Kathará

kathara lab

DNS

Version	2.0
Author(s)	L. Ariemma, T. Caiazzi, G. Di Battista, M. Patrignani, M. Pizzonia, F. Ricci, M. Rimondini
E-mail	contact@kathara.org
Web	http://www.kathara.org/
Description	using the domain name system – kathara version of an existing netkit lab

1

copyright notice

- All the pages/slides in this presentation, including but not limited to, images, photos, animations, videos, sounds, music, and text (hereby referred to as "material") are protected by copyright.
- This material, with the exception of some multimedia elements licensed by other organizations, is property of the authors and/or organizations appearing in the first slide.
- This material, or its parts, can be reproduced and used for didactical purposes within universities and schools, provided that this happens for non-profit purposes.
- Information contained in this material cannot be used within network design projects or other products of any kind.
- Any other use is prohibited, unless explicitly authorized by the authors on the basis of an explicit agreement.
- The authors assume no responsibility about this material and provide this material "as is", with no implicit or explicit warranty about the correctness and completeness of its contents, which may be subject to changes.
- This copyright notice must always be redistributed together with the material, or its portions.

purpose of this lab

- get familiar with DNS
- observe the behavior of name servers and their interactions
- learn simple DNS configurations

lab limitations

- DNS security issues and protocols are not covered
  - we use a version of Bind, which currently is the most widely used domain name server software, that allows ignoring security aspects
- all IP addresses are IPv4

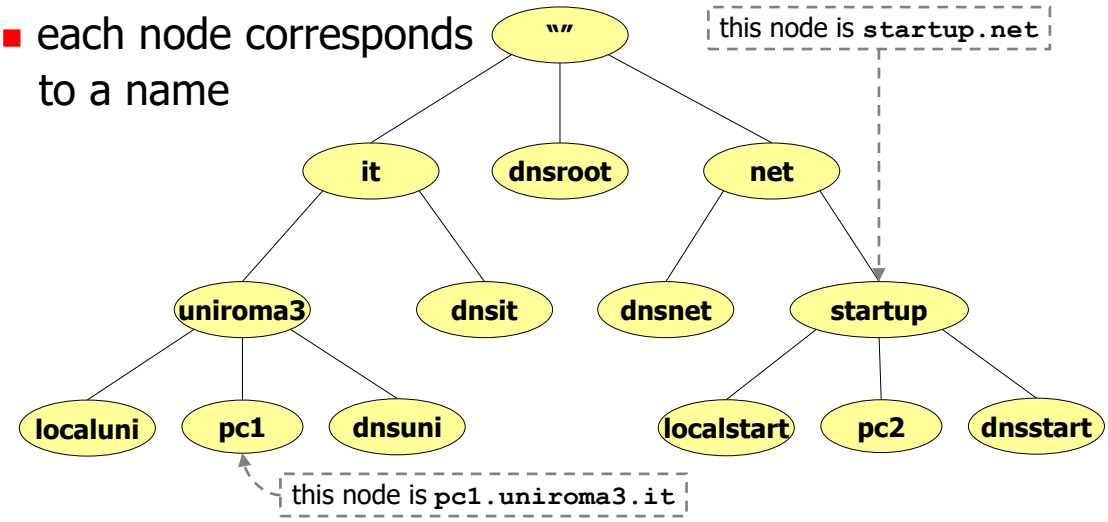
about the DNS

- takes care of associating names with IP addresses
- the name system is distributed over several nodes (hosts) that are hierarchically organized to form a tree
- each node in the hierarchy corresponds to a name
- a domain in the name system is a subtree
- a node in the hierarchy may be delegated to handle names for a particular zone
  - such a node is an authoritative server for that zone
- a zone is a domain which is devoid of those nodes having a different authoritative server (i.e., a tree without subtrees)

5

the DNS name hierarchy

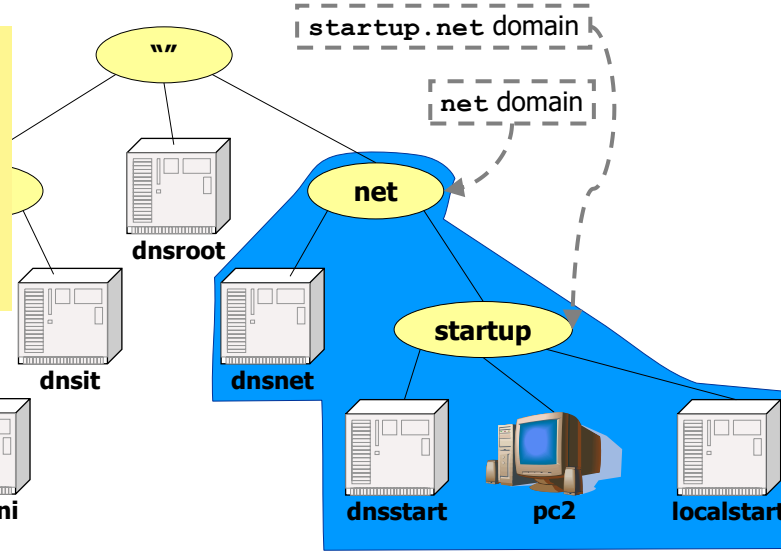
- each node corresponds to a name



6

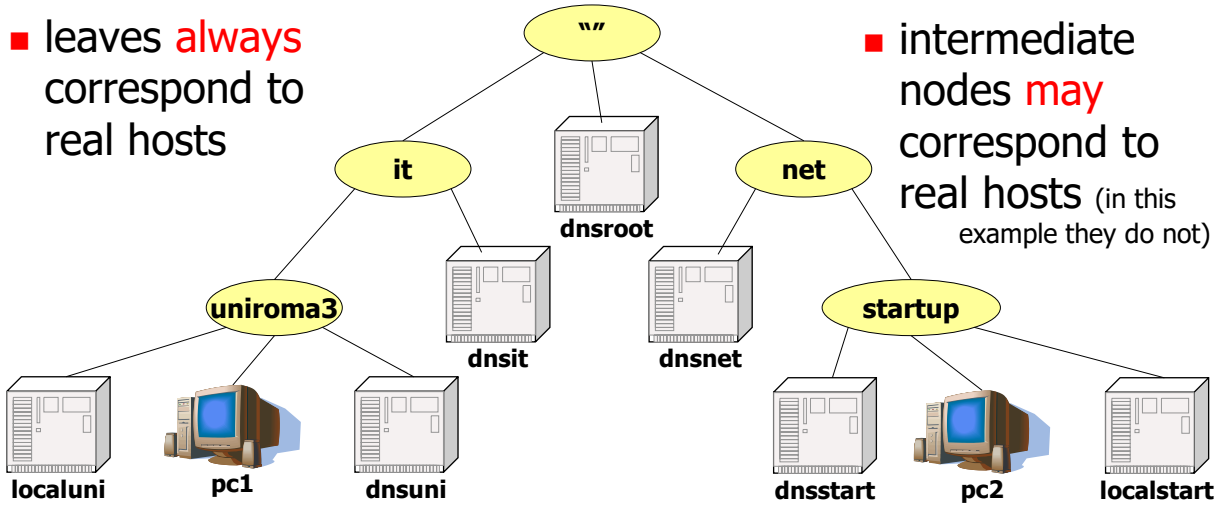
the DNS name hierarchy

- domains are subtrees
  - their name is the name of the root node
  - every node (including leaves) defines a domain
  - domains do overlap



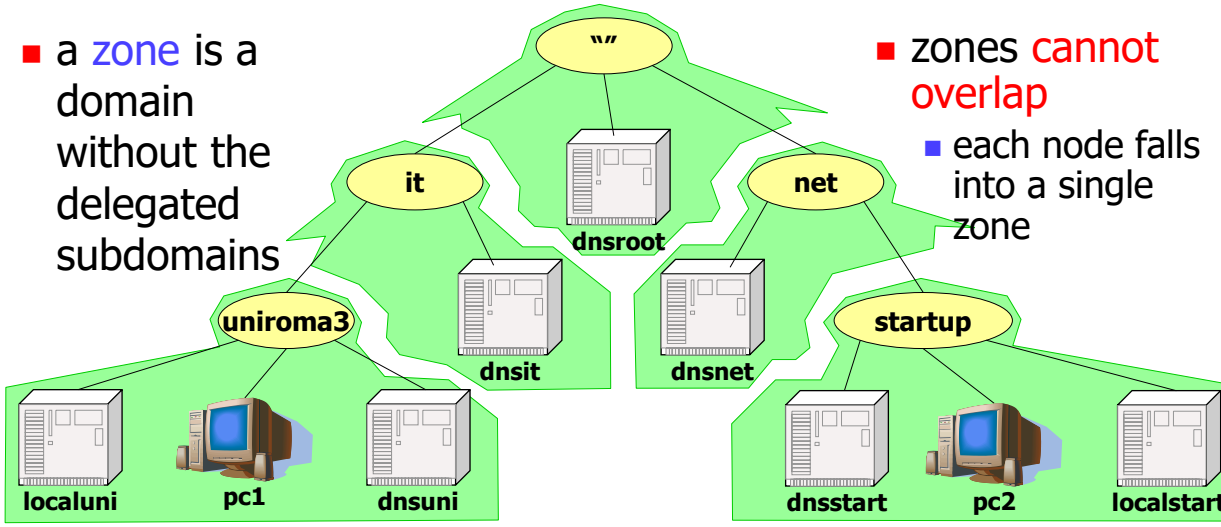
the DNS name hierarchy

- leaves always correspond to real hosts
- intermediate nodes may correspond to real hosts (in this example they do not)



