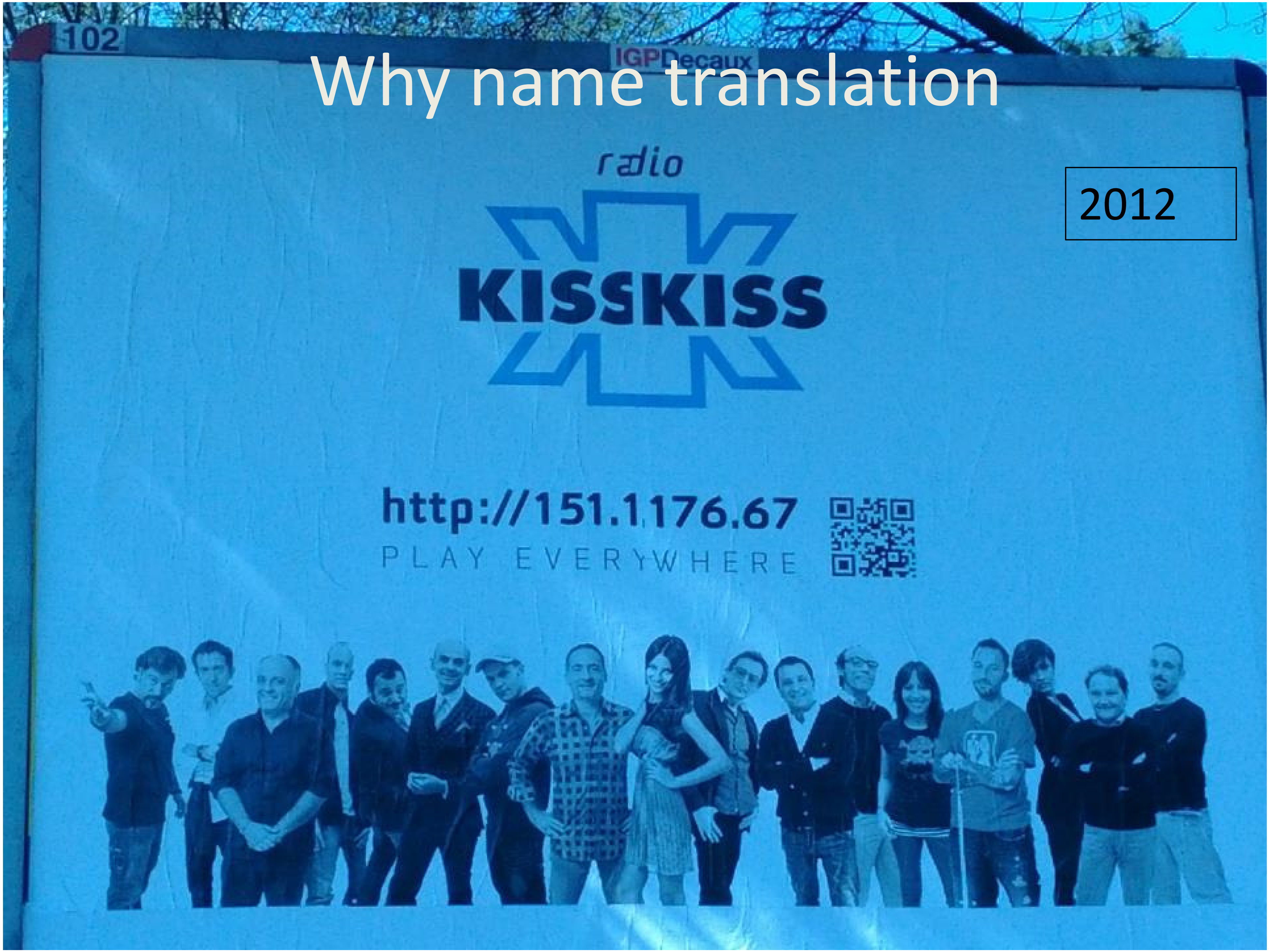
DNS & BIND

Francesco Zampognaro

Marco Bonola

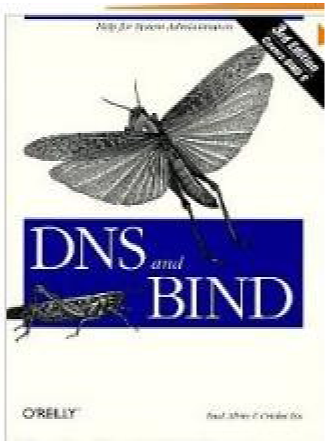
Why name translation

2012



Need for name translation

- initially because IP address are not human friendly
  - “mnemonic” names
- …imagine IPV6!
  - 2002:a050:6768:0:e2f8:47ff:fe38:c5cc:
- Important also for:
  - load balancing
  - decoupling IP and name (i.e. when changing hosting)
  - many other things (e.g. anti-spam!)
- Where to study:
  - DNS and BIND (O’ reilly)
  - Pro DNS and BIND (Aitchison)



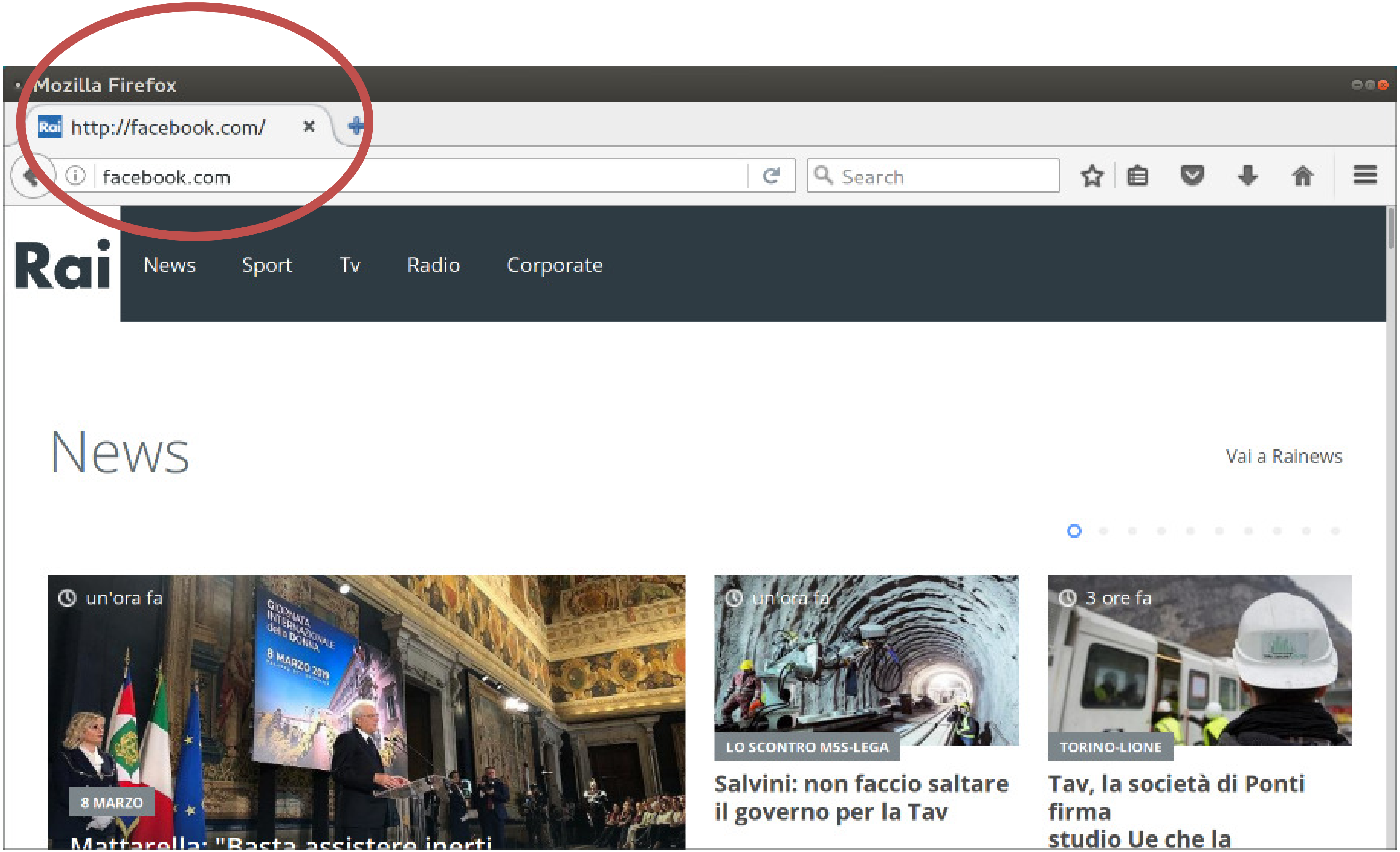
Before DNS…

- Elizabeth Feinler at the Stanford Research Institute, since 1972, maintained a text file named HOSTS.TXT that mapped host names to the numerical addresses of computers on the ARPANET
- Today…each computer has still a “local resolver” file hosts.txt
  - c:\Windows\System32\Drivers\etc\hosts
  - /etc/hosts

```
127.0.0.1     localhost
```

- Try to put in it:
  - 212.162.68.90   facebook.com
- Inefficiencies: name collisions, management, consistencies, scalability…
- For small testing setups it can be used. On pc2 (192.168.0.10):

```
192.168.0.2     pc1
192.168.0.1     gw
```

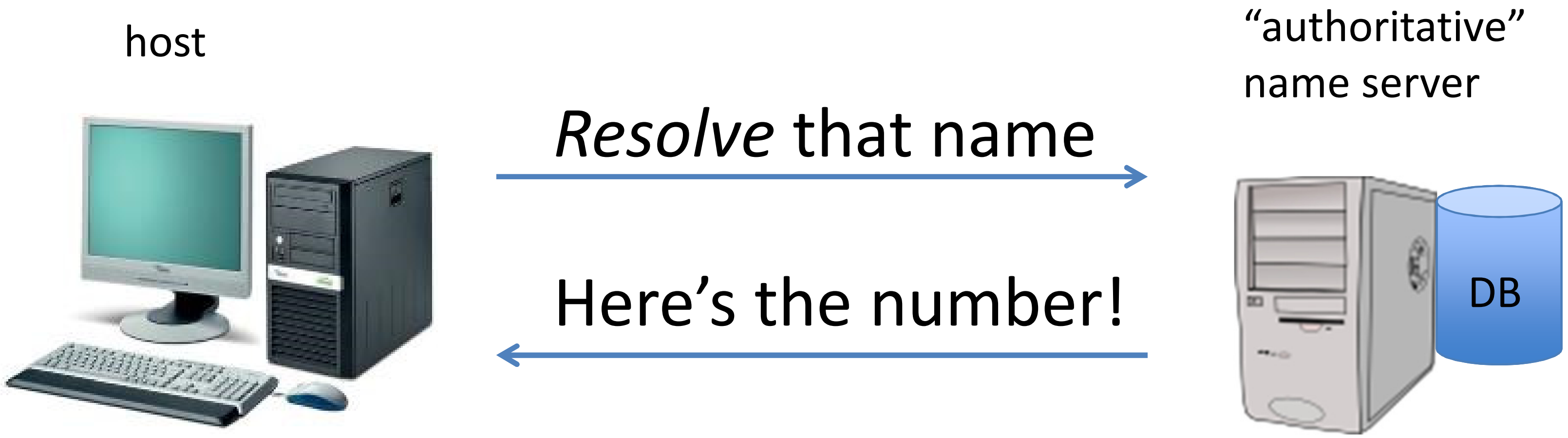


Mattarella: "Basta assistere inerti

Salvini: non faccio saltare il governo per la Tav

Tav, la società di Ponti firma
studio Ue che la

Simple solution



host

“authoritative” name server

Resolve that name

Here’s the number!

