Networking - Part 2

Michel FACERIAS

Polytech - Université de Montpellier

16 décembre 2022

All numbers are written using a base coded pattern $\sum a_n b^n$:

- a_n is a digit of the n column ;
- b^n is the weight of a digit, b is the base.

Then :

- $435_{(10)} = 4x10^2 + 3x10^1 + 5x10^0$;
- $1101_{(2)} = 1x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 13_{(10)}$

A logical address is an identifier corresponding to layer 3 of the OSI model which is the **network** layer. It identifies hosts in a network :

- **unicast** address is used to specifically reach a particular host ;
- **multicast** address is used to reach a group of hosts ;
- **broadcast** address addresses all hosts at once.

In the TCP/IP protocol, it is the IP layer that carries the address parameter. It is named **IPv4 address** [1].

1. Throughout the rest of this course, we will use the term IP address to designate the address according to version 4 of the TCP/IP protocol as specified in RFC 791

An IPv4[2] address is set off 32 bits.
To be human-readable, we prefer to write it as a series of 4 decimal numbers, separated by dots.
Each of these numbers represents 8 bits[3] out of the 32. Its decimal value is therefore between 0 and 255 like bellow.

As decimal	125		.	12		.	12		.	3					
As binary	0111	1101	.	0000	1100	.	0000	1100	.	0000	0011				

2. IP version 4 is actually the most used network protocol. It will be gradually replaced by IP version 6. IPv6 is out of the scope of this course
3. To read easier, we separate in 2 blocks of 4 bits

An IP address is divide in **two parts**. A **netmask** is used to identify each part of the address. The netmask is also a set of 32 bits :

- the left part only contain **1** digits, this is the **network** part ;
- the right part only contain **0** digits, this is the **host** part.

```
Address (dec)      125     .      12     .      12     .       3
Address (bin)  0111 1101 . 0000 1100 . 0000 1100 . 0000 0011
                         <--------network----------------->   <-hosts->
Netmask (bin)  1111 1111 . 1111 1111 . 1111 1111 . 0000 0000
Netmask (dec)      255     .    255     .     255     .       0
```

The full IP address in this exemple could be written with netmask :

- as an IP address pattern : 125.12.12.3 / 255.255.255.0 ;
- as the number of bits of the network part : 125.12.12.3/24

A **network** address is calculated from an IP address and a netmask using a **bitwise AND**.

```
Adr (dec)     125     .     12      .      12     .       3

Adr (bin) 0111 1101 . 0000 1100 . 0000 1100 . 0000 0011 >-+
                                                         |
                                                         |
Msk (bin) 1111 1111 . 1111 1111 . 1111 1111 . 0000 0000 >-+
                                                         |
Msk (dec)     255     .    255      .     255     .       0      bit
                                                              AND
                                                               |
Net (bin) 0111 1101 . 0000 1100 . 0000 1100 . 0000 0000 <-+

Net (dec)     125     .     12      .      12     .       0
```

Here, it's **125.12.12.0** and this is the **lowest address** of the block when the **host part is set to 0**.
All hosts with the **same network address** belong to the **same IP network**.

This is the **range of addresses** that a host can **use in a network**.
The first is when the host part is set to 1.
The last is when the host part is set to its maximum (all bits at 1).

```
Net (dec)     125     .     12      .      12     .      0
Net (bin) 0111 1101 . 0000 1100 . 0000 1100 . 0000 0000
          <--------network----------------->   <-hosts->
Low (bin) 0111 1101 . 0000 1100 . 0000 1100 . 0000 0001
Low (dec)     125     .     12      .      12     .      1
          <--------network----------------->   <-hosts->
Hig (bin) 0111 1101 . 0000 1100 . 0000 1100 . 1111 1111
Hig (dec)     125     .     12      .      12     .    255
```

The **first host address** of the block is **125.12.12.1**
The **last host address** of the block is **125.12.12.255**