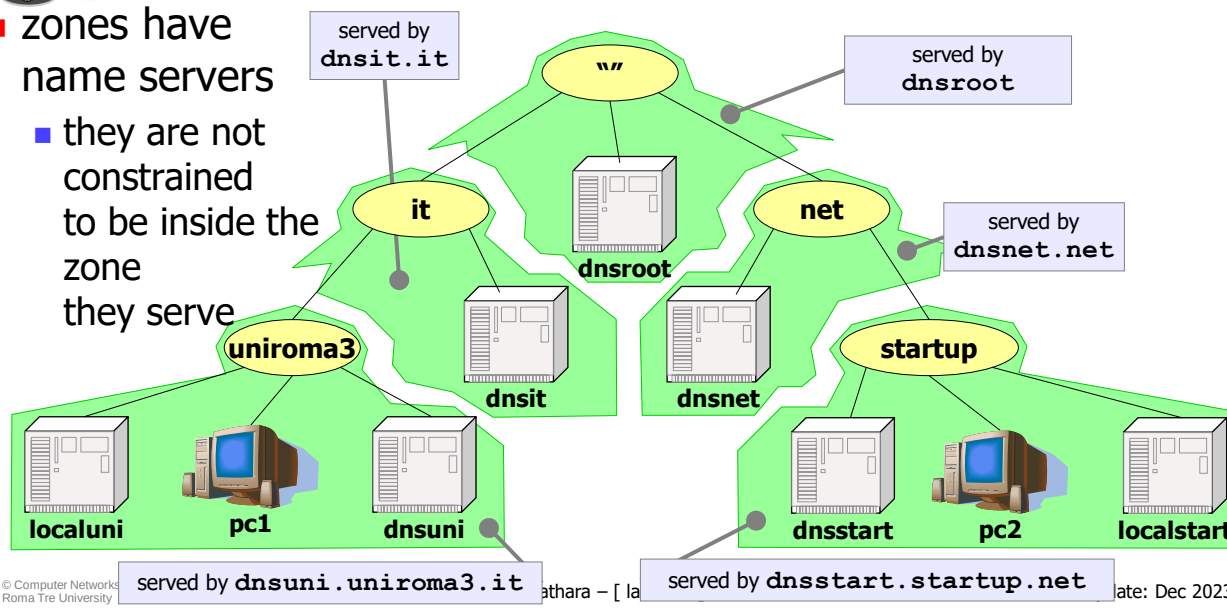
zones

- a zone is a domain without the delegated subdomains
- zones cannot overlap
  - each node falls into a single zone



zones

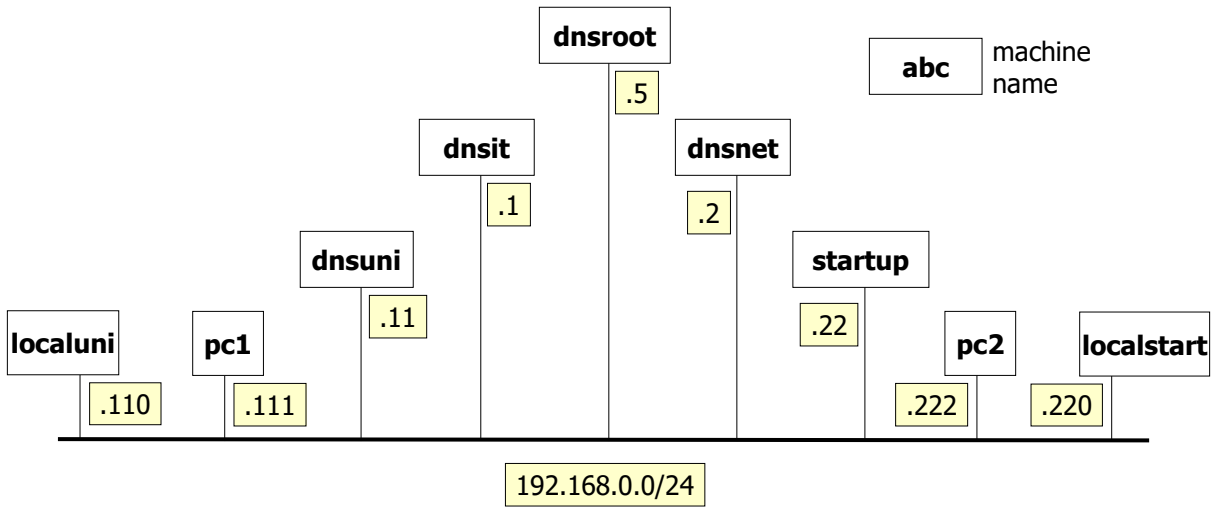
- zones have name servers
  - they are not constrained to be inside the zone they serve



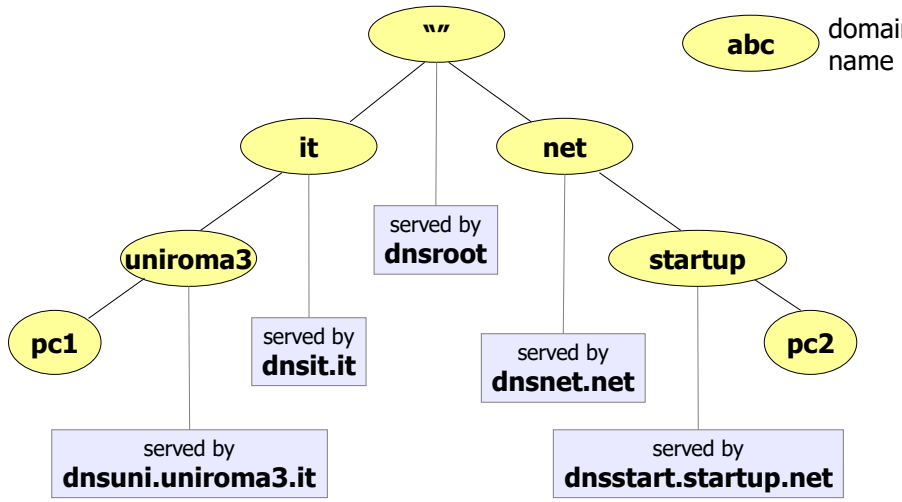
more about the DNS

- the dns hierarchy is largely orthogonal with respect to the actual network topology
- in order to focus on the behavior of the dns we choose a flat topology, consisting of a single collision domain

step 1 – network topology



step 1 – DNS (zone) hierarchy

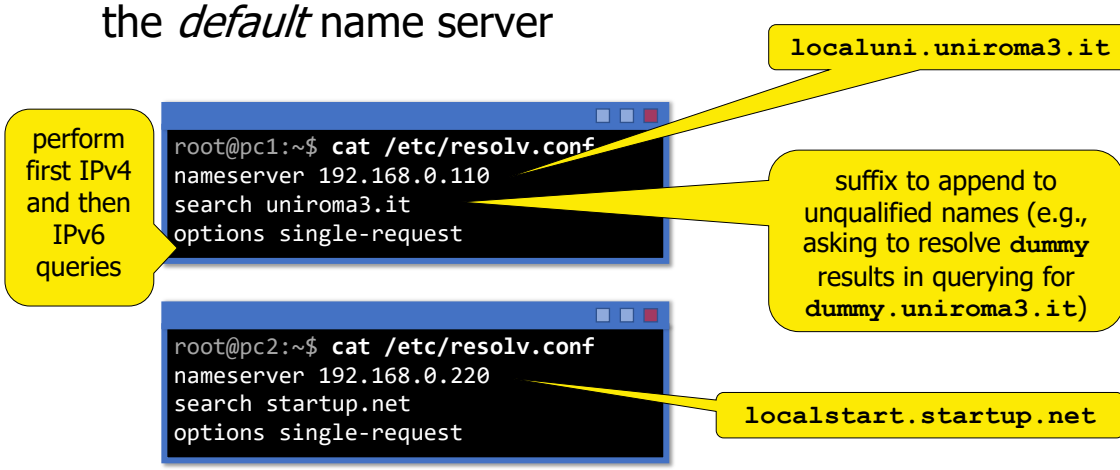


step 2 – starting the lab

- the lab is configured to
  - start all the 9 devices
  - automatically configure network interfaces (IPv4 only)
  - automatically configure the authoritative name servers
  - automatically configure name servers offering a recursive resolution service
  - automatically start the name server software (*bind*) on each name server
    - the daemon corresponding to bind is called *named*

step 2 – exploring the configuration

- configuration on the PCs consists of the specification of the *default* name server



step 2 – exploring the configuration

- configuration on the name servers specifies
  - associations between zones and name servers
  - information about the root name servers
  - authoritative information
    - associations between names and IP addresses
  - authorization to resolve recursive queries

step 2 – exploring the configuration

- configuration on the name servers specifies
  - associations between zones and name servers

```
root@dnsuni:~$ cat /etc/bind/named.conf
include "/etc/bind/named.conf.options";

zone "." {
    type hint;
    file "/etc/bind/db.root";
};

zone "uniroma3.it" {
    type master;
    file "/etc/bind/db.it.uniroma3";
};
```

include some additional configuration

where to find information about the root name server

we are the primary master for zone **`uniroma3.it`**

where to find data about the names in this zone

step 2 – exploring the configuration

- configuration on the name servers specifies
  - additional configuration

```
root@dnsuni:~$ cat /etc/bind/named.conf.options
options {
    directory "/var/cache/bind";
};
```

use this folder to store the cache. COMPULSORY, otherwise, named wont 't start

format of a resource record

```
<domain> <class> <type> <rdata>
```

- domain: the record owner (=domain to which the record refers)
- class: usually IN (=Internet system); may be HS (=hesiod) or CH (=chaos)
- type: see next slide...
- rdata: record data (depends on the record type)

step 2 – exploring the configuration

available record types

A **a host address.**
A6 Obsolete format of IPv6 address.
AAAA **an IPv6 address.**
AFSDB (x) location of AFS database servers. Experimental.
CERT holds a digital certificate.
CNAME **identifies the canonical name of an alias.**
DNAME for delegation of reverse addresses. Replaces the domain name specified with another name to be looked up. Described in RFC 2672.
GPOS Specifies the global position. Superseded by LOC.
HINFO identifies the CPU and OS used by a host.
ISDN (x) representation of ISDN addresses. Experimental.
KEY stores a public key associated with a DNS name.
KX identifies a key exchanger for this DNS name.
LOC (x) for storing GPS info. See RFC 1876. Experimental.
MX **identifies a mail exchange for the domain. See RFC 974 for details.**
NAPTR name authority pointer.
NSAP a network service access point.
NS **the authoritative nameserver for the domain.**
NXT used in DNSSEC to securely indicate that RRs with an owner name in a certain name interval do not exist in a zone and indicate what R
PTR a pointer to another part of the domain name space.
PX provides mappings between RFC 822 and X.400 addresses.
RP (x) information on persons responsible for the domain. Experimental.
RT (x) route-through binding for hosts that do not have their own direct wide area network addresses. Experimental.
SIG ("signature") contains data authenticated in the secure DNS. See RFC 2535 for details.
SOA **identifies the start of a zone of authority.**
SRV information about well known network services (replaces WKS).
TXT text records.
WKS (h) information about which well known network services, such as SMTP, that a domain supports. Historical, replaced by newer RR SRV.
X25 (x) representation of X.25 network addresses. Experimental

step 2 – exploring the configuration

- configuration on the name servers specifies
  - information about the root name servers

```
root@dnsuni:~$ cat /etc/bind/db.root
.                       IN  NS    ROOT-SERVER.
ROOT-SERVER.            IN  A     192.168.0.5
```

a **resource record**

step 2 – exploring the configuration

- configuration on the name servers specifies
  - authoritative information

```
root@dnsuni:~$ cat /etc/bind/db.it.uniroma3
$TTL    60000
```

time to live, in seconds
(determines how long a resource record should be cached)

step 2 – exploring the configuration

- configuration on the name servers specifies
  - authoritative information

```
root@dnsuni:~$ cat /etc/bind/db.it.uniroma3
$TTL    60000
@               IN      SOA     dnsuni.uniroma3.it.
root.dnsuni.uniroma3.it. (
                        2006031201 ; serial
                        28 ; refresh
                        14 ; retry
                        3600000 ; expire
                        0 ; negative cache ttl
                        )
```