Definition of “bind” protocol (1984)

- need of a *scalable* solution (in 2018 > ~334M domains[1])
- avoid name collision
- Reliability
- introduce hierarchical names*: www.example.com.*
- Key concept: *authority and delegation*

“silent dot”

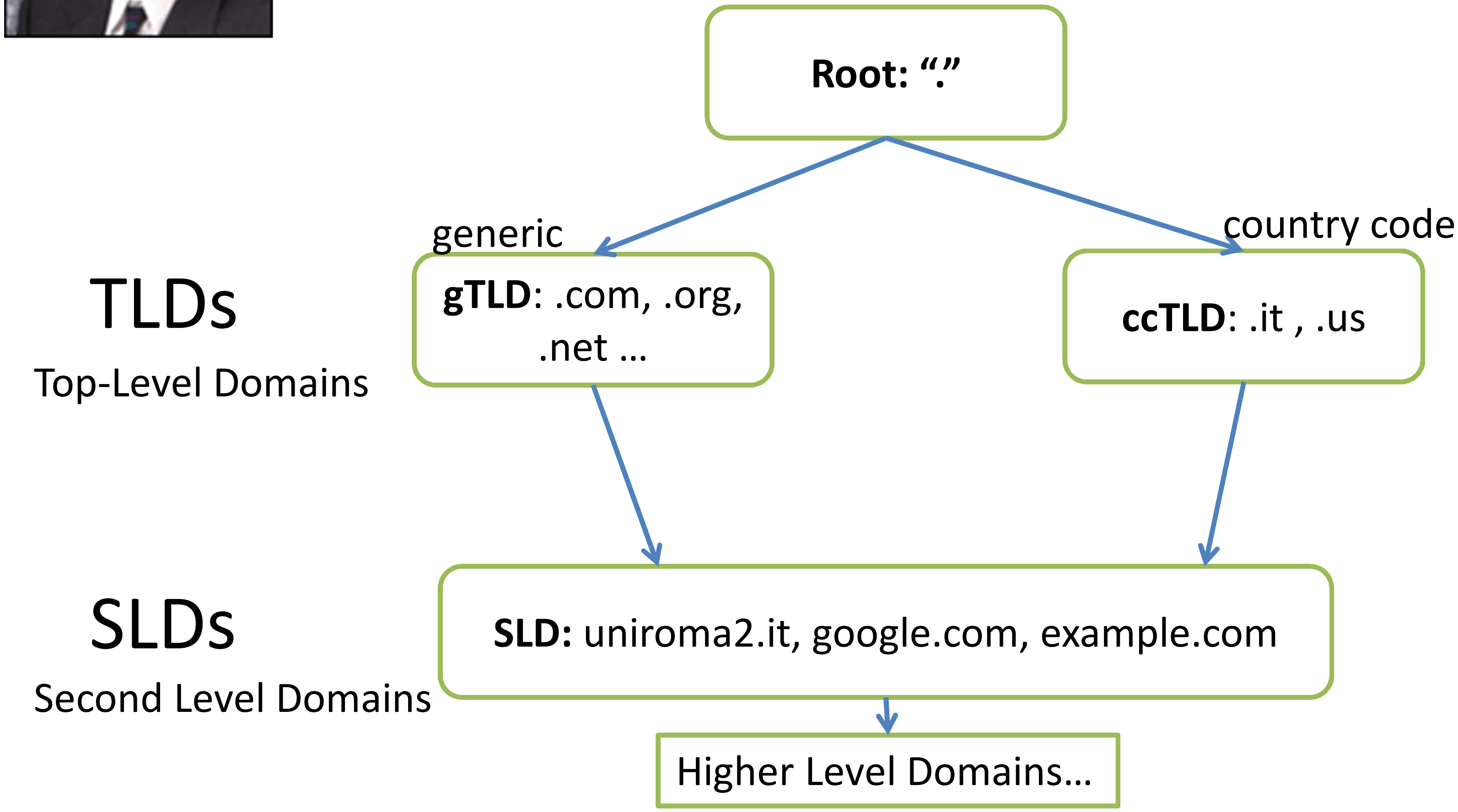
[1] https://blog.verisign.com/domain-names/verisign-q1-2018-domain-name-industry-brief-internet-grows-to-333-8-million-domain-name-registrations-in-the-first-quarter-of-2018/

Internet Domain Name System

- DNS's distributed database is indexed by domain names servers
- Each domain name is essentially just a path in a large inverted tree, called the *domain name space*
- Each node in the tree has a text label (without dots) that can be up to 63 characters long
- The full *domain name* of any node in the tree is the sequence of labels on the path from that node to the root
- An absolute domain name is also referred to as a *fully qualified domain name*, often abbreviated *FQDN*
- An AUTHORITATIVE name server is an authority for the domain name in question, managing local “trusted” information.
- Scalability is reached through DELEGATION (and secondly by master/slave setups and anycast)
- DNS requires that sibling nodes − nodes that are children of the same parent − have different labels. This restriction guarantees that a domain name uniquely identifies a single node in the tree (easier collision avoidance)

First experiment by Paul Mockapetris 1983

Internet Domain Name System



Root: “.”

generic

gTLD: .com, .org, .net …

country code

ccTLD: .it , .us

TLDs

Top-Level Domains

SLDs

Second Level Domains

SLD: uniroma2.it, google.com, example.com

Higher Level Domains…

A **Domain** is a string representing the realm of an **Authority**

for root: IANA (departement of ICANN—www.icann.org/)

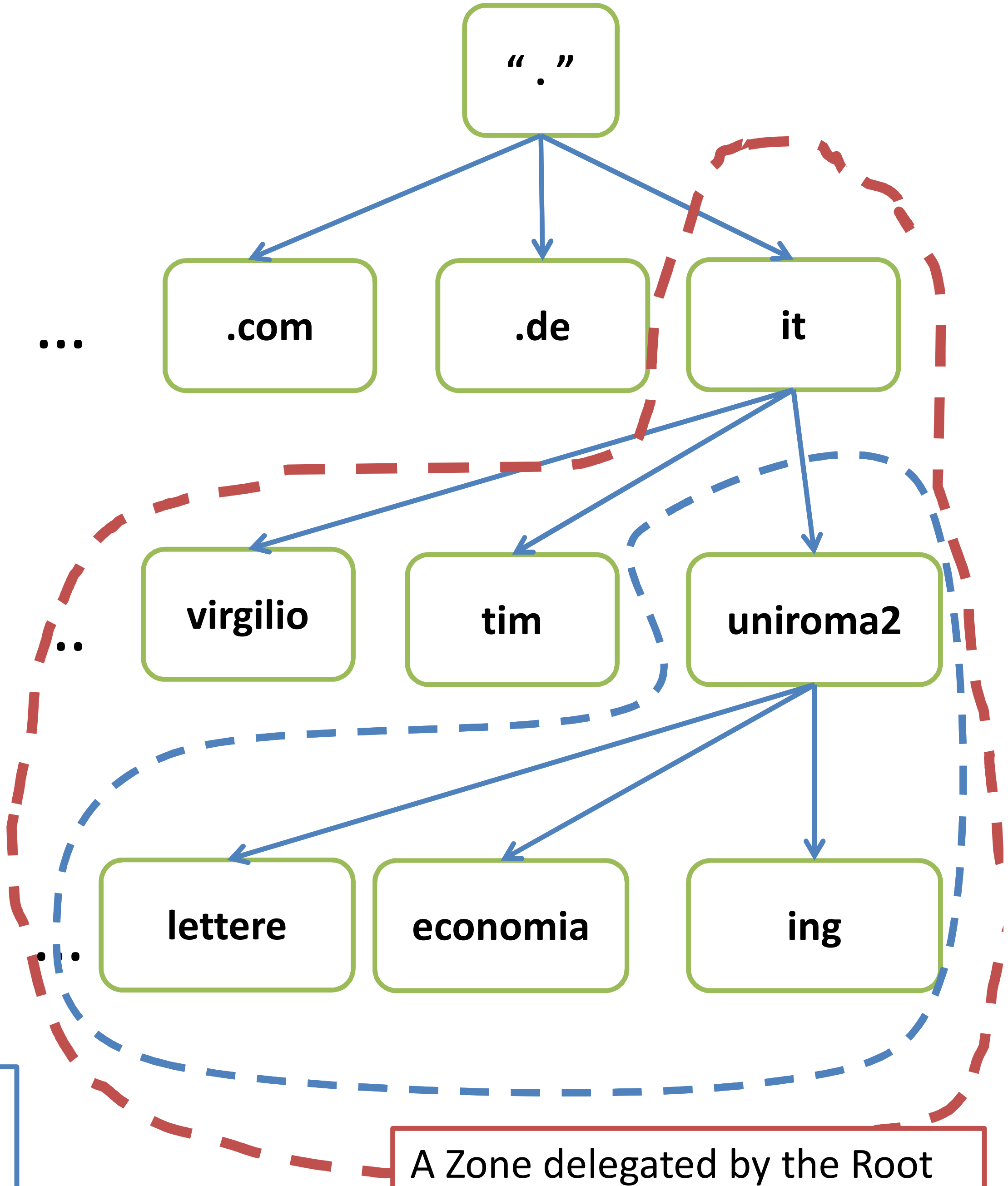
for .it: is @ Istituto per le Applicazioni Telematiche del CNR, PISA.