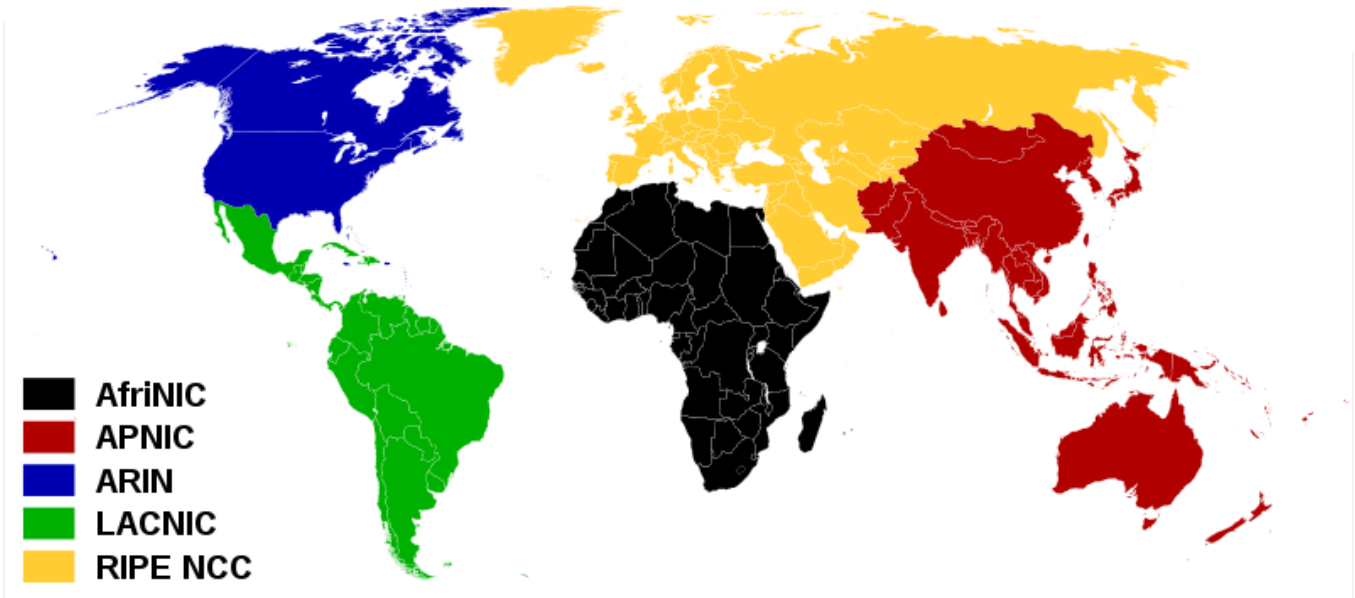
A **broadcast** address is used **to join all the host** that belong to your network.
The **default router** address (ie the gateway address) is the address of a special host **used to go out** of your network.
They are **choose randomly**. But in **best practices**, the **2 higher addresses** of the range are used.

```
Net (dec)          125     .     12      .      12     .      0
Net (bin)      0111 1101 . 0000 1100 . 0000 1100 . 0000 0000
                <--------network------------------>   <-hosts->
Dft  GW (bin)  0111 1101 . 0000 1100 . 0000 1100 . 1111 1110
Dft  GW (dec)      125     .     12      .      12     .    254
                <--------network------------------>   <-hosts->
Bcast  (bin)   0111 1101 . 0000 1100 . 0000 1100 . 1111 1111
Bcast  (dec)       125     .     12      .      12     .    255
```

In the **125.12.12.0/24** network :

- the broadcast should be **125.12.12.255/24** ;
- the default gateway should be **125.12.12.254/24**.

Public addresses are globally assigned by the **Internet Assigned Numbers Authority (IANA)** [4].
Each region of the world has its own **Regional Internet Registry**.
In Europe, the **Réseaux IP Européens (RIPE)** [5] is managing the addresses granted by IANA.



4. http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml
5. in french in the text - https://www.ripe.net

Private addresses can be used as desired. They are included in the following ranges :

- RFC 1918 - Address Allocation for Private Internets [6] ;
  - 10.0.0.0/8 - from 10.0.0.0 to 10.255.255.255
  - 172.16.0.0/12 - from 172.16.0.0 to 172.31.255.255
  - 192.168.0.0/16 - 192.168.0.0 to 192.168.255.255
- RFC 5737 - IPv4 Address Blocks Reserved for Documentation [7]
  - 192.0.2.0/24 - from 192.0.2.0 to 192.0.2.255
  - 198.51.100.0/24 - from 198.51.100.0 to 192.51.100.255
  - 203.0.113.0/24 - from 203.0.113.0 to 203.0.113.255