- must be all on a single line; line breaks can only be introduced when using parentheses
- a zone data file can contain only one SOA record

Start of Authority record

step 2 – exploring the configuration

- configuration on the name servers specifies
  - authoritative information

```
root@dnsuni:~$ cat /etc/bind/db.it.uniroma3
$TTL    60000
@               IN
oot.dnsuni.uniroma3.it.
```

this record is referred to the current origin (`uniroma3.it`)

- all domain names in this data file that are not fully qualified (do not end with a '.') are relative to the *origin*
- the *origin* is the domain name in the *zone* statement of the server configuration file:

```
zone "uniroma3.it" {
        type master;
        file "/etc/bind/db.it.uniroma3";
};
```

step 2 – exploring the configuration

- configuration on the name servers specifies
  - authoritative information

```
root@dnsuni:~$ cat /etc/bind/db.it.uniroma3
$TTL    60000
@               IN      SOA     dnsuni.uniroma3.it.
root.dnsuni.uniroma3.it. (
                        2006031  serial
                        28 ;
```

primary master (=authority) server for this zone (`dnsuni.uniroma3.it`);
don't forget the trailing dot, or the origin name (`uniroma3.it`) would be appended!

step 2 – exploring the configuration

- configuration on the name servers specifies
  - authoritative information

```
root@dnsuni:~$ cat /etc/bind/db.it.uniroma3
$TTL    60000
@               IN      SOA     dnsuni.uniroma3.it.
root.dnsuni.uniroma3.it. (
                        2006031201 ; serial
                                   bire
                                   cache ttl
```

mail address of the person that is responsible for the zone (`root@dnsuni.uniroma3.it`)

- the first '.' must be replaced by a '@'
- only meant to be used by humans; has no use within the dns service

step 2 – exploring the configuration

- configuration on the name servers specifies
  - authoritative information

```
root@dnsuni:~$ cat /etc/bind/db.it.uniroma3
$TTL    60000
@               IN      SOA     dnsuni.uniroma3.it.
root.dnsuni.uniroma3.it. (
                        2006031201 ; serial
                        28 ; refresh
                        14 ; retry
                        3600000 ; expire
                        0 ; negative cache ttl
                        )
```

makes sense for master/slave server configurations

step 2 – exploring the configuration

- configuration on the name servers specifies
  - authoritative information

```
root@dnsuni:~$ cat /etc/bind/db.it.uniroma3
$TTL    60000
@               IN      SOA     dnsuni.uniroma3.it.
root.dnsuni.uniroma3.it. (
                        2006031201 ; serial
                        28 ; refresh
```

serial number

- determines how recent the information is
- influences all data within the zone
- conventional format:
  YYYYMMDDNN (year, month, day, # of changes within that day)

step 2 – exploring the configuration

- configuration on the name servers specifies
  - authoritative information

```
root@dnsuni:~$ cat /etc/bind/db.it.uniroma3
$TTL    60000
@               IN      SOA     dnsuni.uniroma3.it.
root.dnsuni.uniroma3.it. (
                        2006031201 ; serial
                        28 ; refresh
                        14 ; retry
                        3600000 ; expire
```

refresh interval (seconds)

tells a slave how often to check that the data for this zone is up to date

step 2 – exploring the configuration

- configuration on the name servers specifies
  - authoritative information

```
root@dnsuni:~$ cat /etc/bind/db.it.uniroma3
$TTL    60000
@               IN      SOA     dnsuni.uniroma3.it.
root.dnsuni.uniroma3.it. (
                        2006031201 ; serial
                        28 ; refresh
                        14 ; retry
                        3600000 ; expire
                        0 ; negative cache ttl
                        )
```

interval (seconds) between subsequent attempts to contact the master

step 2 – exploring the configuration

- configuration on the name servers specifies
  - authoritative information

```
root@dnsuni:~$ cat /etc/bind/db.it.uniroma3
$TTL    60000
@               I
root.dnsuni.unir
                        28 ; refresh
                        14 ; retry
                        3600000 ; expire
                        0 ; negative cache ttl
                        )
```

if the slave fails to contact the master for this amount of time, it considers the zone data too old and stops giving answers about it

slave expire time (seconds)

step 2 – exploring the configuration

- configuration on the name servers specifies
  - authoritative information

```
root@dnsuni:~$ cat /etc/bind/db.it.uniroma3
$TTL    60000
@               IN      SOA     dnsuni.uniroma3.it.
root.dnsuni.uniroma3.it. (
                        2006031201 ; serial
                        28 ; refresh
                        14 ; retry
                        3600000 ; expire
                        0 ; negative cache ttl
                        )
```

ttl for negative responses from authoritative name servers

step 2 – exploring the configuration

- configuration on the name servers specifies
  - associations between names and ip addresses

```
root@dnsuni:~$ cat /etc/bind/db.it.uniroma3
$TTL    60000
@               IN      SOA     dnsuni.uniroma3.it.
r          .it. (
                        2006031201 ;
                        28 ; refresh
                        14 ; retry
                        3600000 ; exp
                        0 ; negative cache ttl
                        )
@                   IN      NS      dnsuni.uniroma3.it.
dnsuni.uniroma3.it. IN      A       192.168.0.11
pc1.uniroma3.it.    IN      A       192.168.0.111
```

record type NS (name server)

the authoritative name server for this zone (**uniroma3.it**) is **dnsuni.uniroma3.it**

33

step 2 – exploring the configuration

- configuration on the name servers specifies
  - associations between names and ip addresses

```
root@dnsuni:~$ cat /etc/bind/db.it.uniroma3
$TTL    60000
@               IN      SOA     dnsuni.uniroma3.it.
root.dnsuni.uniroma3.it. (
                        2006031201 ; serial
                        28 ; refresh
                        14 ; retry
                        3600000 ; exp
                        0 ; negative
                        )
@                   IN      NS      dnsuni.uniroma3.it.
dnsuni.uniroma3.it. IN      A       192.168.0.11
pc1.uniroma3.it.    IN      A       192.168.0.111
```

record type A (address)

two machines in this zone: **dnsuni.uniroma3.it** **pc1.uniroma3.it**

34

step 2 – exploring the configuration

- configuration on the name servers specifies
  - associations between names and ip addresses

```
root@dnsit:~$ tail -n 5 /etc/bind/db.it
@                       IN      NS      dnsit.it.
dnsit.it.               IN      A       192.168.0.1

uniroma3.it.            IN      NS      dnsuni.uniroma3.it.
dnsuni.uniroma3.it.     IN      A       192.168.0.11
```

dnsit.it is the authority for this zone (**it**)

dnsuni.uniroma3.it is the authority for zone **uniroma3(.it)**

step 2 – exploring the configuration

- configuration on the name servers specifies
  - allowing recursive queries and disabling dnssec

```
root@localuni:~$ cat /etc/bind/named.conf.options
options {
    directory "/var/cache/bind";
    allow-recursion { 192.168.0.0/24; };
    auto-dnssec off;
    dnssec-validation no;
    dnssec-enable no;
    dnssec-lookaside no;
    filter-aaaa-on-v4 yes;
    send-cookie no;
};
```

disable dnssec

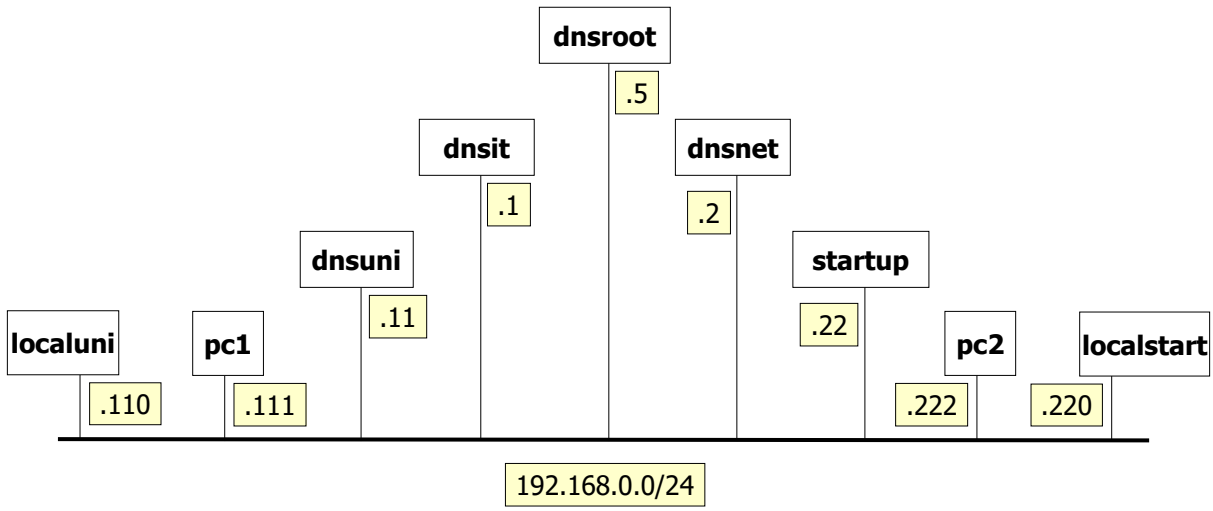
filter AAAA addresses on IPv4 only

allow recursive queries from **192.168.0.0/24**

Do not send DNS cookies

let's start the lab

step 3 – experiment setting



sniff the traffic

- connect the wireshark device to collision domain A

```
user@localhost:~/kathara-lab_dns$ kathara lconfig -n wireshark --add A
```

- open any browser on the host machine
  - on **localhost:3000**
  - sniff eth1

step 3 – ping from pc1

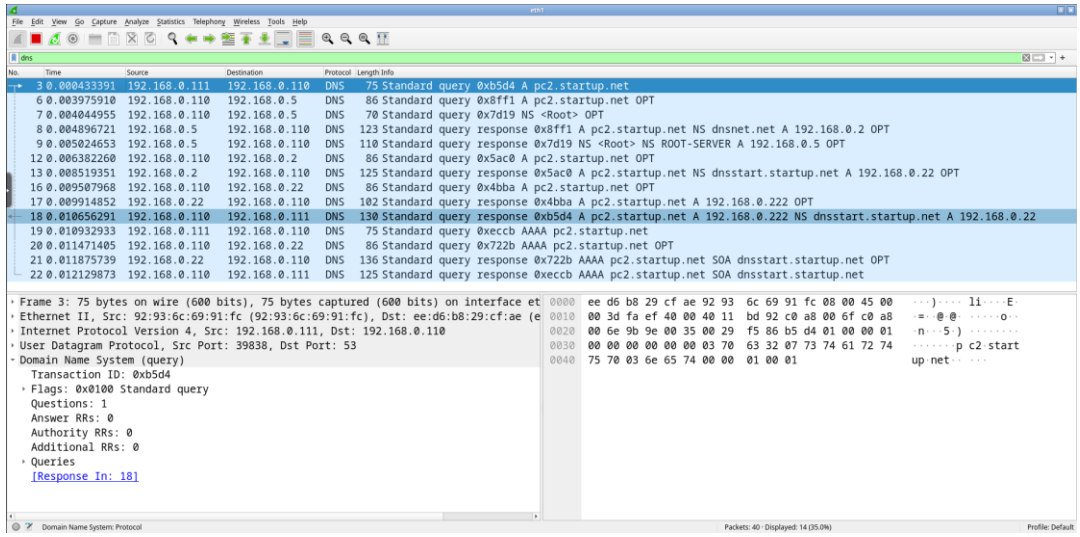
- execute a ping command towards pc2

Numeric output only. No attempt will be made to lookup symbolic names for host addresses.