DNS Tree

- The administrative responsibility of part of the Domain Name Space can be delegated: this is called a zone

- The zone can sub-delegate

- Zone are represented using "zone" files (RFC 1034-1035)



A Zone sub-delegated to uniroma2

A Zone delegated by the Root Authority to the "IT" Authority

Resource Records

- Every “leaf” of the tree could have some Resource Records that contain information about the domain name
  - RR have different *standardized* types (e.g. A “direct”, PTR “inverse”, MX “mail ref”)
  - For instance, the IPv4 Address associated with a name (Resource Record of type A)

Registrar, Registry, Maintainer

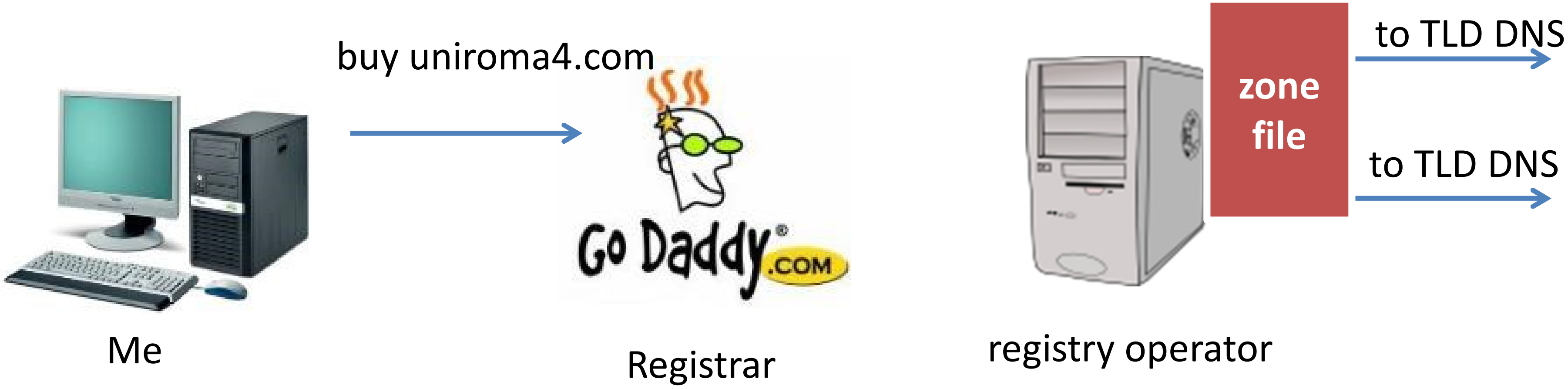
- **Registry**: organization that handle the database of names (e.g. TLD)
- **Registrar**: organization (also commercial) that can modify the registry database. Registrar has a frontend to the public:
  - – accredited by a gTLD or ccTLD:
    - Example http://www.nic.it/
  - – Works with “web pages” (*asynchronous*)
- **Maintainer**: frontend to the public
  - accredited by a gTLD or ccTLD
  - Works with FAX (*synchronous*) **OBSOLETE***

* From 1 July 2010 no more maintainer contracts for .it domains (source: registro.it)

Whois

```
aquilante:~ orazio$ whois uniroma2.it
Domain:             uniroma2.it
Status: ok
Signed:  no
Created:            1997-12-03 00:00:00
Last Update:        2019-01-30 00:53:17
Expire Date:        2020-01-14
Registrant
  Organization:     Universita' degli Studi di Roma "Tor Vergata"
  Address:          Via Orazio Raimondo, 18
(  …. )
Admin Contact
(…)
Technical Contacts
 (…)
Registrar
  Organization:     Universita' degli Studi di Roma "Tor Vergata"
  Name:             UNIROMA2-REG
  DNSSEC:           no
Nameservers
  dns.uniroma2.it
  dns1.uniroma2.it
  ns1.garr.net
```

Updating names: let’s buy a “domain”



buy uniroma4.com

zone file

to TLD DNS

to TLD DNS

Me

Registrar

registry operator

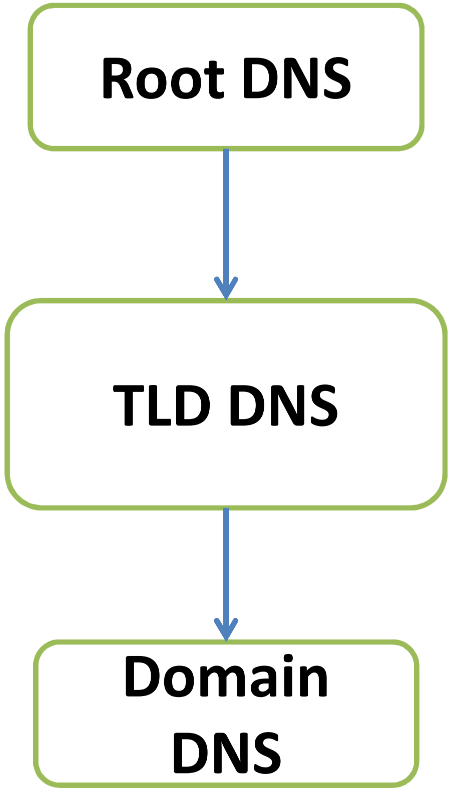
- A registrar interacts with end-users, stores detailed information, and passes a “digest” to a registry operator.
- Registry operator updates a “zone file” (i.e. Data describing the domain) and passes it to interested TLD
- Periodically, ICANN distribute a “TLD master file” to each Root DNS Server.

www.example.com

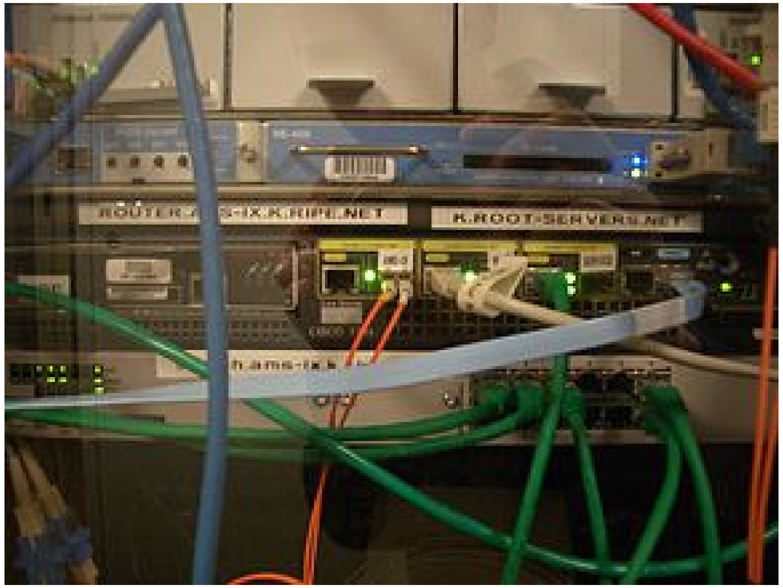
- The domain name example.com was delegated from a **gTLD authority**, which in turn was delegated from **ICANN** (authority for DNS Root Zone)
- The owner of the domain chooses the www part (called host name)
- This is a Fully Qualified Domain Name (**FQDN**)
  - specifies an exact location in the DNS tree hierarchy

DNS Implementation

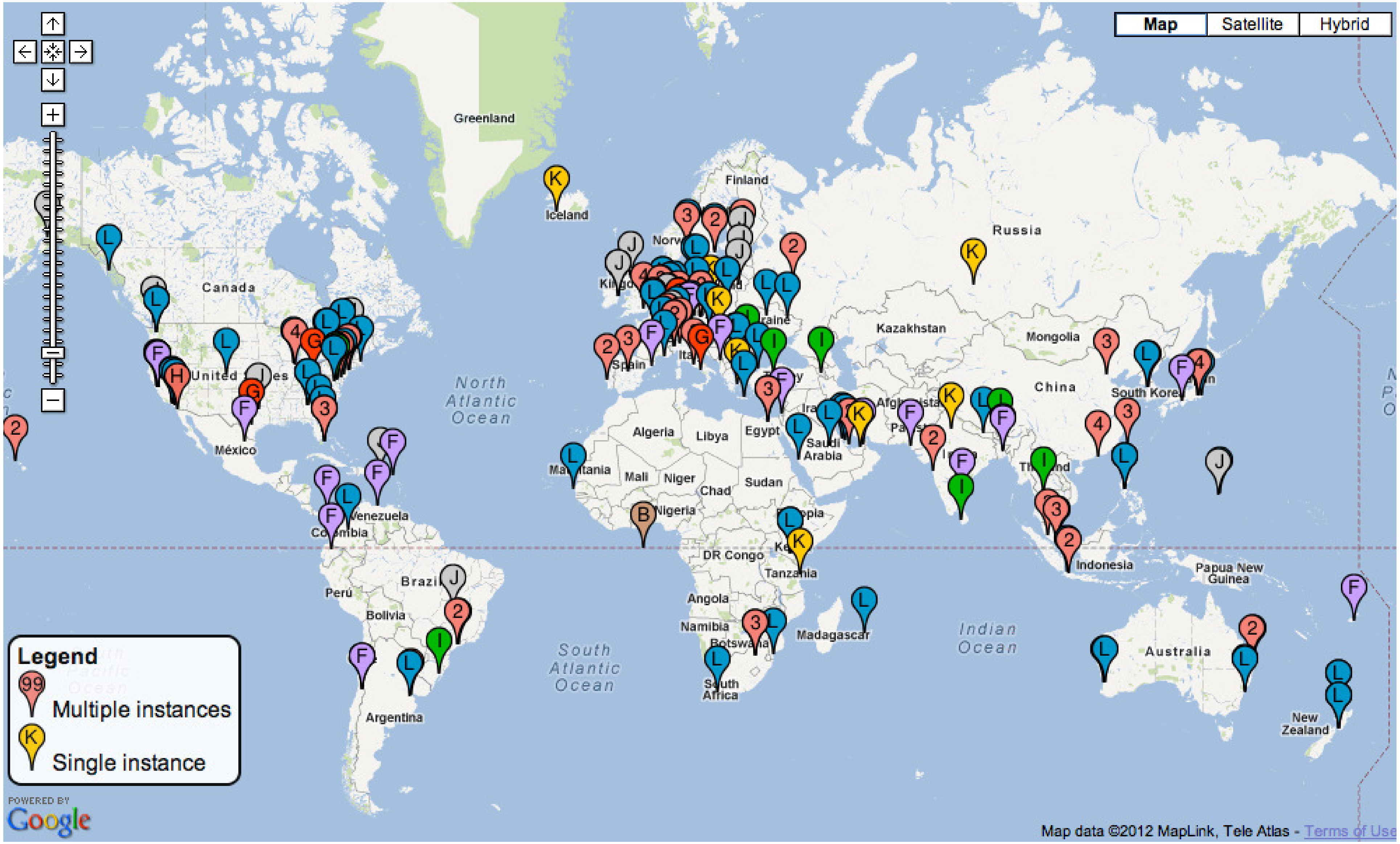
- Exactly maps the domain name delegation structure