6. https://datatracker.ietf.org/doc/html/rfc1918
7. https://datatracker.ietf.org/doc/html/rfc5737

Each host has a routing table. We can have a look on it :

```
#ip route show
default via 192.168.100.254 dev wlo1 proto dhcp metric 600
192.168.100.0/24 dev wlo1 proto kernel scope link src
    192.168.100.101 metric 600
192.168.122.0/24 dev ens3 proto kernel scope link src
    192.168.122.254
```

This computer is linked with two networks via **wlo1** and **ens3** interfaces.
We can directly join hosts belonging to one of these network directly.
If we can not join directly a host, i/e because it belongs to an other network, we must entrust the message to the default gateway. It will forward our messages to the destination host.

Inter-network routing

When the host belongs to my network ?

According to the previous slide, we are **192.168.100.101/24** and want to join **192.168.100.1** host. We are going to check **all the possible routes** in the routing table. Let's calculate if this **destination host** is in the **first network** linked to our host.

```
src IP (dec)       192     .     168     .     100     .     101
src IP (bin)    1100 0000 . 101010001 . 1101 0100 . 1101 0101
/24 mask (bin)  1111 1111 . 1111 1111 . 1111 1111 . 0000 0000
src net  (bin)  1100 0000 . 101010001 . 1101 0100 . 0000 0000
src net  (dec)     192     .     168     .     100     .       0

dst IP (dec)       192     .     168     .     100     .       1
dst IP (bin)    1100 0000 . 101010001 . 1101 0100 . 0000 0001
/24 mask (bin)  1111 1111 . 1111 1111 . 1111 1111 . 0000 0000
dst net  (bin)  1100 0000 . 101010001 . 1101 0100 . 0000 0000
dst net  (dec)     192     .     168     .     100     .       0
```

The destination host belongs to our network, we can speak directly with him.
If this check had failed, we would have tested the other possible routes.

Inter-network routing

When the host does not belong to my network ?

Now, we want to join **192.168.25.8** host. Let's say we have already checked the first possible route and that didn't work. We therefore test the second possibility using **192.168.122.254/24** interface. Let's calculate if this **destination host** is in the **second network** linked to our host.

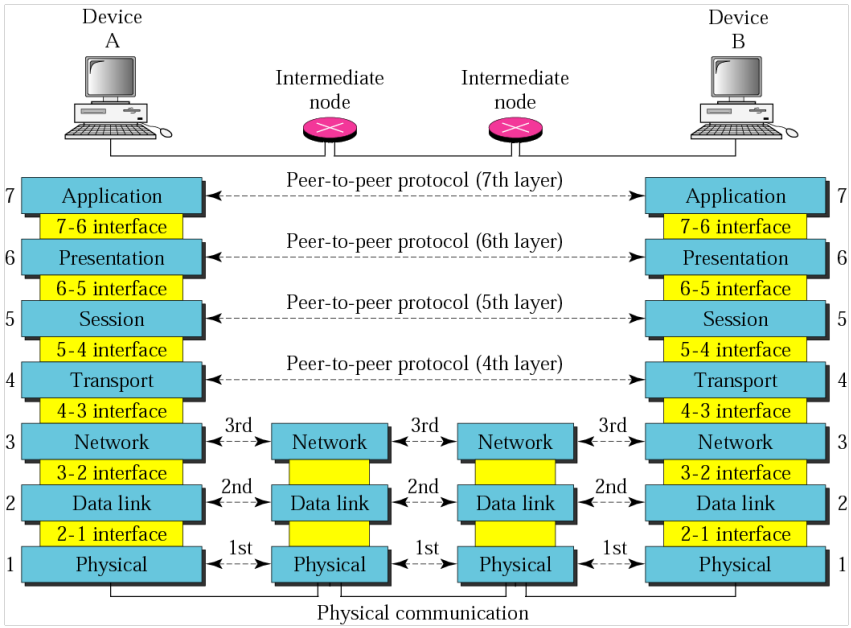
```
src IP (dec)       192     .     168     .     122     .     254
src IP (bin)   1100 0000 . 101010001 . 0111 1010 . 1111 1110
/24 mask (bin) 1111 1111 . 1111 1111 . 1111 1111 . 0000 0000
src net  (bin) 1100 0000 . 101010001 . 0111 1010 . 0000 0000
src net  (dec)     192     .     168     .     122     .       0

dst IP (dec)       192     .     168     .      25     .       8
dst IP (bin)   1100 0000 . 101010001 . 0001 1001 . 0000 1000
/24 mask (bin) 1111 1111 . 1111 1111 . 1111 1111 . 0000 0000
dst net  (bin) 1100 0000 . 101010001 . 0001 1001 . 0000 0000
dst net  (dec)     192     .     168     .      25     .       0
```

The destination host belongs to none of our network, we must entrust the message to the default gateway.