pc1

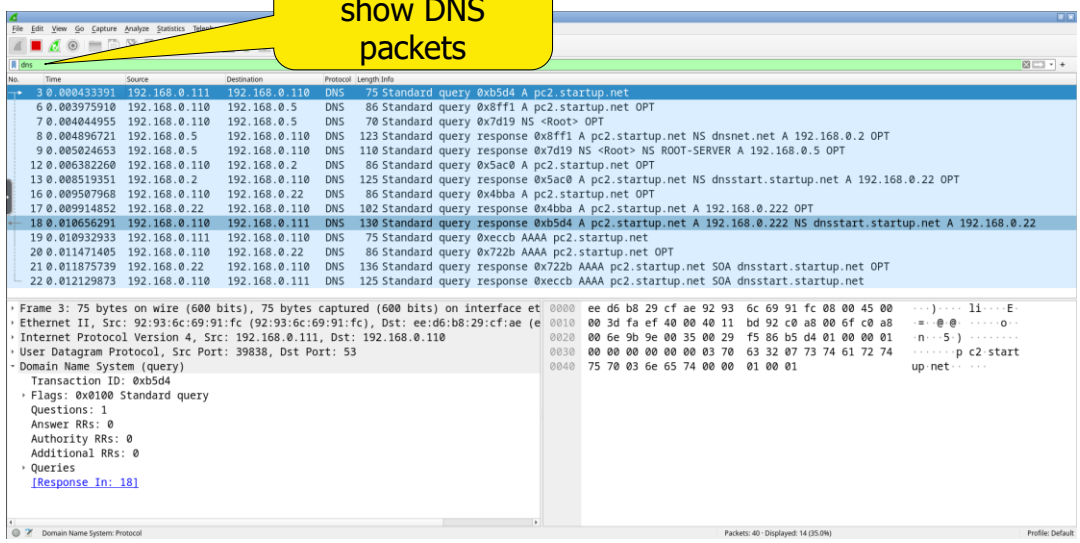
```
root@pc1:/# ping -n pc2.startup.net
PING pc2.startup.net (192.168.0.222) 56(84) bytes of data.
64 bytes from 192.168.0.222: icmp_seq=1 ttl=64 time=1.50 ms
64 bytes from 192.168.0.222: icmp_seq=2 ttl=64 time=0.580 ms
64 bytes from 192.168.0.222: icmp_seq=3 ttl=64 time=0.525 ms
--- pc2.startup.net ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2010ms
rtt min/avg/max/mdev = 0.525/0.867/1.496/0.445 ms
```

step 3 – the sniffer output



ste[illegible]niffer output

filter to only show DNS packets



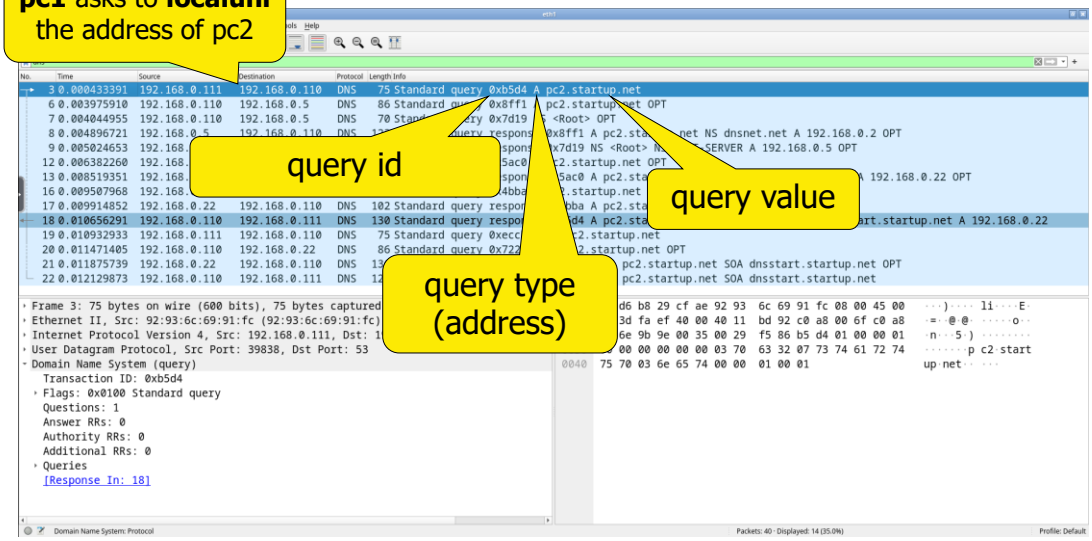
step 3 – the sniffer output

pc1 asks to **localuni** the address of pc2

query id

query value

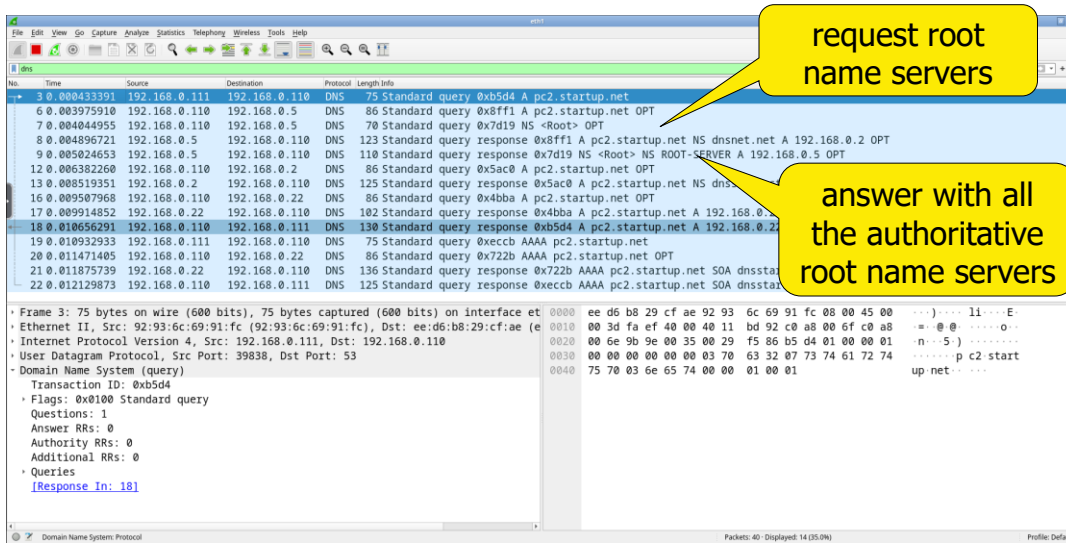
query type (address)



step 3 – the sniffer output

request root name servers

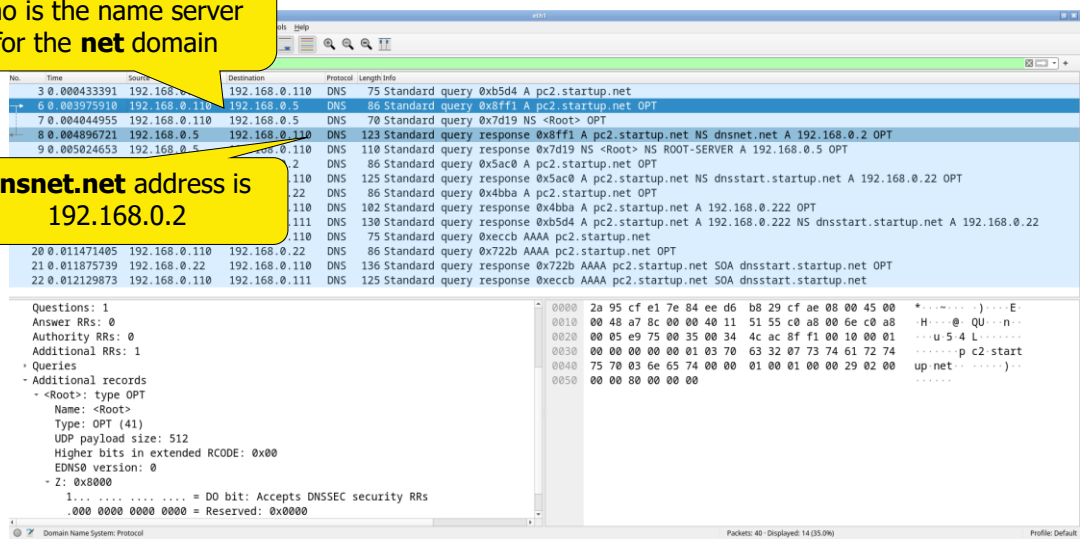
answer with all the authoritative root name servers



