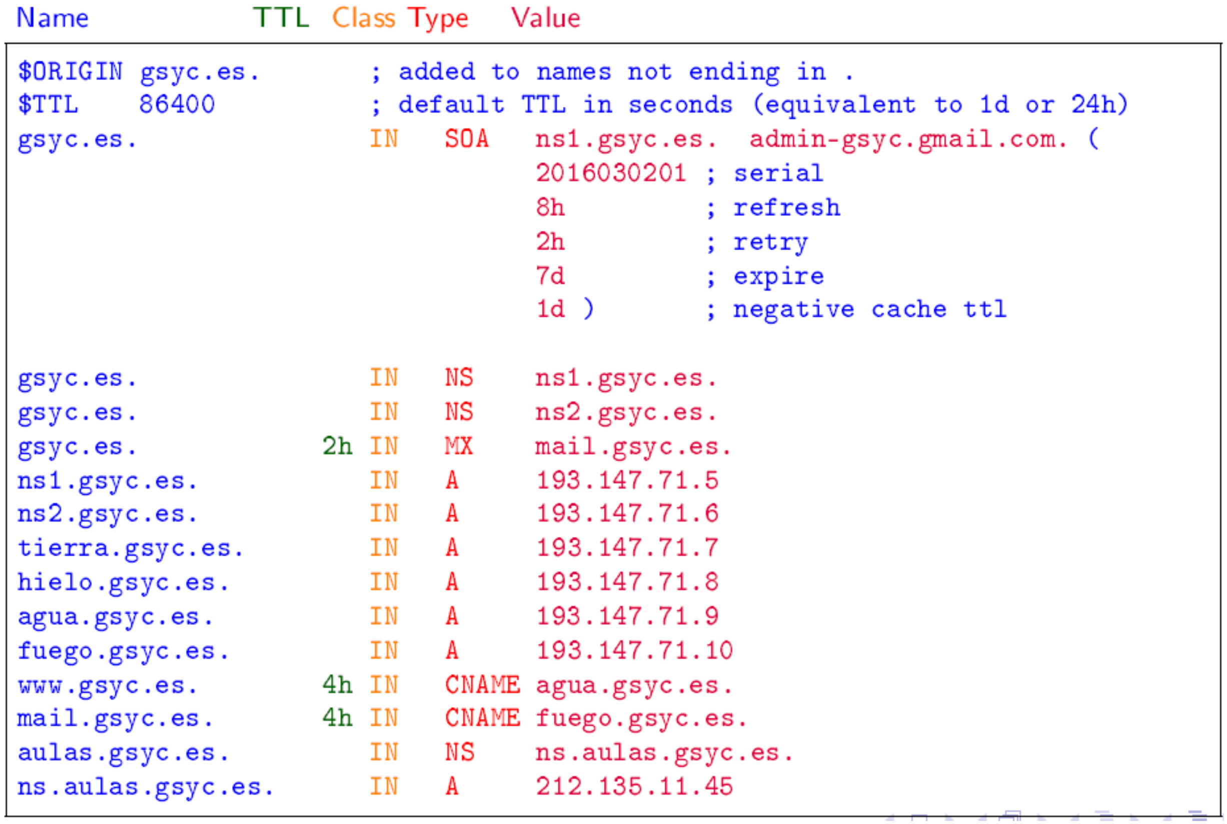
- The information handled by DNS is stored as a **distributed database**
- Each record in this database is called RR (**Resource Record**)
- Each RR has 5 fields:
  - Name: Name of the node (domain name or IP address)
  - TTL: Time that the RR is valid (by default in seconds)
  - Class: In practice the class is always IN (Internet)
  - Type: Kind of RR (see next slide)
  - Value: RR data

4. DNS database

- The types of RR registries are the following:
  - `SOA`: (Start of Authority): Configuration of the zone
  - `A`: Hostname for IPv4 address
  - `AAAA`: Hostname for IPv6 address
  - `NS`: DNS server
  - `MX`: Email server
  - `CNAME`: Alias of a host
  - `PTR`: Reverse translation (using the special domain `in-addr.arpa.`)