Inter-network routing

Full routing process

This diagram shows a real connection between 2 hosts :

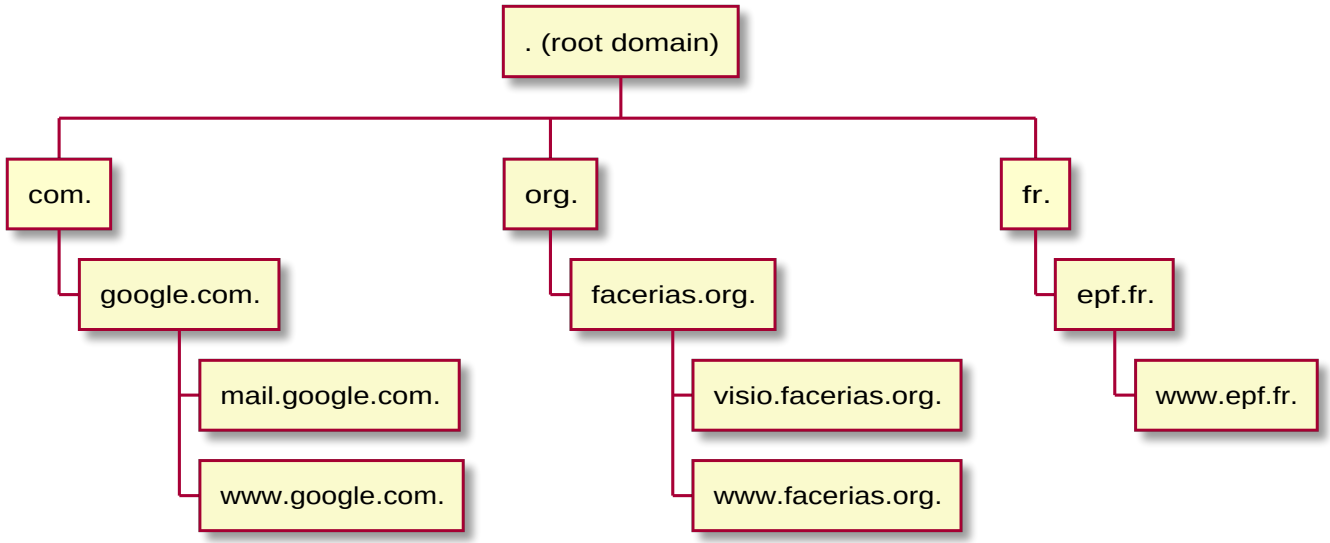


- Host A forge the message and send it on the source network ;
- First intermediate host forward the message to an other network ;
- Second intermediate host forward the message to the destination network ;
- Host B grab the message.

Each time the message goes through an host, **Time To Live** field is decreased. It is a security to avoid that a message transits in the network eternally. If the TTL falls to zero, the message is dropped. An error information message **may** be sent back to the source host.

The canonical pattern of an IPv4 address is not easy to memorize and it is worse with IPv6.
To help people to remember an host address, it is possible to use a name instead. This name is called symbolic IP address.
Using the *host* command we can find the IP address from a hostname.