13 root-servers
(from a.root-servers.net to m)



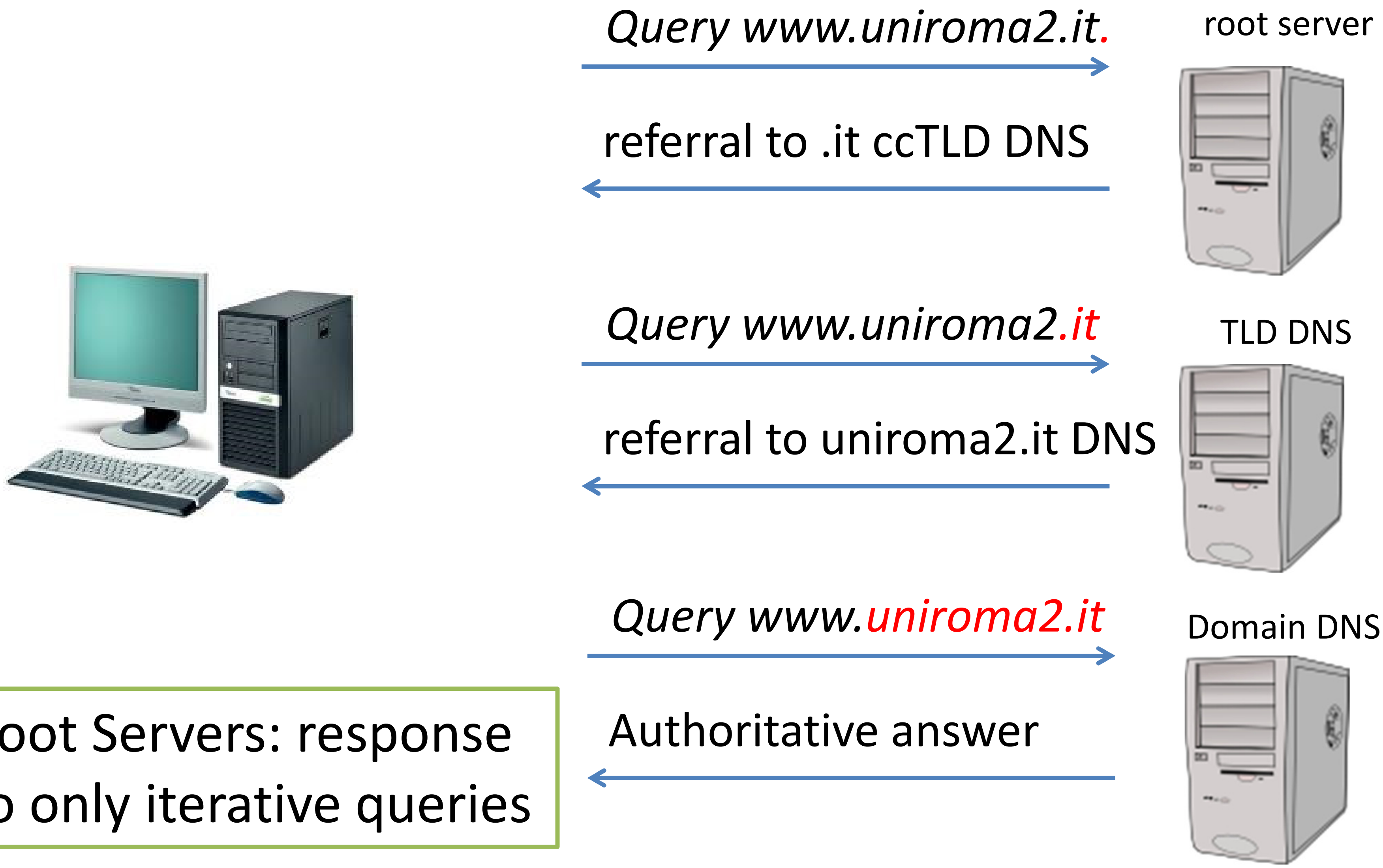
Root servers (anycast)



DNS attacks to root servers

- Wikipedia: “During two intervals on November 30, 2015 and December 1, 2015, several of the root name servers received up to 5 million queries per second each, receiving valid queries for a single undisclosed domain name, and then a different domain the next day. Source addresses were spread throughout IPv4 space, however these may have been spoofed. Some root server networks became saturated, resulting in timeouts, however redundancy among the root servers prevented downstream issues from occurring during this incident.”

DNS Queries: iterative vs recursive



root server

Query www.uniroma2.it.