4. DNS database

- A set of RRs handled in a DNS server is called DNS map
- Example of a DNS map (for BIND servers):



```
Name                TTL  Class Type   Value

$ORIGIN gsyc.es.          ; added to names not ending in .
$TTL    86400             ; default TTL in seconds (equivalent to 1d or 24h)
gsyc.es.                  IN    SOA   ns1.gsyc.es.  admin-gsyc.gmail.com. (
                                      2016030201 ; serial
                                      8h         ; refresh
                                      2h         ; retry
                                      7d         ; expire
                                      1d )       ; negative cache ttl

gsyc.es.                  IN    NS    ns1.gsyc.es.
gsyc.es.                  IN    NS    ns2.gsyc.es.
gsyc.es.               2h IN    MX    mail.gsyc.es.
ns1.gsyc.es.              IN    A     193.147.71.5
ns2.gsyc.es.              IN    A     193.147.71.6
tierra.gsyc.es.           IN    A     193.147.71.7
hielo.gsyc.es.            IN    A     193.147.71.8
agua.gsyc.es.             IN    A     193.147.71.9
fuego.gsyc.es.            IN    A     193.147.71.10
www.gsyc.es.           4h IN    CNAME agua.gsyc.es.
mail.gsyc.es.          4h IN    CNAME fuego.gsyc.es.
aulas.gsyc.es.            IN    NS    ns.aulas.gsyc.es.
ns.aulas.gsyc.es.         IN    A     212.135.11.45
```

Table of contents

1. Introduction
2. Domain names
3. DNS servers
4. DNS database
5. Name resolution
   I. Direct resolution
   II. Reverse resolution
6. DNS messages
7. Takeaways