step 3 – the sniffer output

localuni asks **dnsroot** who is the name server for the **net** domain

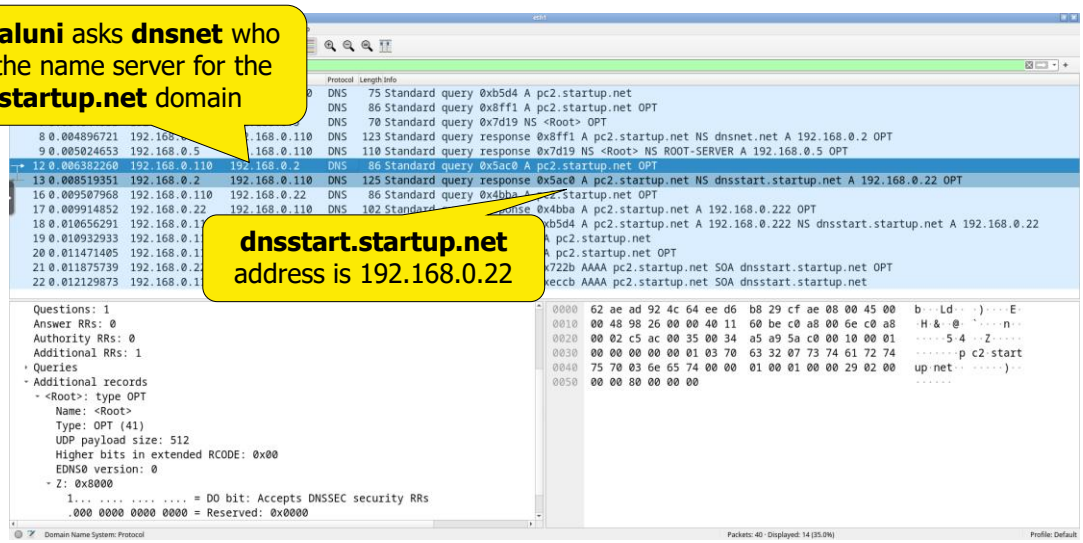
dnsnet.net address is 192.168.0.2



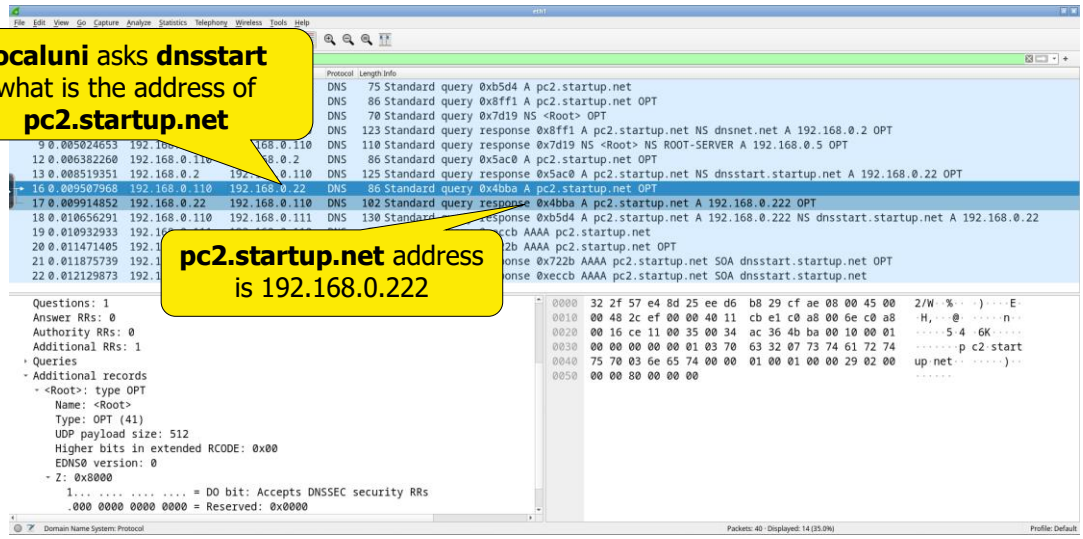
step 3 – the sniffer output

localuni asks **dnsnet** who is the name server for the **startup.net** domain

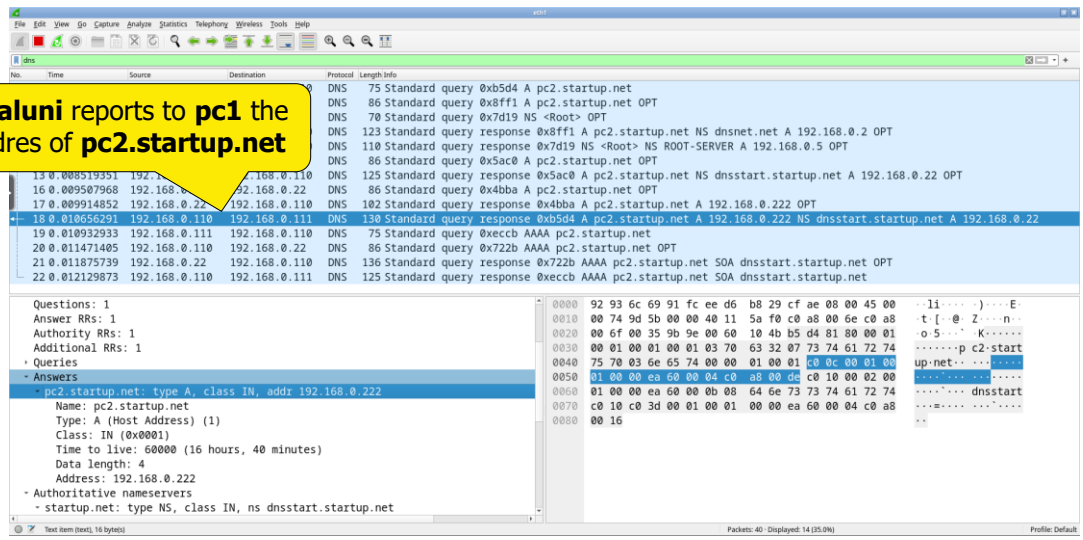
dnsstart.startup.net address is 192.168.0.22