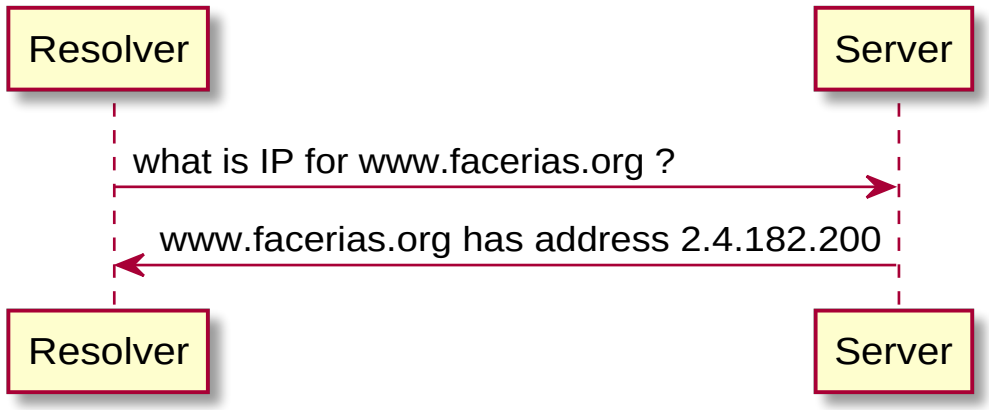
```
# host www.facerias.org
www.facerias.org has address 2.4.182.200
```



The first raw domain are called **Top Level Domains** (TLD). They take place just bellow the **root domain**.
A domain name ended with a dot is absolute. In most of cases, final dot is forgotten.
Historically, they are 2 sorts of TLD :

- Geographical (2 letters) : `fr`, `it`, `gk`, ...
- Organisational (3 letters) : com, net, org, ...

But now, you can find `info`, `cat`, ...



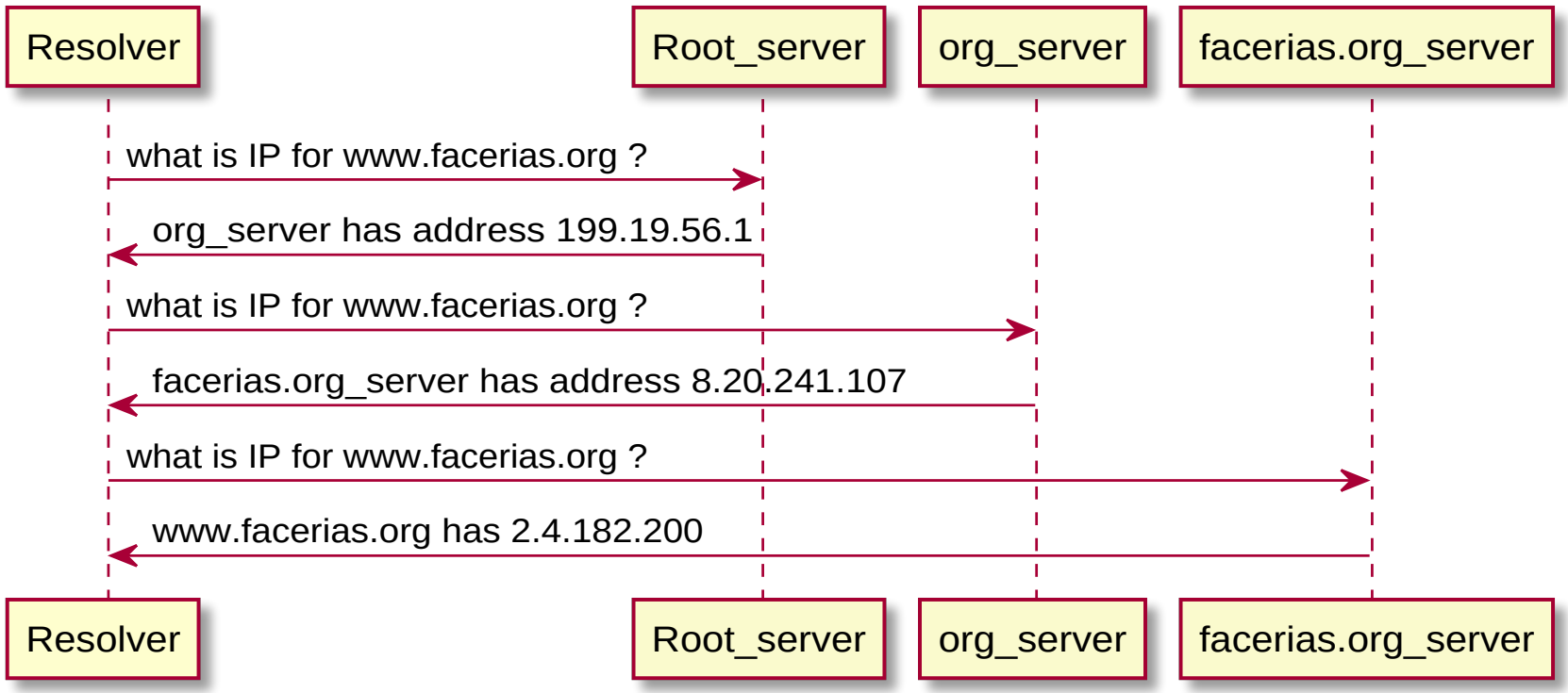
In recursive mode :

- The client (i/e the **resolver**) ask the server ;
- The **server** gives back the full answer.