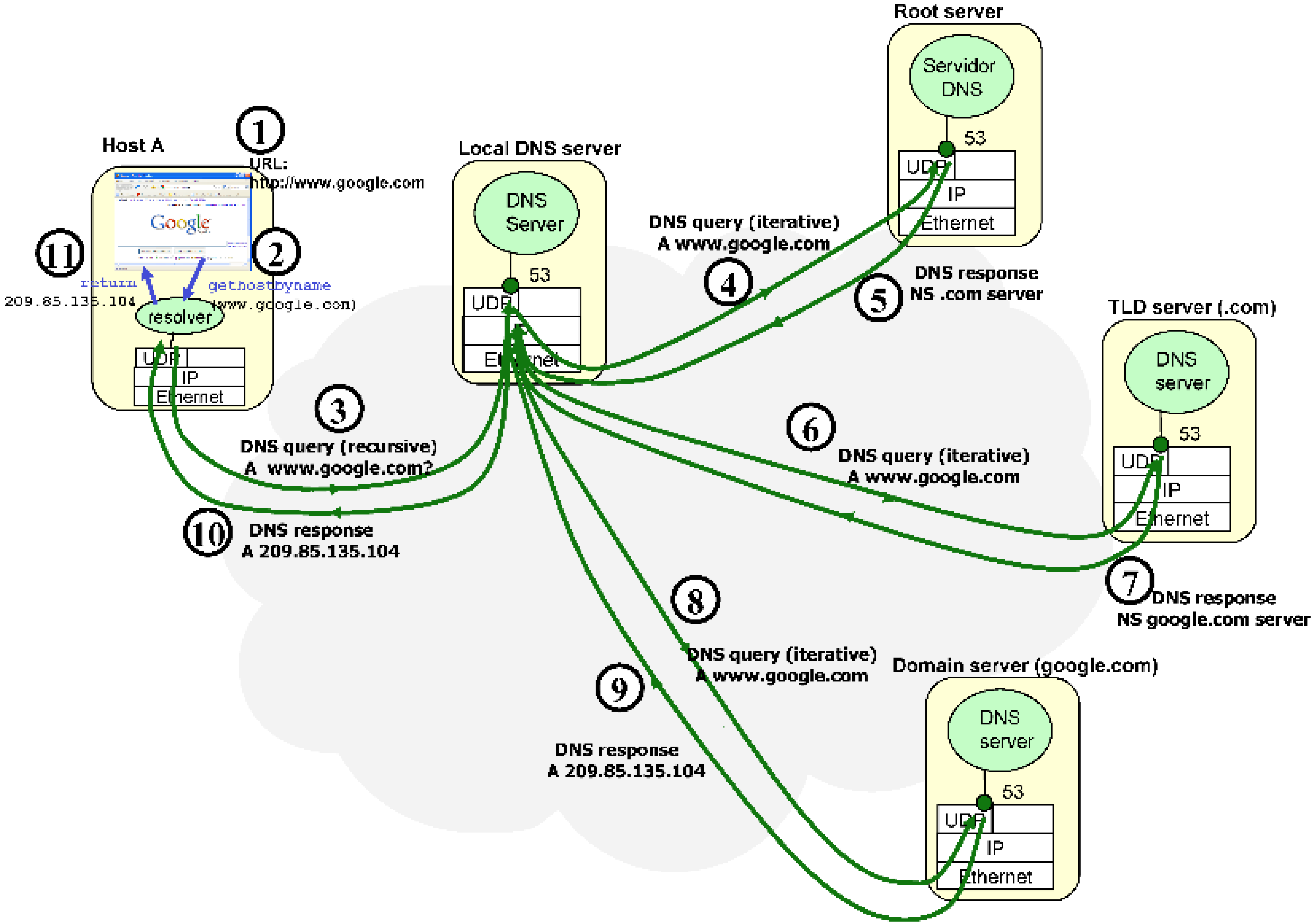
5. Name resolution

- There are two types of DNS queries (made by clients):

1. Recursive query
   - Server is forced to make all necessary queries to resolve a domain
   - This is the usually way in which DNS clients work
2. Iterative query
   - Server replies with the most accurate information about the name resolution (usually the IP address of the next sub-domain)
   - This is the usually way in which DNS servers work

- To improve performance, server maintain **caches** with resolved requests
  - Cache is updated when a server makes a resolution for the first time
  - Clients can also use caches, although it is not usual

5. Name resolution - Direct resolution



5. Name resolution - Reverse resolution

- The **.in-addr.arpa** domain is used for reverse resolution, mapping IP address to hostnames
  - This name has historical origins: it is an acronym for inverse addresses in the Arpanet (the predecessor to today's Internet)
- The elements of the reverse domain are the network addresses built by inverting the numbers that compose it, and ending in in-addr.arpa.
  - For example: the network 138.117.0.0 is the reverse domain 117.138.in-addr.arpa.
- Reverse RRs uses the type PTR
  - There is no technical requirement for PTR records
  - They were designed as a matter of convenience. However, they have become required by some security schemes