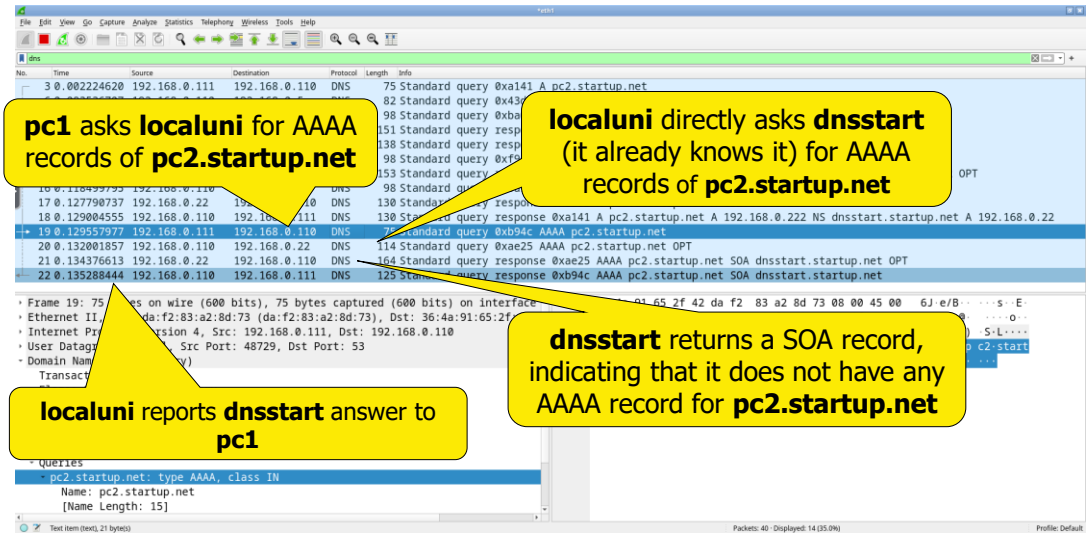
step 3 – the sniffer output



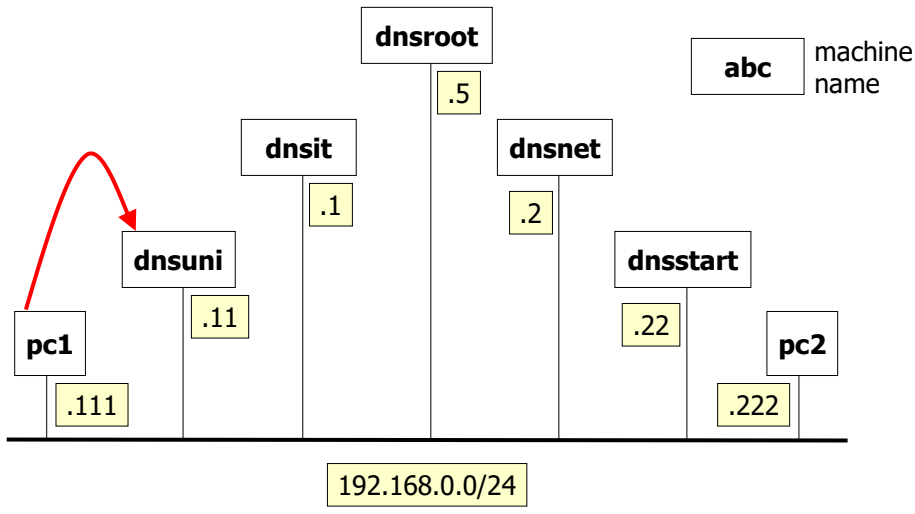
step 3 – the sniffer output



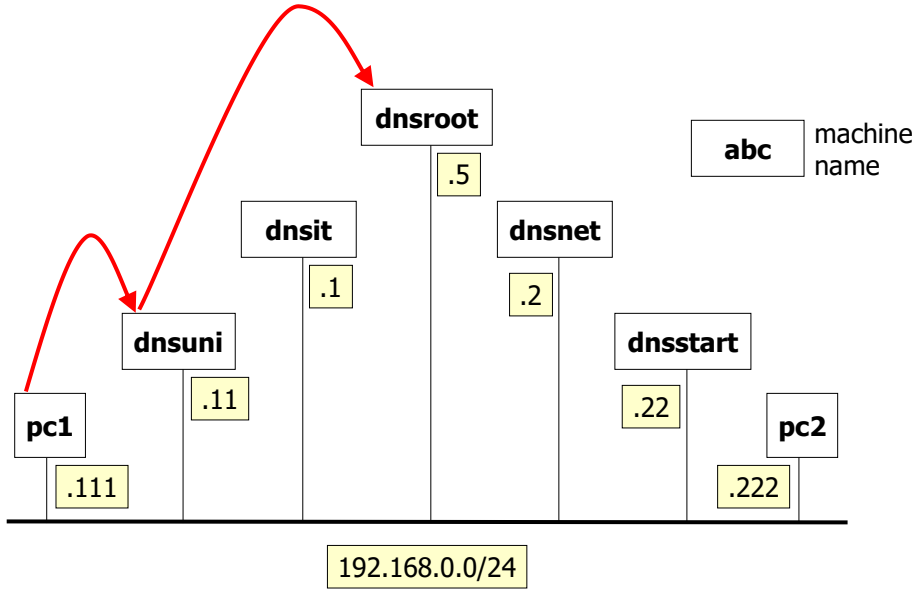
step 3 – the sniffer output



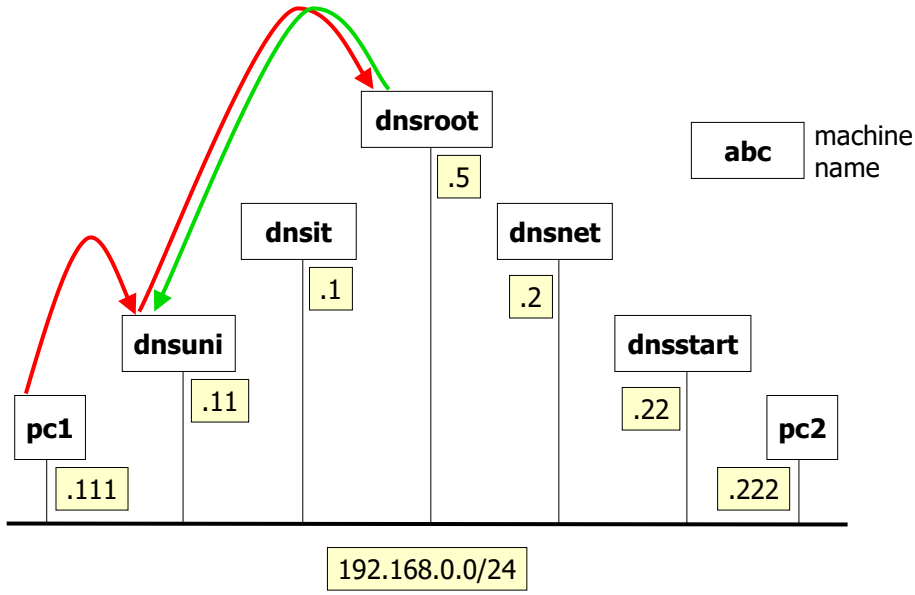
step 3 – exchanged messages



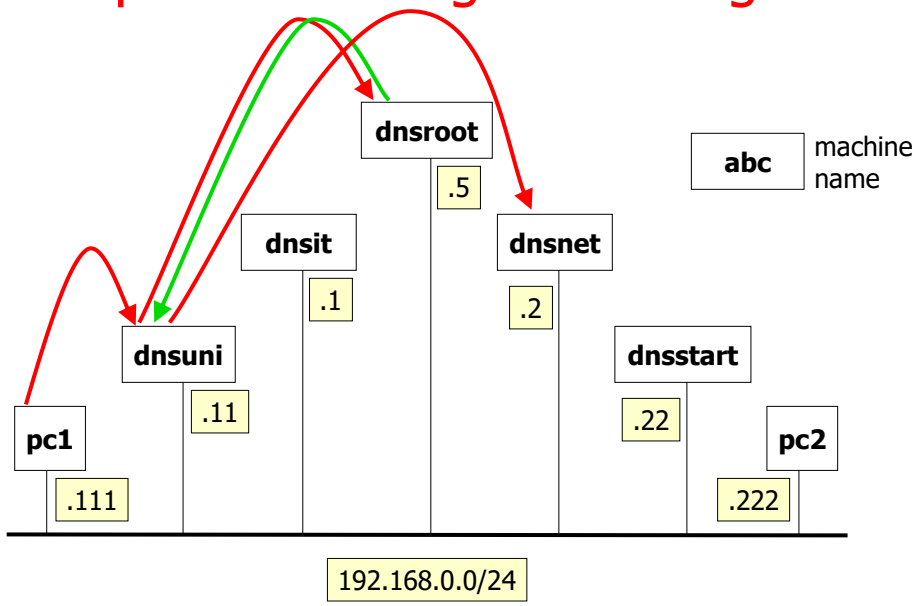
step 3 – exchanged messages



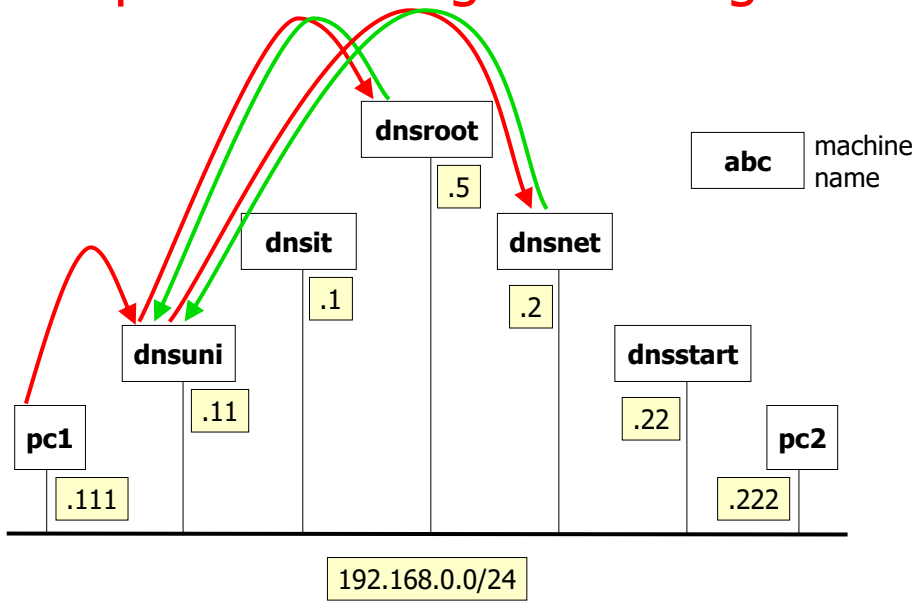
step 3 – exchanged messages



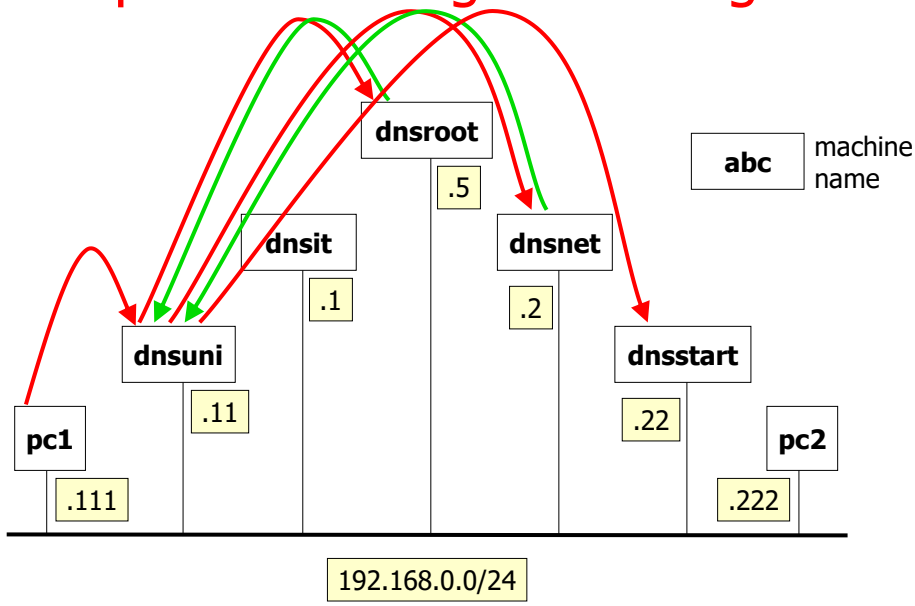
step 3 – exchanged messages



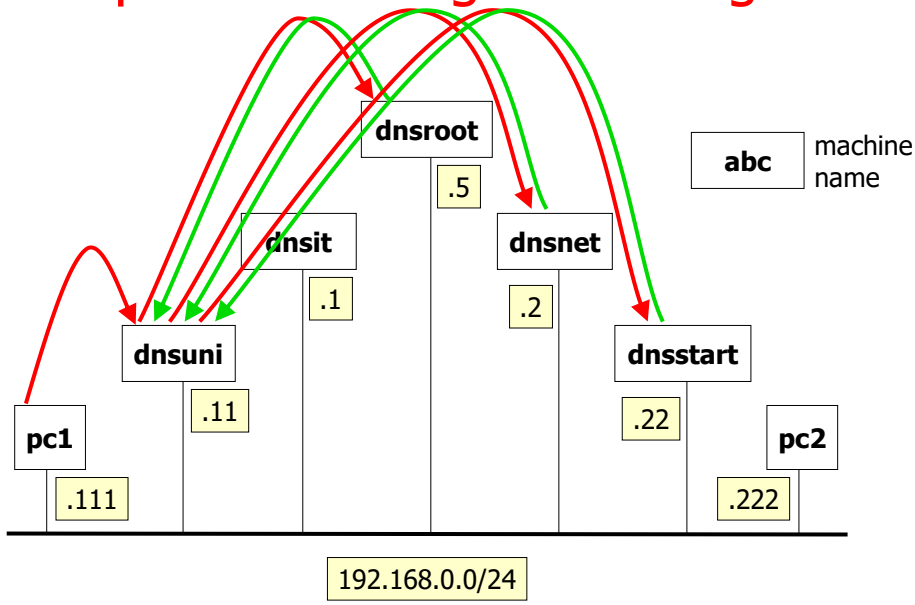
step 3 – exchanged messages



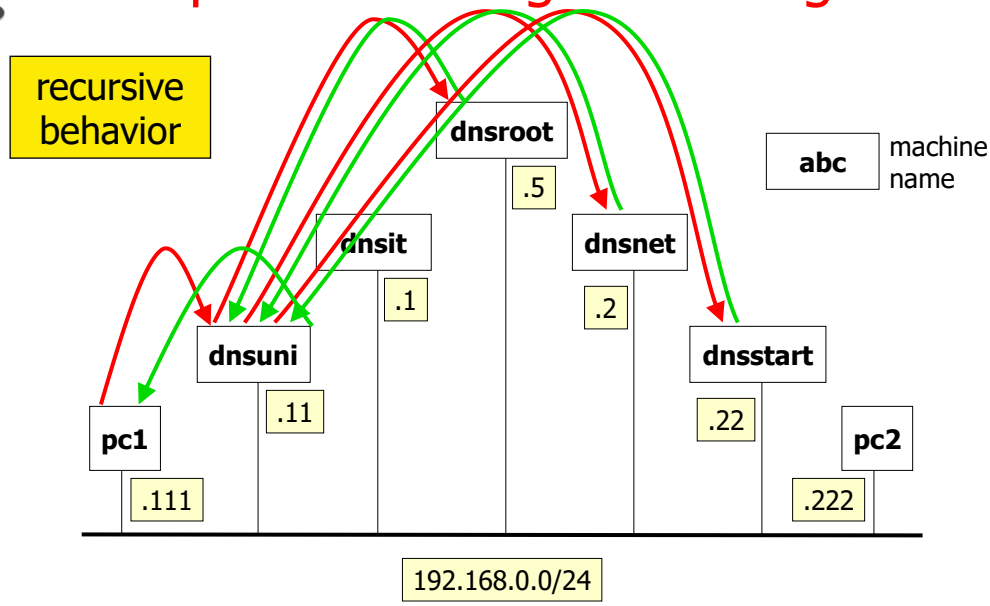
step 3 – exchanged messages



step 3 – exchanged messages



step 3 – exchanged messages

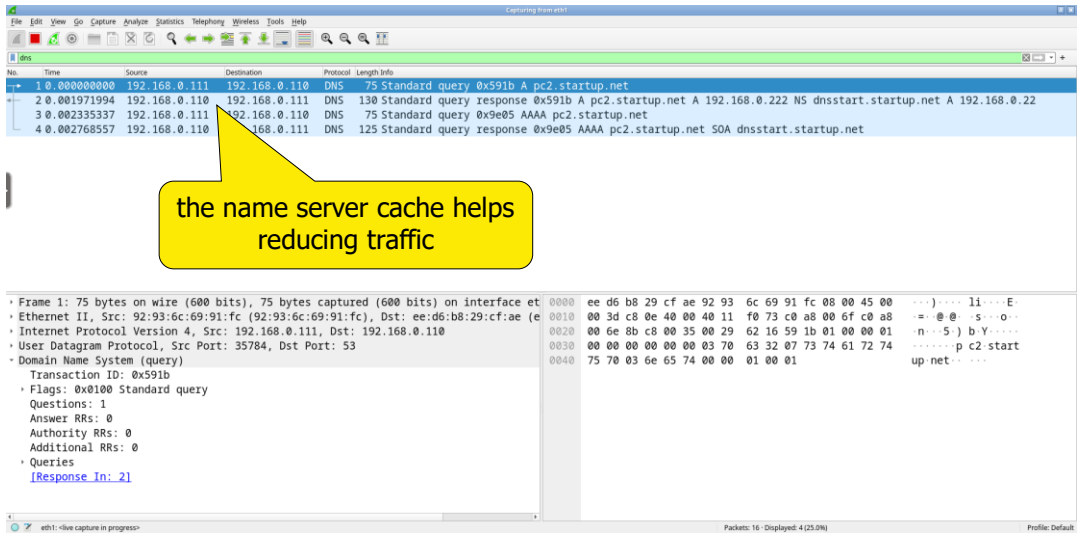


step 4 – repeating the experiment

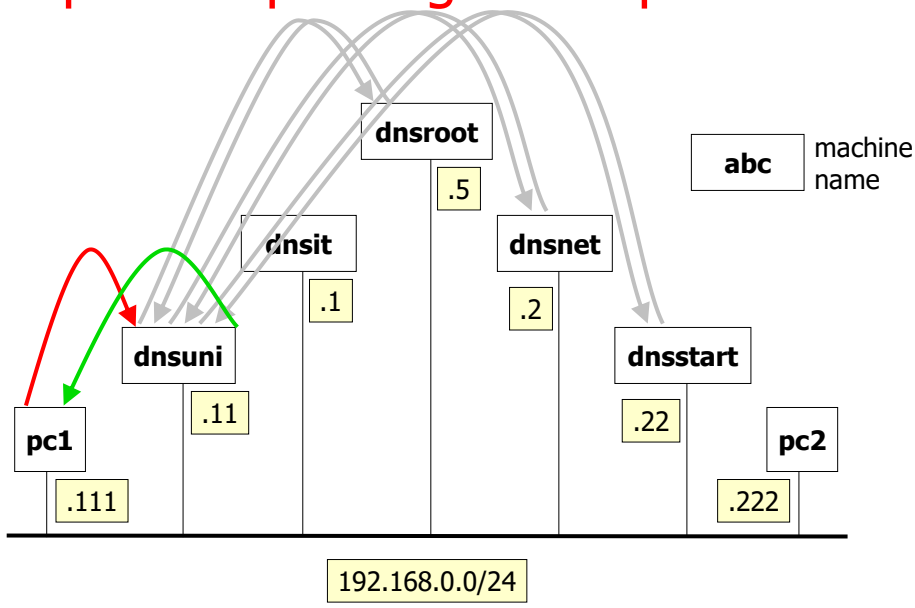
- execute a ping command towards pc2

```
pc1
root@pc1:/# ping -n pc2.startup.net