referral to .it ccTLD DNS

TLD DNS

Query www.uniroma2.it

referral to uniroma2.it DNS

Domain DNS

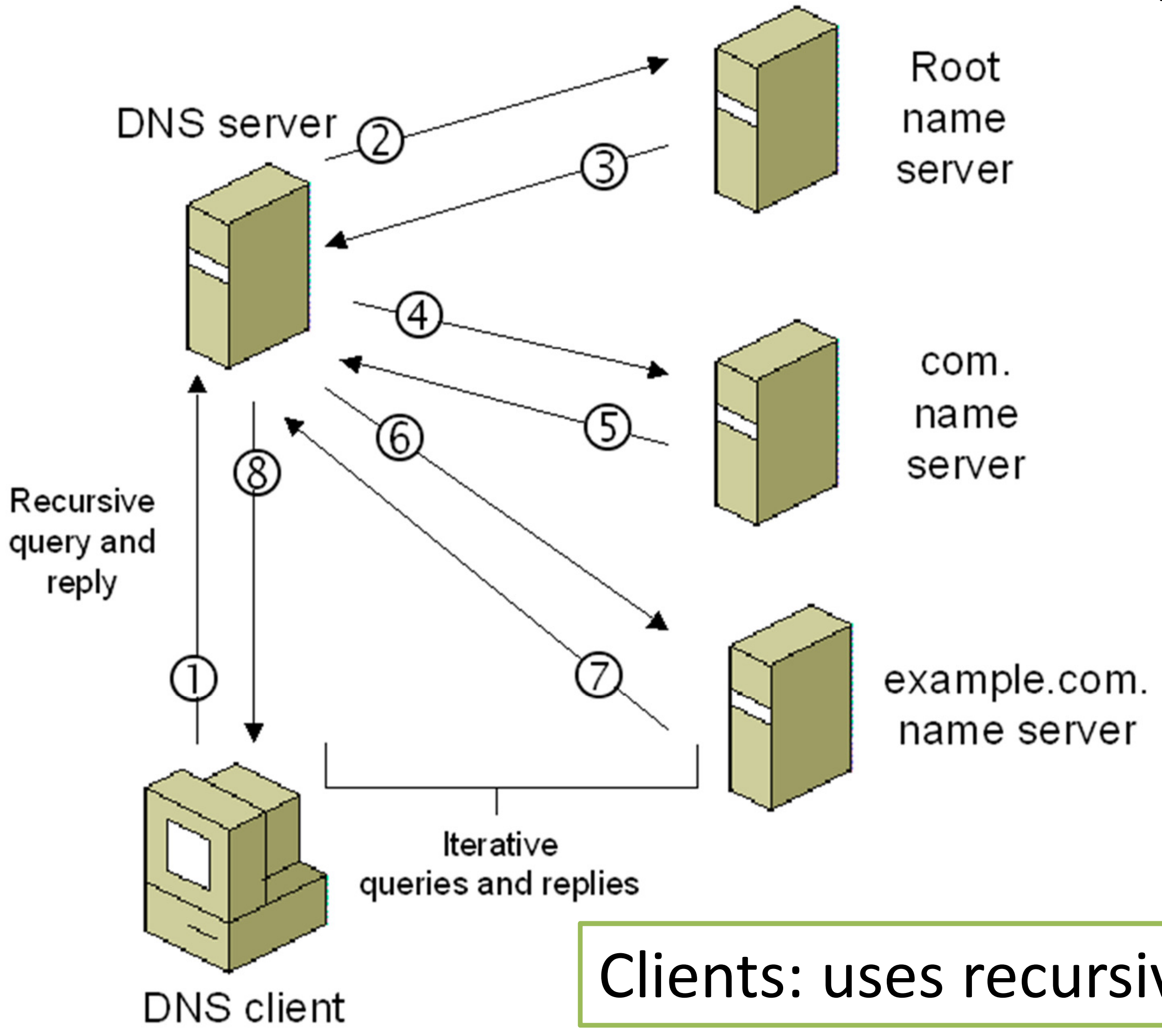
Query www.uniroma2.it

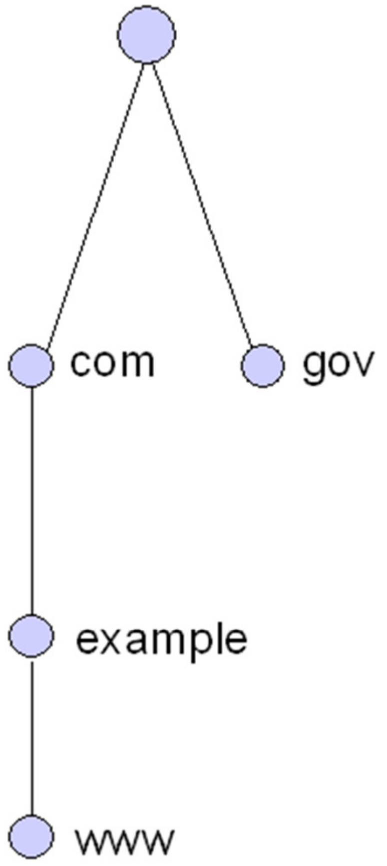
Authoritative answer

Root Servers: response to only iterative queries

DNS Queries: iterative vs recursive

i.e. find an answer





Clients: uses recursive queries only

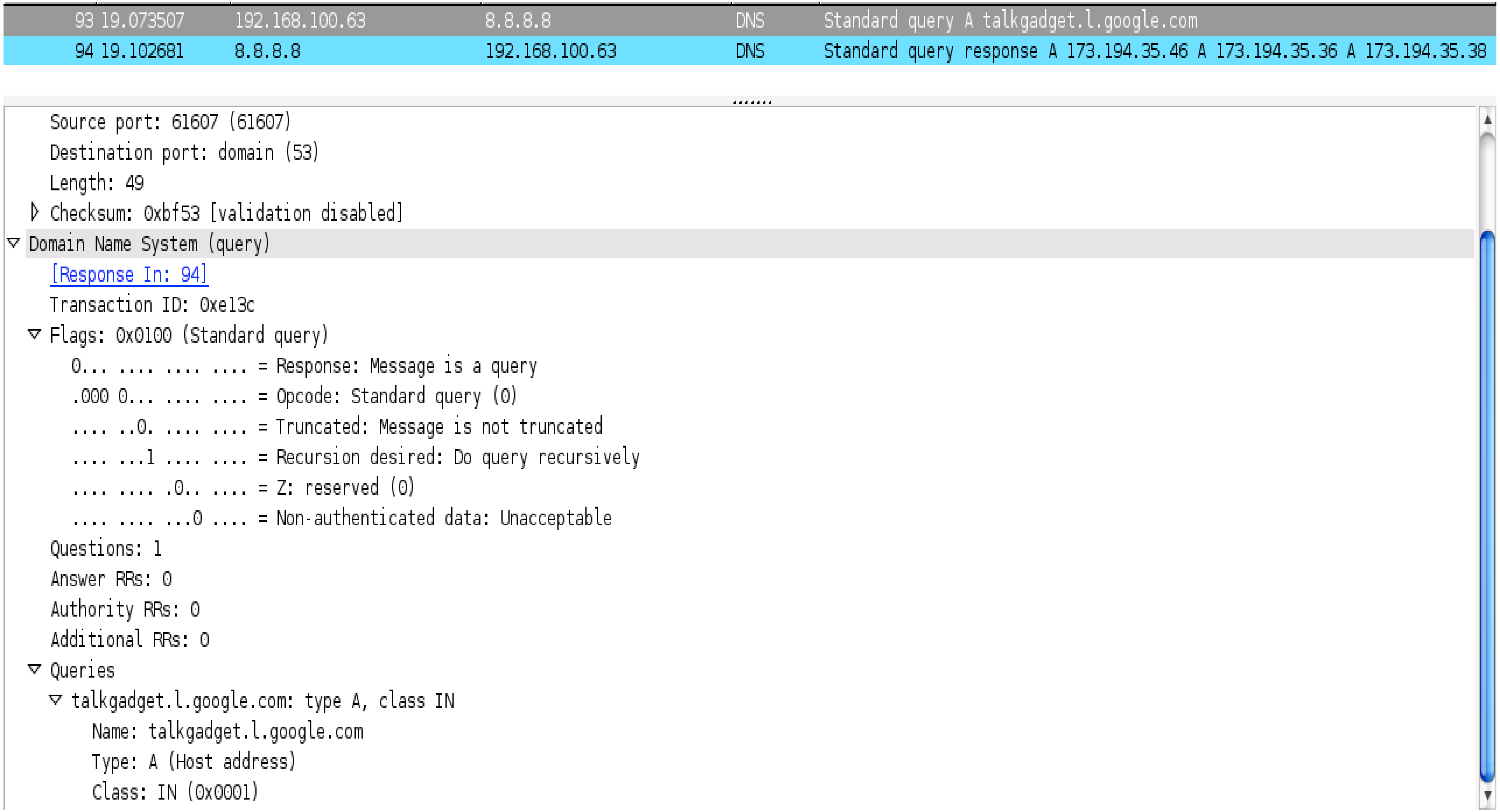
DNSSEQ

- DNS threats are well known: RFC 3833
- Not to hide the request, but to sign it
- https://dnssec.vs.uni-due.de/
  - – Yes, your DNS resolver validates DNSSEC signatures.

- Enabled in root DNS servers since 2011
- Requires support in all zones down to the FQDN

trace in http://stud.netgroup.uniroma2.it/cgrl/2012/traces/dns.pcap

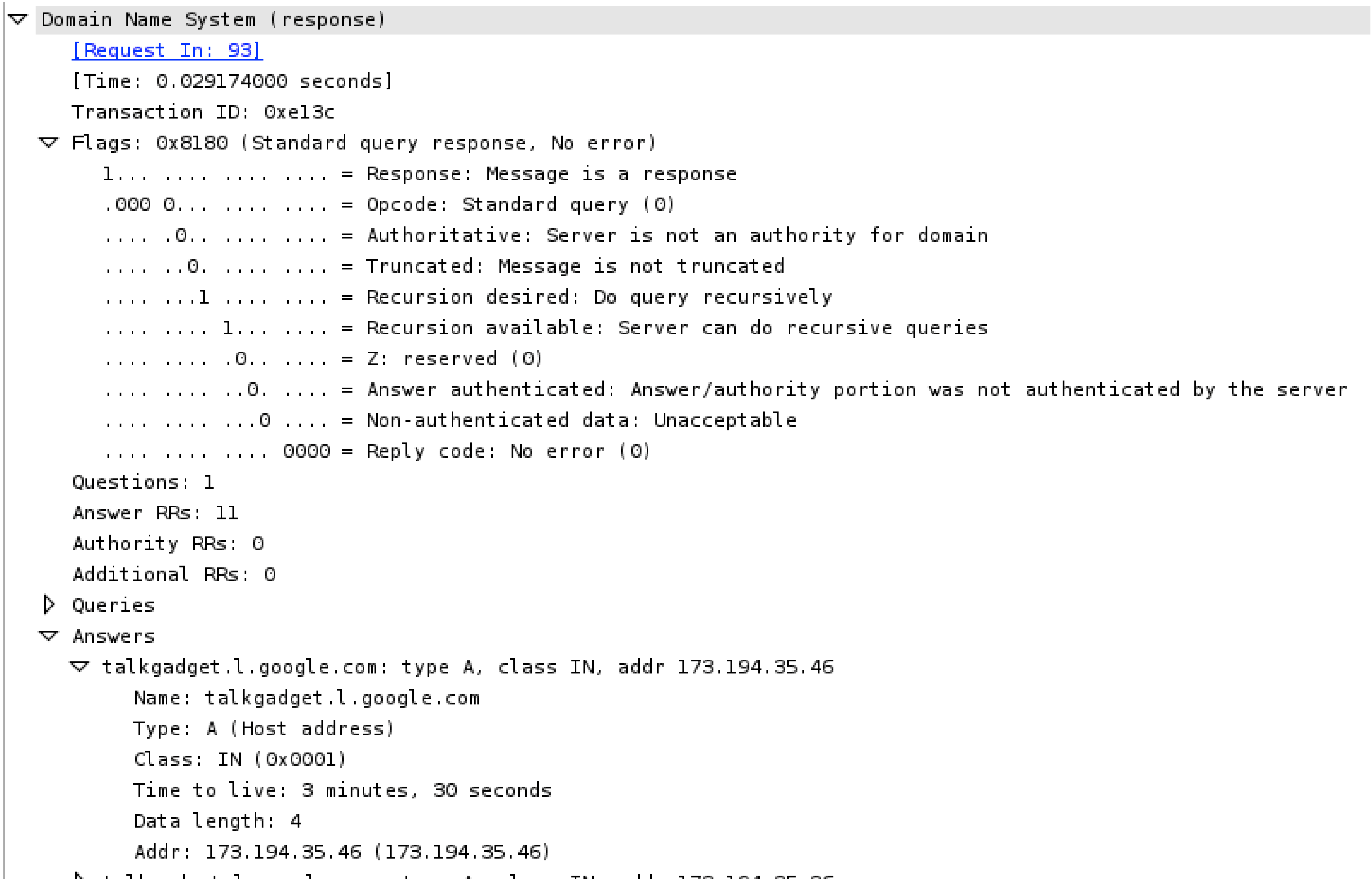
DNS Queries



```
     93 19.073507     192.168.100.63      8.8.8.8             DNS      Standard query A talkgadget.l.google.com
     94 19.102681     8.8.8.8             192.168.100.63      DNS      Standard query response A 173.194.35.46 A 173.194.35.36 A 173.194.35.38
```

```
    Source port: 61607 (61607)
    Destination port: domain (53)
    Length: 49
  ▷ Checksum: 0xbf53 [validation disabled]
▽ Domain Name System (query)
    [Response In: 94]
    Transaction ID: 0xe13c
  ▽ Flags: 0x0100 (Standard query)
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▽ Queries
    ▽ talkgadget.l.google.com: type A, class IN
        Name: talkgadget.l.google.com
        Type: A (Host address)
        Class: IN (0x0001)
```

Dns Response



```
▽ Domain Name System (response)
    [Request In: 93]
    [Time: 0.029174000 seconds]
    Transaction ID: 0xe13c
  ▽ Flags: 0x8180 (Standard query response, No error)
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 11
    Authority RRs: 0
    Additional RRs: 0
  ▷ Queries
  ▽ Answers
    ▽ talkgadget.l.google.com: type A, class IN, addr 173.194.35.46
        Name: talkgadget.l.google.com
        Type: A (Host address)
        Class: IN (0x0001)
        Time to live: 3 minutes, 30 seconds
        Data length: 4
        Addr: 173.194.35.46 (173.194.35.46)
```