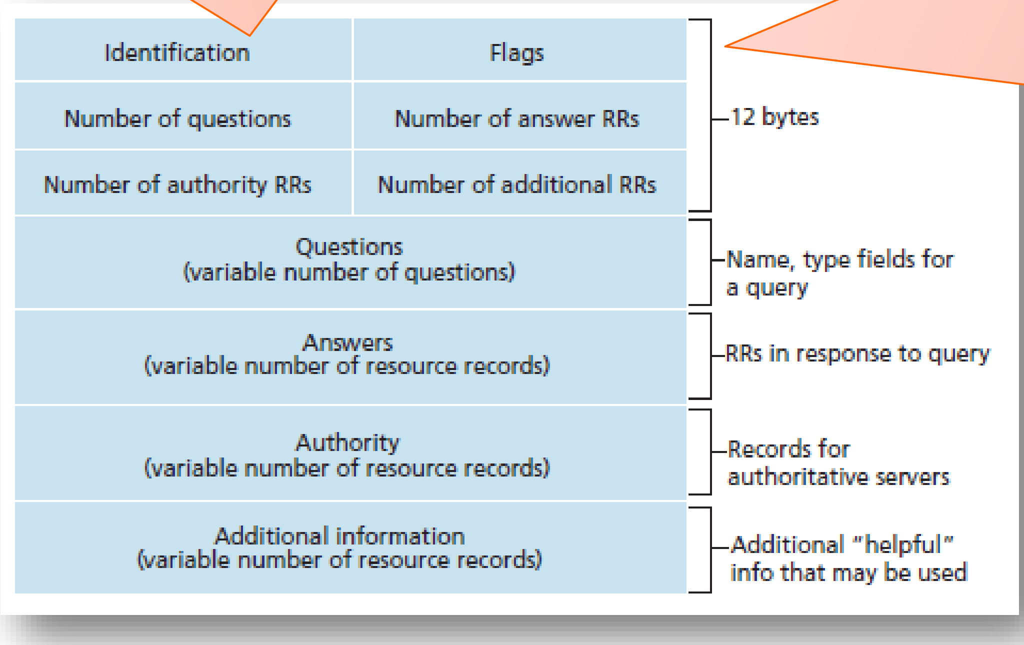
Table of contents

1. Introduction
2. Domain names
3. DNS servers
4. DNS database
5. Name resolution
6. DNS messages
7. Takeaways

6. DNS messages

- There are two main types: requests and responses
- Both types of messages have the same structure:

The identification is set in requests, and responses must set the same value



Identification	Flags
Number of questions	Number of answer RRs
Number of authority RRs	Number of additional RRs
Questions (variable number of questions)	
Answers (variable number of resource records)	
Authority (variable number of resource records)	
Additional information (variable number of resource records)	

12 bytes
Name, type fields for a query
RRs in response to query
Records for authoritative servers
Additional "helpful" info that may be used

Important flags:

- QR (1 bit):
  - 0 = request
  - 1 = response
- Opcode (4 bits):
  - 0 = standard query
  - 3 = NOTIFY (used in master to update slave)
  - 4 = UPDATE (used for dynamic update of the DNS map)
- RD (1 bit):
  - 0 = recursive request
  - 1 = iterative request

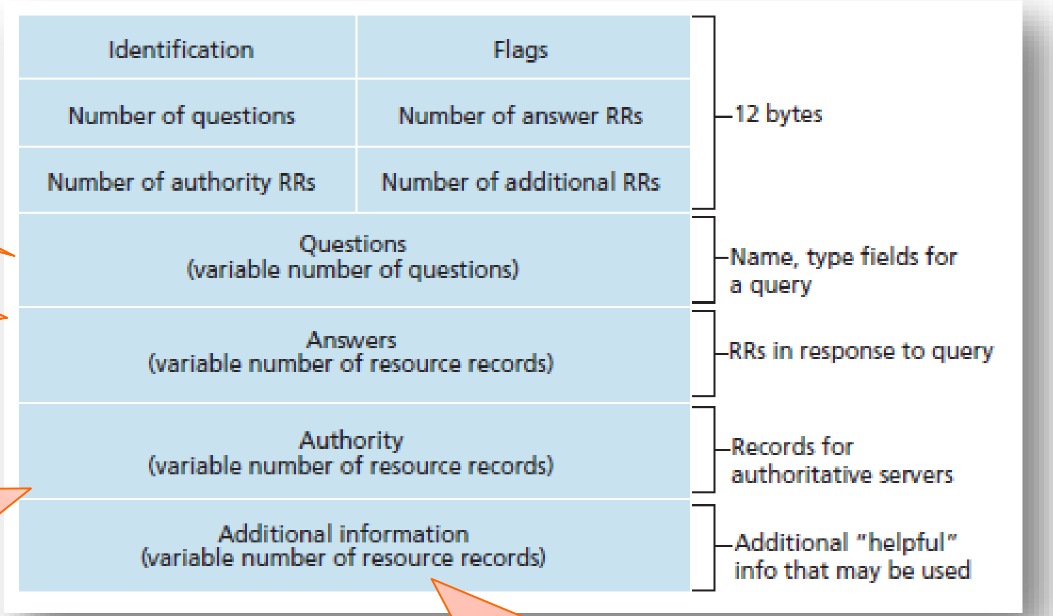
6. DNS messages

- There are two main types: requests and responses
- Both types of messages have the same structure:

Questions contain **name**, **type**, and **class**

Answers contain **name**, **type**, **class**, **TTL**, and **value** (data and length)

Authority contains the DNS servers which has resolved the request or the DNS server which can used to continue the iterative request.

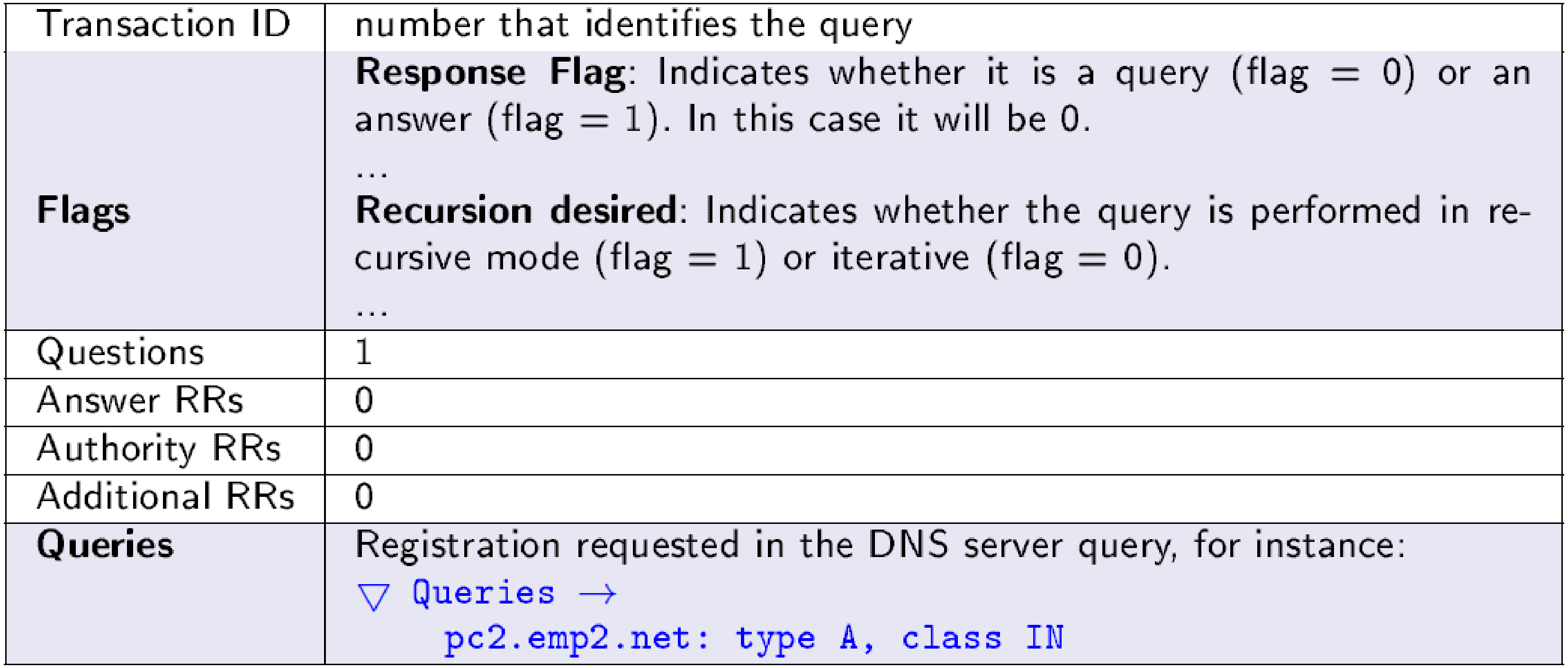


Identification	Flags	12 bytes
Number of questions	Number of answer RRs	
Number of authority RRs	Number of additional RRs	
Questions (variable number of questions)		Name, type fields for a query
Answers (variable number of resource records)		RRs in response to query
Authority (variable number of resource records)		Records for authoritative servers
Additional information (variable number of resource records)		Additional "helpful" info that may be used