PING pc2.startup.net (192.168.0.222) 56(84) bytes of data.
64 bytes from 192.168.0.222: icmp_seq=1 ttl=64 time=1.50 ms
64 bytes from 192.168.0.222: icmp_seq=2 ttl=64 time=0.580 ms
64 bytes from 192.168.0.222: icmp_seq=3 ttl=64 time=0.525 ms
--- pc2.startup.net ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2010ms
rtt min/avg/max/mdev = 0.525/0.867/1.496/0.445 ms
```

step 4 – repeating the experiment



step 4 – repeating the experiment



step 5 – cleaning the cache

localuni

```
root@localuni:/# rndc flush
```

- rndc controls the operation of a name server
- the flush command cleans up caches
  - a new client query triggers the complete sequence of iterative queries

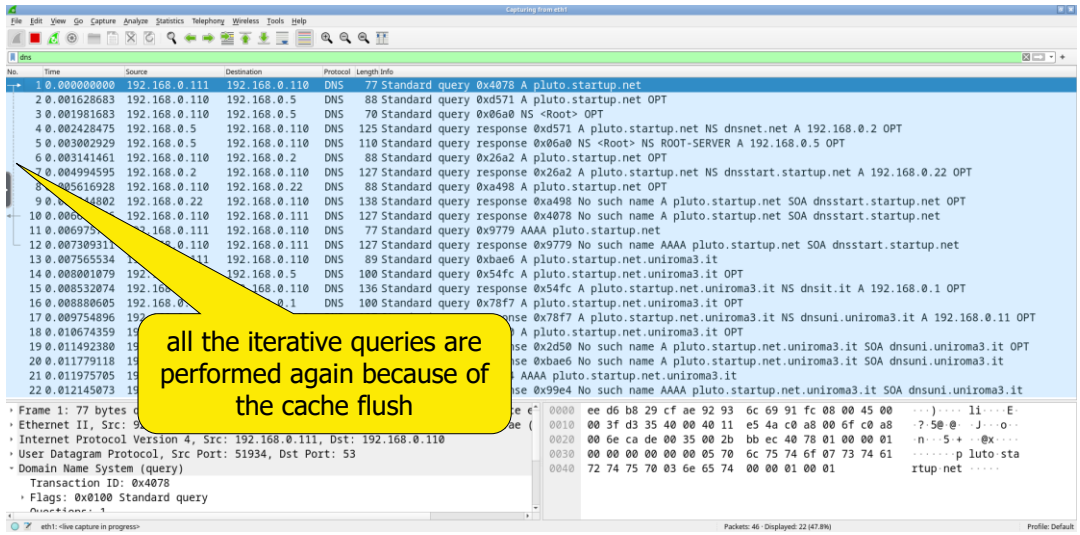
step 6 – ping non-existent target

- execute a ping command towards a non-existent target

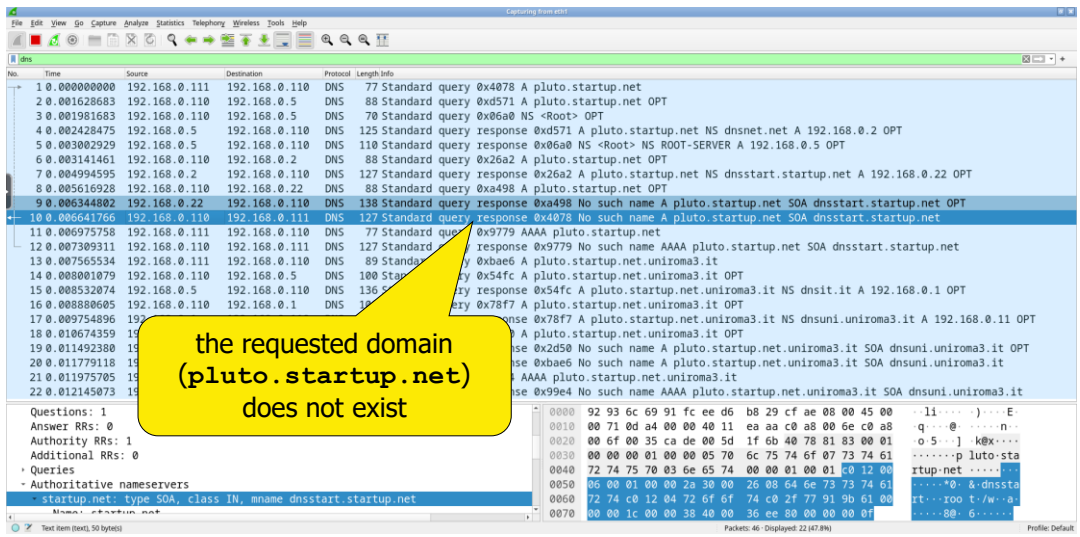
pc1

```
root@pc1:/# ping pluto.startup.net
ping: pluto.startup.net: Name or service not known
```