DNS Resolver

- The client-side of the DNS is usually called a DNS resolver.
- On PC, we usually have simple resolvers (called "stub resolvers") that can not follow referrals
  - Need a recursive-enabled DNS server
- UNIX systems use *gethostbyname* or *gethostbyaddr* methods to invoke name/ip resolution
  - functions provided by the stub resolver
- UNIX has also a Swiss-knife command for DNS analysis: dig

Dig

debian package: *dnsutils*

```
$>  dig uniroma2.it SOA
; <<>> DiG 9.9.5-11ubuntu1.3-Ubuntu <<>> uniroma2.it SOA
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7649
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 4

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;uniroma2.it.                    IN      SOA

;; ANSWER SECTION:
uniroma2.it.        3600  IN     SOA    dns.uniroma2.it. postmaster.uniroma2.it.
2019031100 86400 7200 604800 86400

;; AUTHORITY SECTION:
uniroma2.it.        3600  IN     NS     ns1.garr.net.
uniroma2.it.        3600  IN     NS     dns1.uniroma2.it.
uniroma2.it.        3600  IN     NS     dns.uniroma2.it.

;; ADDITIONAL SECTION:
dns.uniroma2.it.    1514  IN     A      160.80.1.3
dns.uniroma2.it.    2448  IN     AAAA 2001:760:4016:1::3
dns1.uniroma2.it.  2553  IN     A      160.80.5.8

;; Query time: 2 msec
;; SERVER: 160.80.1.8#53(160.80.1.8)
;; WHEN: Tue Mar 12 16:06:59 CET 2019
;; MSG SIZE  rcvd: 210
```

Dig

Examples:

- dig @8.8.8.8 www.google.com
  - resolve with the 8.8.8.8 DNS
- dig @8.8.8.8 www.google.com +trace
  - recursively do all the queries
- dig . ns
  - show in short form all the ns fields of root servers
- dig -x 164.132.251.209 +short
  - reverse lookup (short output)

Nslookup - host

- Other commands

```
$> nslookup repubblica.it (opz. DNS to use)
Server:          160.80.1.8
Address:  160.80.1.8#53

Non-authoritative answer:
Name:     repubblica.it
Address: 213.92.16.101

host (opz. –t record=NS/SOA/MX/…)  URL
```

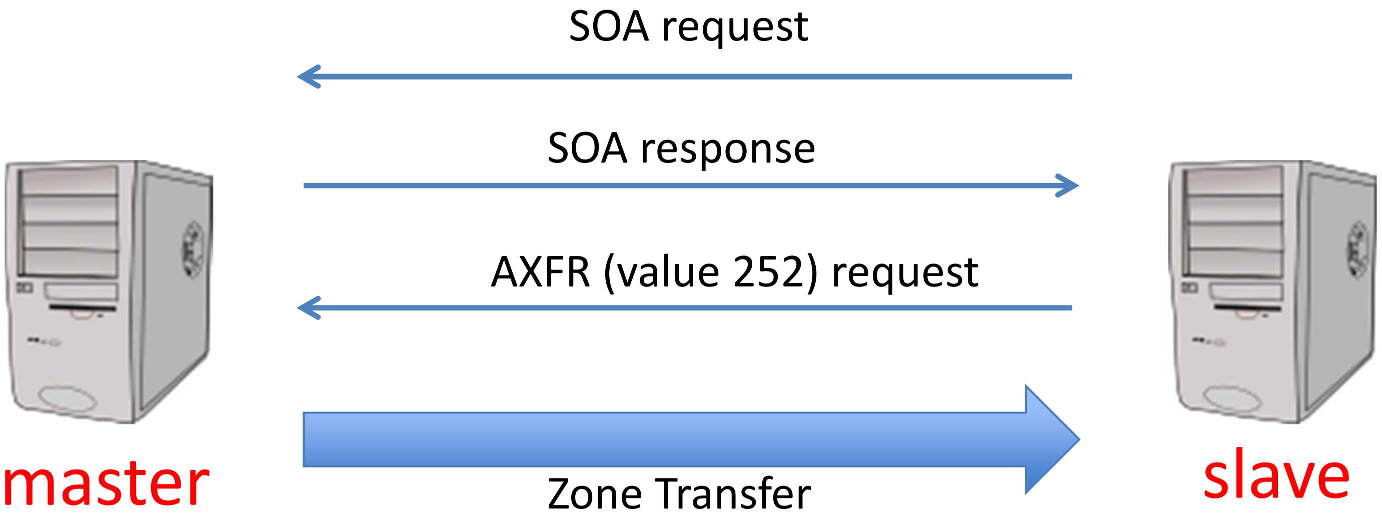
tcpdump for dns

```
tcpdump –n –t port domain –i any –s0
IP 192.168.0.111.3072 > 192.168.0.11.53:
    34896+ A? www.uniroma2.it. (36)
```

Fields:

- Query ID (+ means recursion preferred)
- Query type (find A record)
- Query value (for ? www.uniroma2.it.)
- Lenght of pkt

Distributed DNS configuration



- Initiated by client (slave), unless a "NOTIFY" is received
- redundancy for load balancing and fault resilience
- zones are passed from master to slave
  - full or partial zone transfer
- timing? sync?

Start of Authority record (abbreviated as **SOA record**) containing administrative information about the zone.

RFC 1035

Zone File: Example

```
$ORIGIN example.com.    ; changes the 'zone name' which is added to any 'unqualified' name
$TTL 1h                 ; default expiration time TTL value
example.com.  IN  SOA  ns.example.com. myemail.example.com. (
          2007120710 ; serial number of this zone file
          1d      ; slave refresh (1 day)
          2h      ; slave retry time in case of a problem (2 hours)
          4w      ; slave expiration time (4 weeks)
          1h      ; maximum caching time in case of failed lookups (1 hour)
          )
example.com.  NS   ns        ; ns.example.com is a nameserver for example.com
example.com.  NS   ns.somewhere.example. ; a backup nameserver for example.com
example.com.  MX   10 mail.example.com.  ; the mailserver for example.com
@          MX   20 mail2.example.com. ; equivalent to above line, "@" represents zone origin
@          MX   50 mail3           ; equivalent to above line, but using a relative host name
example.com.  A    192.0.2.1          ; IPv4 address for example.com
              AAAA  2001:db8:10::1     ; IPv6 address for example.com
ns         A    192.0.2.2          ; IPv4 address for ns.example.com
           AAAA  2001:db8:10::2     ; IPv6 address for ns.example.com
mail        A    192.0.2.3          ; IPv4 address for mail.example.com,
mail2       A    192.0.2.4          ; IPv4 address for mail2.example.com
mail3       A    192.0.2.5          ; IPv4 address for mail3.example.com
www        CNAME example.com.       ; www.example.com is an alias for example.com
```

Comments

directives

SOA RR

NS RR

MX RR

A and AAAA RR

CNAME RR

Resource Records (RR)

- A Start of Authority (SOA) RR :
  - describes global characteristics of the zone domain
  - one and only one for each zone file (first RR in a zone file)
- Name Server (NS) RR: Defines name servers that are authoritative for the zone or domain. There must be two or more NS Resource Records in a zone file. NS RRs may reference servers in this domain or in a foreign or external domain. These RRs are mandatory.
- Mail Exchanger (MX) RR: Defines the mail servers for the zone (optional)
- Address (A) RR: Define the IPv4 address of all the hosts (or services) that exist in this zone and which are required to be publicly visible. IPv6 entries are defined using AAAA (called Quad A) RRs (optional)
- Canonical Name (CNAME) RR: Defines an Alias RR, which allows one host (or service) be defined as the alias name for another host (optional)
- And: PTR, TXT, SRV and NSEC, RRSIG, DS, DNSKEY, KEY (DNSSEC)

Syntax: SOA RR

- Specifies authoritative information about a DNS zone

Zone Domain	Class	RR	NS	email dnsmaster
example.com.	IN	SOA	ns.example.com.	email.example.com.

- Several parameters
  - serial: date (convention: YYYYMMDDSS)
  - refresh: tell to slave how often check for changes (default 3600)
  - retry: interval between two subsequent attempt to contact the master in case of problems (default 600)
  - expire: if slave fails to contact master after expire time, it stops to resolve that zone (default 86400)
  - ttl The minimum time-to-live value applies to all resource records in the zone file (default 3600)

Syntax: NS RR

- Delegates a DNS zone to use the given authoritative name servers

Zone Name	TTL	class	rr	dns name
example.com.		IN	NS	ns1.example.com.

- The name field can be any of:
  - A Fully Qualified Domain Name (FQDN) e.g. example.com. (ends with a dot)
  - An unqualified name (does not end with a dot)
  - An '@' (substitutes the current value of $ORIGIN)
  - a 'space' or 'blank' (tab) - this is replaced with the previous value of the name field. If no name has been previously defined this may result in the value of $ORIGIN.

Syntax: A RR

- Resolve a name to a IPv4 address

Name	TTL	class	rr	Address
example.com.		IN	A	93.184.216.119

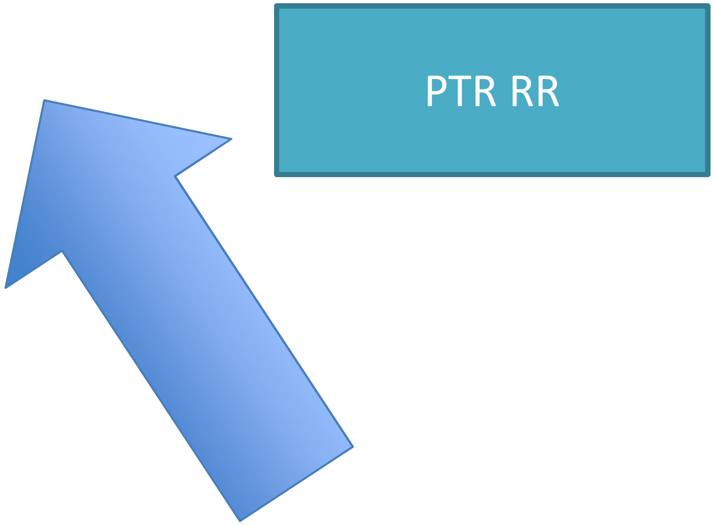
- CNAME: aliases
- MX, TXT, PTR: other options (discussed later)

Reverse Mapping

- How to find the name corresponding to 1.2.3.4?
  - And more generally, how to build a tree to keep the structure scalable (as in the case of name) ?
  - but…why? example: the **anti-spam** case. RFC 1912[1] "every Internet-reachable host should have a name" and "for every IP address, there should be a matching PTR record“. It is not an Internet Standard requirement, so not all IP addresses have a reverse entry. But “good” servers do.
- Flip the IP and search in the IN-ADDR.ARPA domain
  - Several names corresponds to the same IP (we will see that in virtual-hosting). We can only get the “main”!!