Additional information is typically used to resolve IP address of DNS servers of the authority section

6. DNS messages

- Example: request message



Transaction ID	number that identifies the query
Flags	**Response Flag**: Indicates whether it is a query (flag = 0) or an answer (flag = 1). In this case it will be 0. ... **Recursion desired**: Indicates whether the query is performed in recursive mode (flag = 1) or iterative (flag = 0). ...
Questions	1
Answer RRs	0
Authority RRs	0
Additional RRs	0
Queries	Registration requested in the DNS server query, for instance: ▽ `Queries` → `pc2.emp2.net: type A, class IN`

6. DNS messages

- Example: response with the requested record

Transaction ID	same number that in the request message
Flags	**Response Flag**: Indicates whether it is a query (flag = 0) or an answer (flag = 1). In this case it will be 1. ...
Questions	1
Answer RRs	1
Authority RRs	1
Additional RRs	1

Queries	Copy of the DNS query, for instance:
	`▽ Queries →`
	`pc2.emp2.net: type A, class IN`

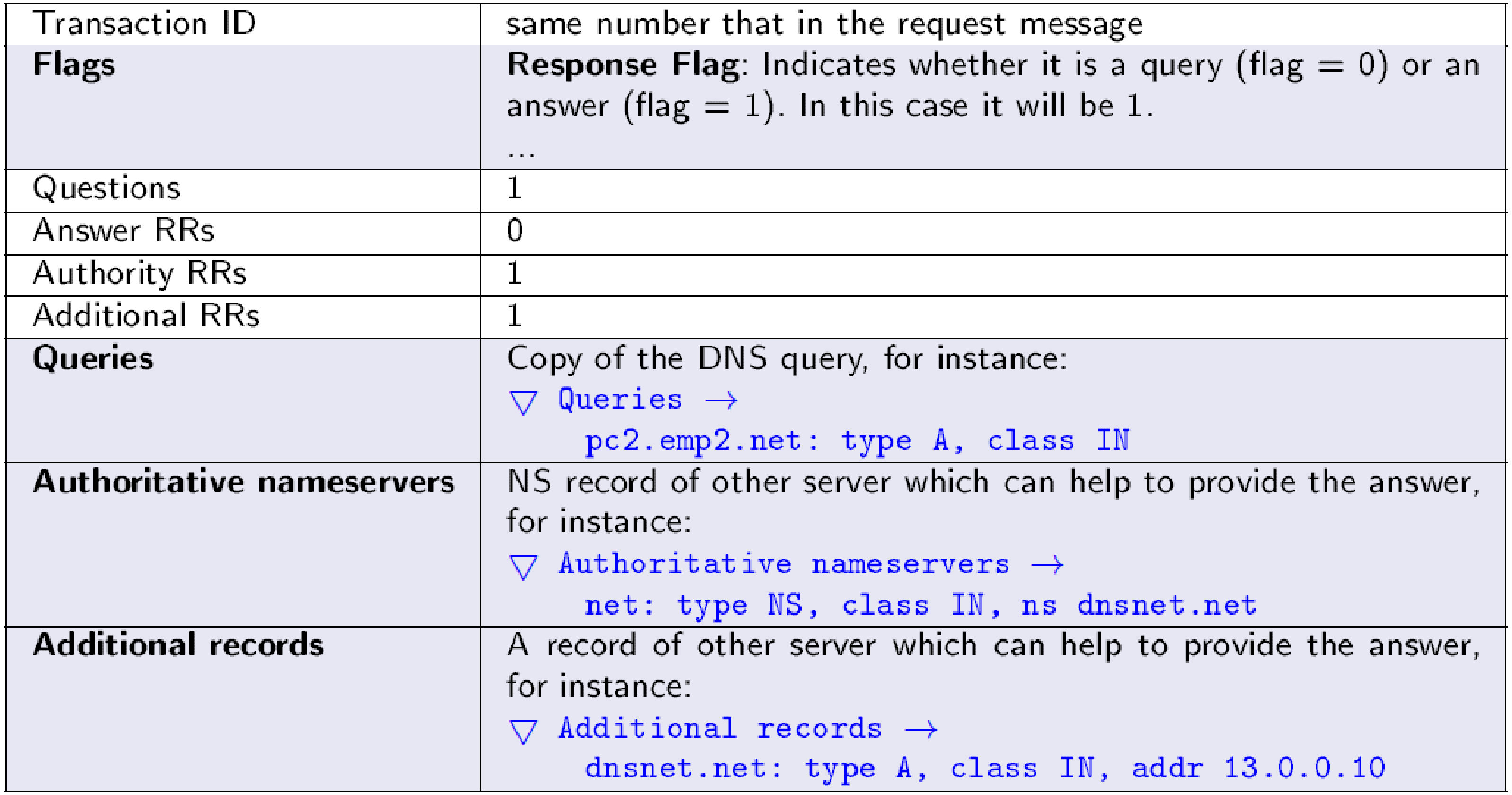
Answers	A record containing the answer, for instance:
	`▽ Answers →`
	`▽ pc2.emp2.net: type A, class IN, addr 14.0.0.100`
	`Name: pc2.emp2.net`
	`Type: A (Host address)`
	`Class: IN (0x0001)`
	`Time to live: 1 day`
	`Data length: 4`
	`Addr: 14.0.0.100`