This is the most standard mode, used between local hosts and local Domain Name Service servers.
DNS service stands at UDP port 53.

Domain name

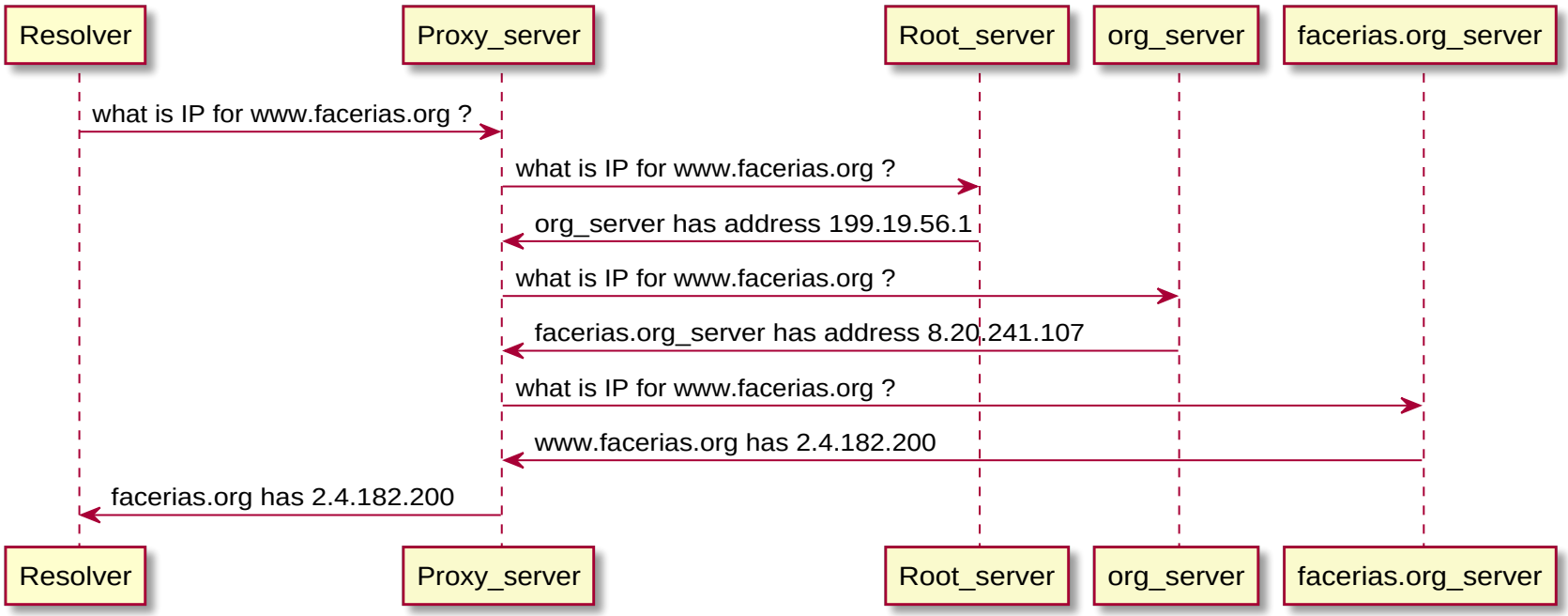
Client-Server iterative query



In iterative mode, the resolver iterates queries to complete the task.
The Domain Name Service stands at UDP port 53 too.

Domain name

Client-Server full query



Full query shows the role of the proxy DNS server :

- it acts as a server with the hosts in the local networks in recursive mode ;
- it acts as a resolver (client) with hosts on the Internet in iterative mode.

The proxy server handles cache to optimize resolving tasks.

Domain Name Service stands at UDP port 53 in any cases.

End