Reverse Mapping: zone file

```
…
$ORIGIN 254.168.192.IN-ADDR.ARPA.
…
17 IN PTR www.example.org
```



PTR RR

Example reverse DNS loockup:
dig -x 213.92.16.101
dig repubblica.it
dig kata-redir.kataweb.it

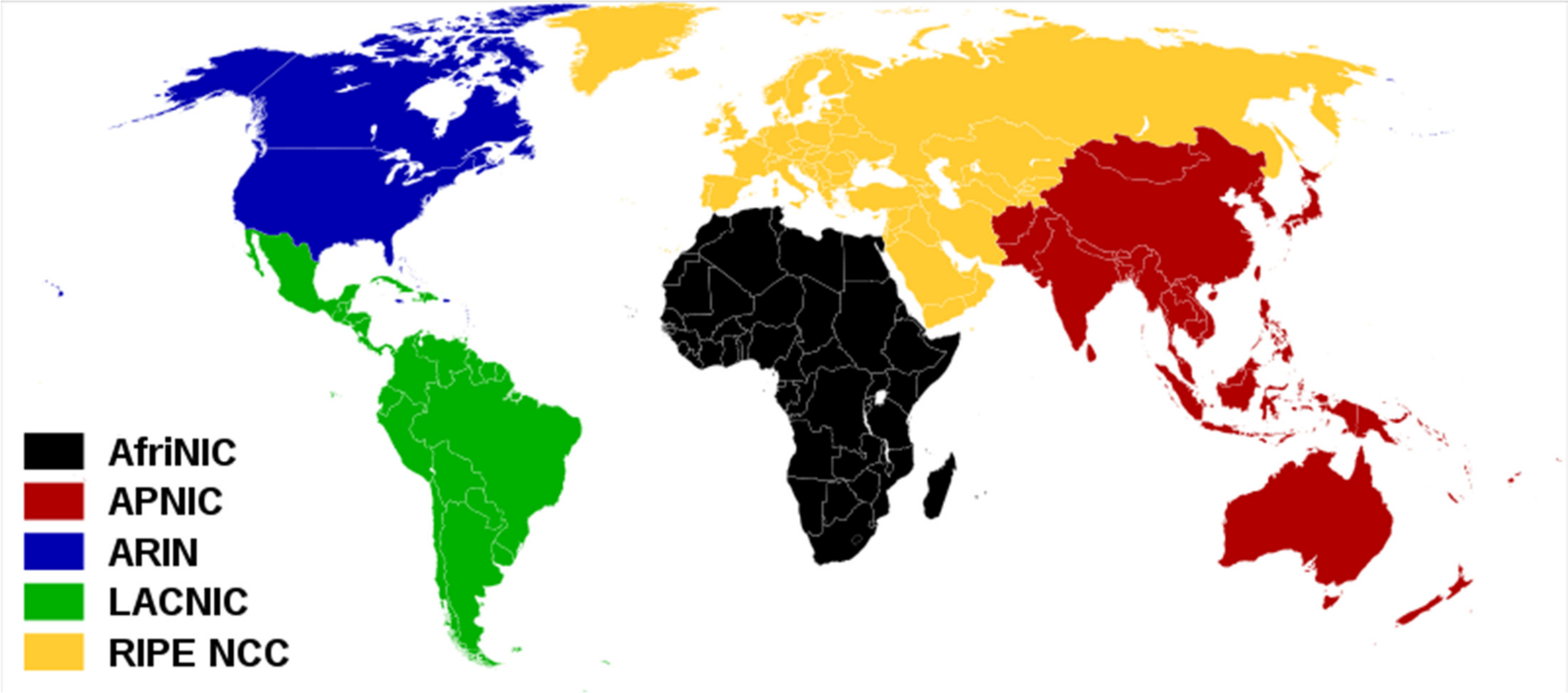
192.168.254.17

Reverse Mapping

- Completely different search!! The DNS tree structure does not help.
  - this part of the domain name space is structured according to address, and hence can guarantee that the appropriate data can be located without an exhaustive search of the domain space.
- IPv4 addresses are allocated in netblocks by the RIRs ….

RIRs

- Regional Internet Registry
- Manage IP addresses and AS numbers



Reverse Mapping

- IPv4 addresses are allocated in netblocks by the RIRs to either a Local Internet Registry, LIR (typically ISP, or National Internet Registry (NIR), which in turn will allocate to an LIR.)
- Each Internet Registry level is delegated the responsibility for reverse mapping the addresses it has been assigned.
- The LIR may delegate the responsibility for reverse mapping to the end user

Italian LIRs

https://www.ripe.net/membership/indices/IT.html

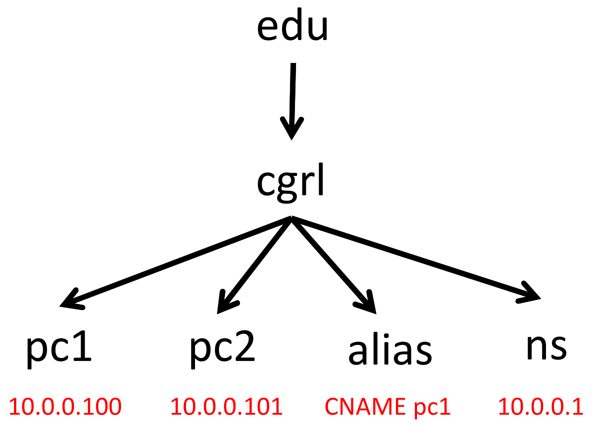
Interested? Search for **Internet Governance**
http://en.wikipedia.org/wiki/Internet_governance

Things are getting serious!

BIND

First simple example: cgrl.edu

DNS (ns.cgrl.edu.) is the authoritative name server for the zone cgrl.edu.