Authoritative nameservers	NS record of the server which has provided the answer, for instance:
	`▽ Authoritative nameservers →`
	`emp2.net: type NS, class IN, ns dnsemp2.emp2.net`

Additional records	A record of the server which has provided the answer, for instance:
	`▽ Additional records →`
	`dnsemp2.emp2.net: type A, class IN, addr 14.0.0.10`

6. DNS messages

- Example: response without the requested record, redirecting to a different server



Transaction ID	same number that in the request message
Flags	**Response Flag**: Indicates whether it is a query (flag = 0) or an answer (flag = 1). In this case it will be 1. ...
Questions	1
Answer RRs	0
Authority RRs	1
Additional RRs	1
Queries	Copy of the DNS query, for instance: ▽ `Queries` → `pc2.emp2.net: type A, class IN`
Authoritative nameservers	NS record of other server which can help to provide the answer, for instance: ▽ `Authoritative nameservers` → `net: type NS, class IN, ns dnsnet.net`
Additional records	A record of other server which can help to provide the answer, for instance: ▽ `Additional records` → `dnsnet.net: type A, class IN, addr 13.0.0.10`

Table of contents

1. Introduction
2. Domain names
3. DNS servers
4. DNS database
5. Name resolution
6. DNS messages
7. Takeaways

7. Takeaways

- DNS (Domain Name System) is an application protocol whose most important feature is to translate (*resolve*) readable names for humans (called **domain names**) into IP addresses (direct resolution) and vice versa (reverse resolution)
- DNS have a **hierarchical structure** (names and servers):
  - Root → TLD → Domain server
- DNS info is stored as a distributed database:
  - Each record in this database is called **RR** (Resource Record)
  - A set of RRs handled in a DNS server is called **DNS map**
  - Relevant RR types are:
    - `SOA`: Configuration of the zone
    - `A`: Hostname for IPv4 address
    - `NS`: DNS server
    - `PTR`: Reverse translation (using the special domain in-addr.arpa.)