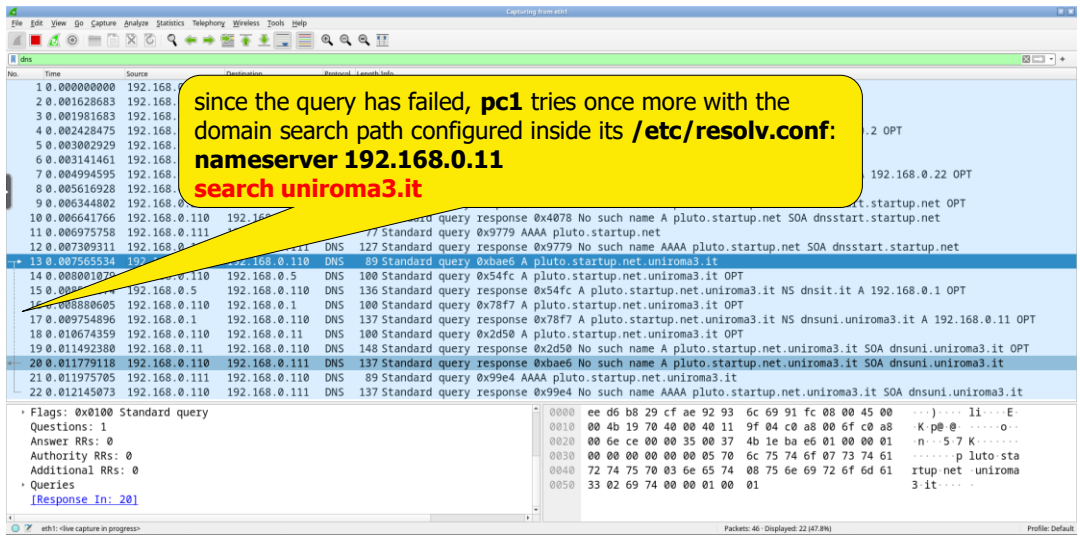
step 6 – non-existent target



step 6 – non-existent target



step 6 – non-existent target

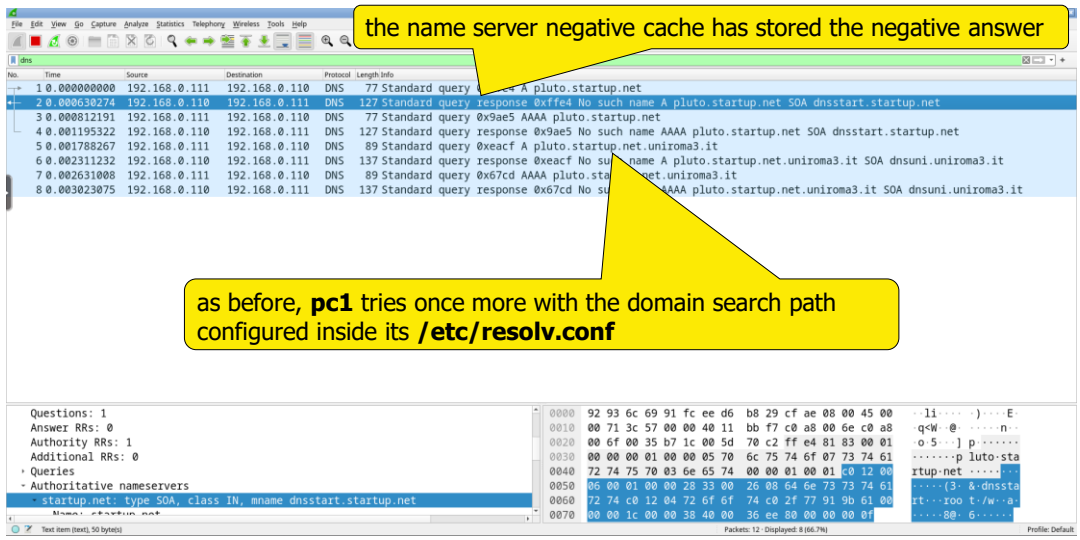


step 6 – ping non-existent target

- repeat the ping command towards the non-existent target

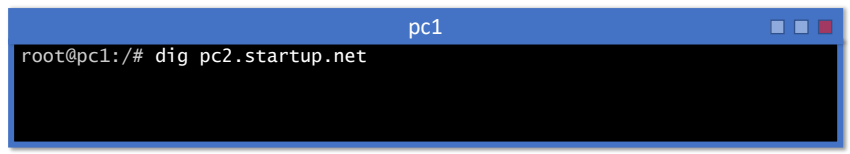
```
pc1
root@pc1:/# ping pluto.startup.net
ping: pluto.startup.net: Name or service not known
```

step 6 – ping non-existent target



step 7 – advanced queries

- resource records can be searched by using **`dig`**
  - highly customizable queries
  - detailed responses



step 7 – advanced queries

answer flags:
qr: query response
rd: recursion desired (the user asked for a recursive lookup)
ra: recursion available (the server allows recursive lookups)

pc1

```
root@pc1:/# dig pc2.startup.net
; <<>> DiG 9.18.19-1~deb12u1-Debian <<>> pc2.startup.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22137
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: b4c95f2012cf8905d705a97c657b10f314be1a5abe3ec74d (good)
;; QUESTION SECTION:
;pc2.startup.net.               IN      A

;; ANSWER SECTION:
pc2.startup.net.        60000   IN      A       192.168.0.222

;; AUTHORITY SECTION:
startup.net.            59173   IN      NS      dnsstart.startup.net.

;; ADDITIONAL SECTION:
dnsstart.startup.net.   59173   IN      A       192.168.0.22

;; Query time: 0 msec
;; SERVER: 192.168.0.110#53(192.168.0.110) (UDP)
;; WHEN: Thu Dec 14 14:28:03 UTC 2023
;; MSG SIZE  rcvd: 127
```


step 7 – advanced queries

these sections correspond to those contained in DNS packets

pc1

```
root@pc1:/# dig pc2.startup.net
; <<>> DiG 9.18.19-1~deb12u1-Debian <<>> pc2.startup.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22137
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: b4c95f2012cf8905d705a97c657b10f314be1a5abe3ec74d (good)
;; QUESTION SECTION:
;pc2.startup.net.               IN      A

;; ANSWER SECTION:
pc2.startup.net.        60000   IN      A       192.168.0.222

;; AUTHORITY SECTION:
startup.net.            59173   IN      NS      dnsstart.startup.net.

;; ADDITIONAL SECTION:
dnsstart.startup.net.   59173   IN      A       192.168.0.22

;; Query time: 0 msec
;; SERVER: 192.168.0.110#53(192.168.0.110) (UDP)
;; WHEN: Thu Dec 14 14:28:03 UTC 2023
;; MSG SIZE  rcvd: 127
```

step 7 – advanced queries

pc1

```
root@pc1:/# dig pc2.startup.net
; <<>> DiG 9.18.19-1~deb12u1-Debian <<>> pc2.startup.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22137
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: b4c95f2012cf8905d705a97c657b10f314be1a5abe3ec74d (good)
;; QUESTION SECTION:
;pc2.startup.net.               IN      A

;; ANSWER SECTION:
pc2.startup.net.        60000   IN      A       192.168.0.222

;; AUTHORITY SECTION:
startup.net.            59173   IN      NS      dnsstart.startup.net.

;; ADDITIONAL SECTION:
dnsstart.startup.net.   59173   IN      A       192.168.0.22

;; Query time: 0 msec
;; SERVER: 192.168.0.110#53(192.168.0.110) (UDP)
;; WHEN: Thu Dec 14 14:28:03 UTC 2023
;; MSG SIZE  rcvd: 127
```

records being searched (class: **`IN`**, type: **`A`** ⇒ address records)

a dns message never contains more than one question section

step 7 – advanced queries

pc1

```
root@pc1:/# dig pc2.startup.net
; <<>> DiG 9.18.19-1~deb12u1-Debian <<>> pc2.startup.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22137
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: b4c95f2012cf8905d705a97c657b10f314be1a5abe3ec74d (good)
;; QUESTION SECTION:
;pc2.startup.net.               IN      A

;; ANSWER SECTION:
pc2.startup.net.        60000   IN      A       192.168.0.222

                                        NS      dnsstart.startup.net.

                                        A       192.168.0.22

                                0.110) (UDP)
;; WHEN: Thu Dec 14 14:28:03 UTC 2023
;; MSG SIZE  rcvd: 127
```

records that form the answer to the question

may be more than one

time to live of a resource record that is cached on the server
- try invoking **`dig`** once more to see it decreasing
- constant if the record is not cached (i.e., it is stored on the name server being queried – by default the one configured in **`/etc/resolv.conf`**)

step 7 – advanced queries

pc1

```
root@pc1:/# dig pc2.startup.net
; <<>> DiG 9.18.19-1~deb12u1-Debian <<>> pc2.startup.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22137
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: b4c95f2012cf8905d705a97c657b10f314be1a5abe3ec74d (good)
;; QUESTION SECTION:
;pc2.startup.net.               IN      A

;; ANSWER SECTION:
pc2.startup.net.        60000   IN      A       192.168.0.222

;; AUTHORITY SECTION:
startup.net.            59173   IN      NS      dnsstart.startup.net.

;; ADDITIONAL SECTION:
dnsstart.startup.net.   59173   IN      A       192.168.0.22

;; Query time: 0 msec
;; SERVER: 192.168.0.110#53(192.168.0.110) (UDP)
;; WHEN: Thu Dec 14 14:28:03 UTC 2023
;; MSG SIZE  rcvd: 127
```

records describing authoritative name servers are returned here

additional records are returned here

step 8 – an iterative query

- restart bind on the name server

localuni

```
root@localuni:/# systemctl restart bind9
```

- perform an iterative query using **dig**

pc1

```
root@pc1:/# dig +noquestion +noadditional +norecurse pc2.startup.net
```

avoid displaying question and additional sections

disable recursion

step 8 – an iterative query

the server answers by specifying the authoritative name server to be contacted to get the desired information

pc1

```
root@pc1:/# dig +noquestion +noadditional +norecurse pc2.startup.net

; <<>> DiG 9.18.19-1~deb12u1-Debian <<>> +noquestion +noadditional
+norecurse pc2.startup.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15543
;; flags: qr ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 5ea4ced6cdaf30599571a9e0657b15c2381005312bcc21e9 (good)
;; AUTHORITY SECTION:
.                       0       IN      NS      ROOT-SERVER.

;; Query time: 0 msec
;; SERVER: 192.168.0.110#53(192.168.0.110) (UDP)
;; WHEN: Thu Dec 14 14:48:34 UTC 2023
;; MSG SIZE  rcvd: 96
```


step 8 – an iterative query

query a specific name server (**dnsroot**)

dnsnet.net is the authoritative name server for zone **net**

pc1

```
root@pc1:/# dig +noquestion +noadditional +norecurse @192.168.0.5
pc2.startup.net

; <<>> DiG 9.18.19-1~deb12u1-Debian <<>> +noquestion +noadditional
+norecurse @192.168.0.5 pc2.startup.net
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24163
;; flags: qr ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 301c8e7f8267ad01ed2cc63e657b1736676e072f5ecd90bf (good)
;; AUTHORITY SECTION:
net.                    60000   IN      NS      dnsnet.net.

;; Query time: 0 msec
;; SERVER: 192.168.0.5#53(192.168.0.5) (UDP)
;; WHEN: Thu Dec 14 14:54:46 UTC 2023
;; MSG SIZE  rcvd: 109
```

step 8 – an iterative query

query a specific name server (**dnsnet.net**)

dnsstart.startup.net is the authoritative name server for zone **startup.net**

pc1

```
root@pc1:/# dig +noquestion +noadditional +norecurse @192.168.0.2
pc2.startup.net

; <<>> DiG 9.18.19-1~deb12u1-Debian <<>> +noquestion +noadditional
+norecurse @192.168.0.2 pc2.startup.net
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42339
;; flags: qr ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: d96ea5b92e6860aeed2cc63e657b1878e06d204302aa8149 (good)
;; AUTHORITY SECTION:
startup.net.            60000   IN      NS      dnsstart.startup.net.

;; Query time: 9 msec
;; SERVER: 192.168.0.2#53(192.168.0.2) (UDP)
;; WHEN: Thu Dec 14 15:00:08 UTC 2023
;; MSG SIZE  rcvd: 111
```


step 8 – an iterative query

query a specific name server (**dnsstart.startup.net**)

the address of **pc2.startup.net**

pc1

```
root@pc1:/# dig +noquestion +noadditional +norecurse @192.168.0.22
pc2.startup.net

; <<>> DiG 9.18.19-1~deb12u1-Debian <<>> +noquestion +noadditional
+norecurse @192.168.0.22 pc2.startup.net
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49113
;; flags: qr aa ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: b119e8f8b644792eb3bafbd6657b17989cd95e340adb2072 (good)
;; ANSWER SECTION:
pc2.startup.net.        60000   IN      A       192.168.0.222

;; AUTHORITY SECTION:
startup.net.            60000   IN      NS      dnsstart.startup.net.

;; Query time: 0 msec
;; SERVER: 192.168.0.22#53(192.168.0.22) (UDP)
;; WHEN: Thu Dec 14 14:56:24 UTC 2023
;; MSG SIZE  rcvd: 127
```