edu

cgrl

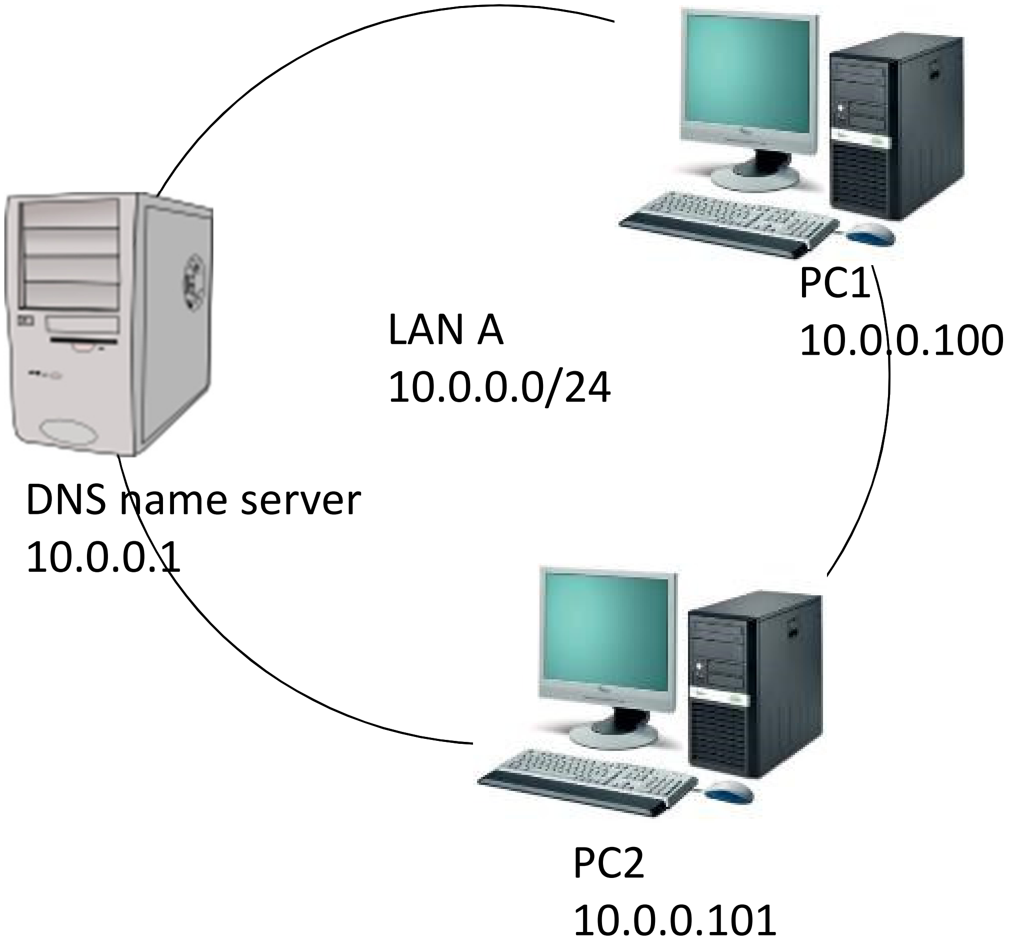
pc1 10.0.0.100

pc2 10.0.0.101

alias CNAME pc1

ns 10.0.0.1

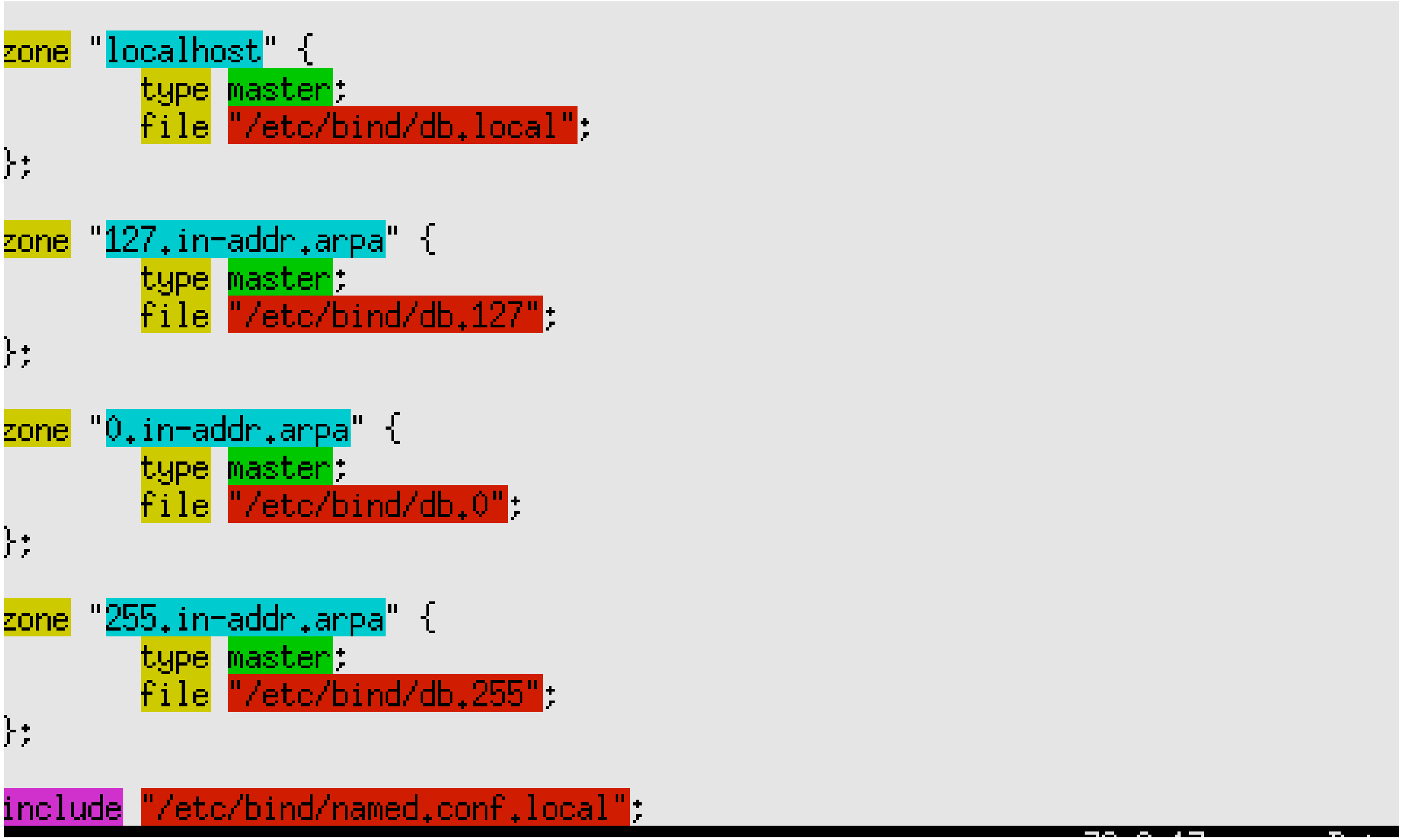
pc1.cgrl.edu
pc2.cgrl.edu
alias.cgrl.edu
ns.cgrl.edu



Bind

- bind executable: /usr/sbin/named
- rndc: command line administration of the named daemon
- Like many daemons got its start/stop script in /etc/init.d
  - /etc/init.d/bind [start stop restart status reload]
- Good news! Only one (usually short) conf file: /etc/bind/named.conf
- Bad news! it includes several other files!! such as:
    - Zone files: in /etc/bind/. Example: db.edu.cgrl
    - options: /etc/bind/named.conf.options
    - other files

/etc/bind/named.conf

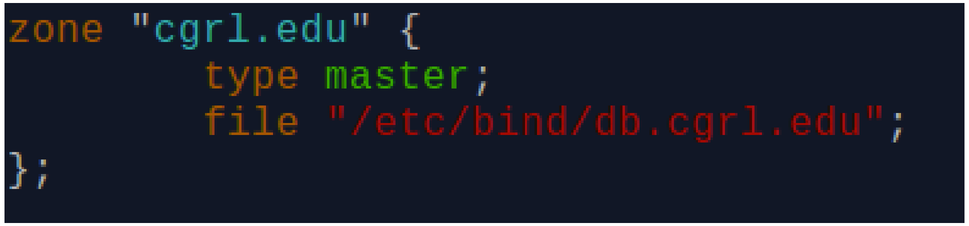


```
zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
        type master;
        file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
        type master;
        file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
        type master;
        file "/etc/bind/db.255";
};

include "/etc/bind/named.conf.local";
```

FIRST STEP: Add a zone for cgrl.edu to /etc/bind/db.edu.cgrl

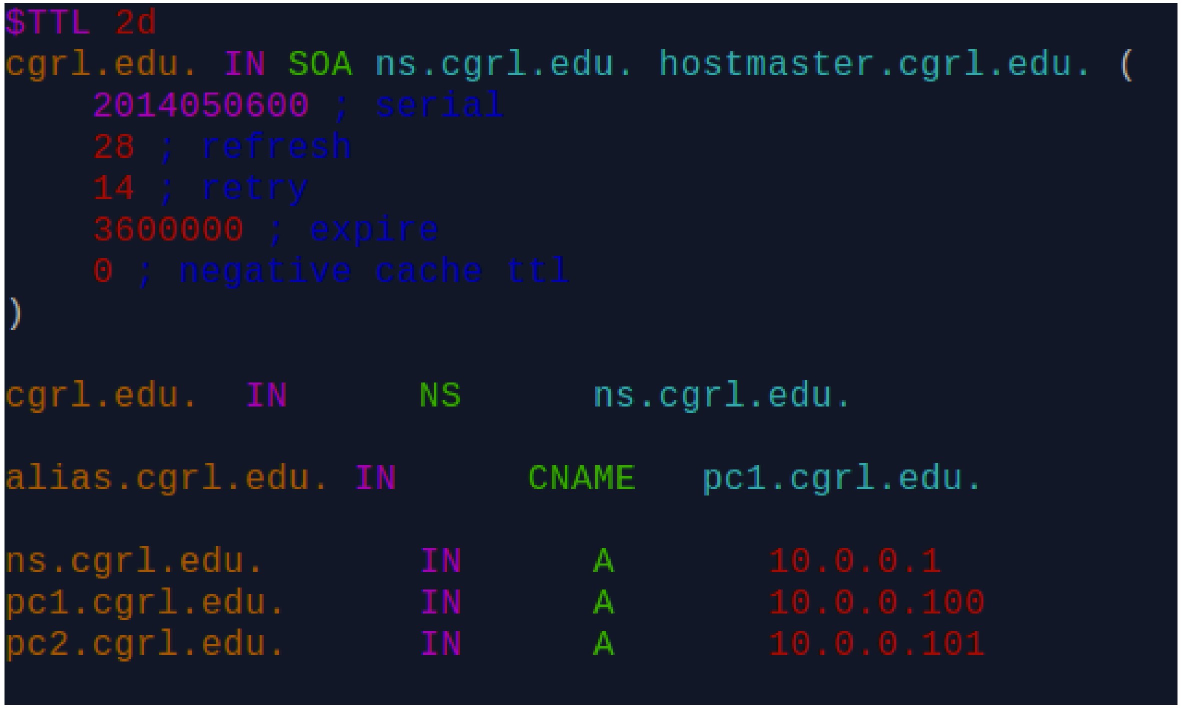
BIND configuration

/etc/bind/named.conf



```
zone "cgrl.edu" {
         type master;
         file "/etc/bind/db.cgrl.edu";
};
```

/etc/bind/db.edu.cgrl

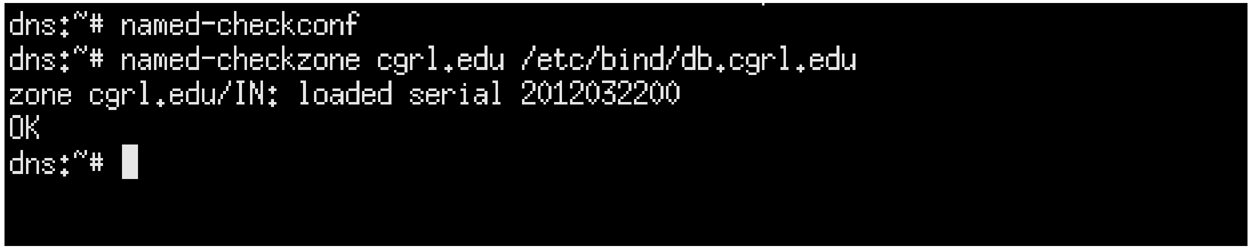


```
$TTL 2d
cgrl.edu. IN SOA ns.cgrl.edu. hostmaster.cgrl.edu. (
    2014050600 ; serial
    28 ; refresh
    14 ; retry
    3600000 ; expire
    0 ; negative cache ttl
)

cgrl.edu.  IN      NS      ns.cgrl.edu.

alias.cgrl.edu. IN      CNAME   pc1.cgrl.edu.

ns.cgrl.edu.       IN      A       10.0.0.1
pc1.cgrl.edu.      IN      A       10.0.0.100
pc2.cgrl.edu.      IN      A       10.0.0.101
```

NOTE: we are not using wildcards and special characters… more later on

Check BIND configuration

- To check zone files:
  - named-checkzone $ZONE_NAME $ZONE_FILE
- To check conf files:
  - named-checkconf
- View in syslog (or, if in another log file if you changed it)

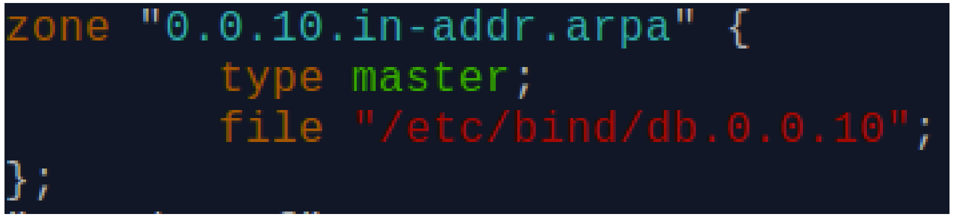


```
dns:~# named-checkconf
dns:~# named-checkzone cgrl.edu /etc/bind/db.cgrl.edu
zone cgrl.edu/IN: loaded serial 2012032200
OK
dns:~#
```

And for reverse address mapping?

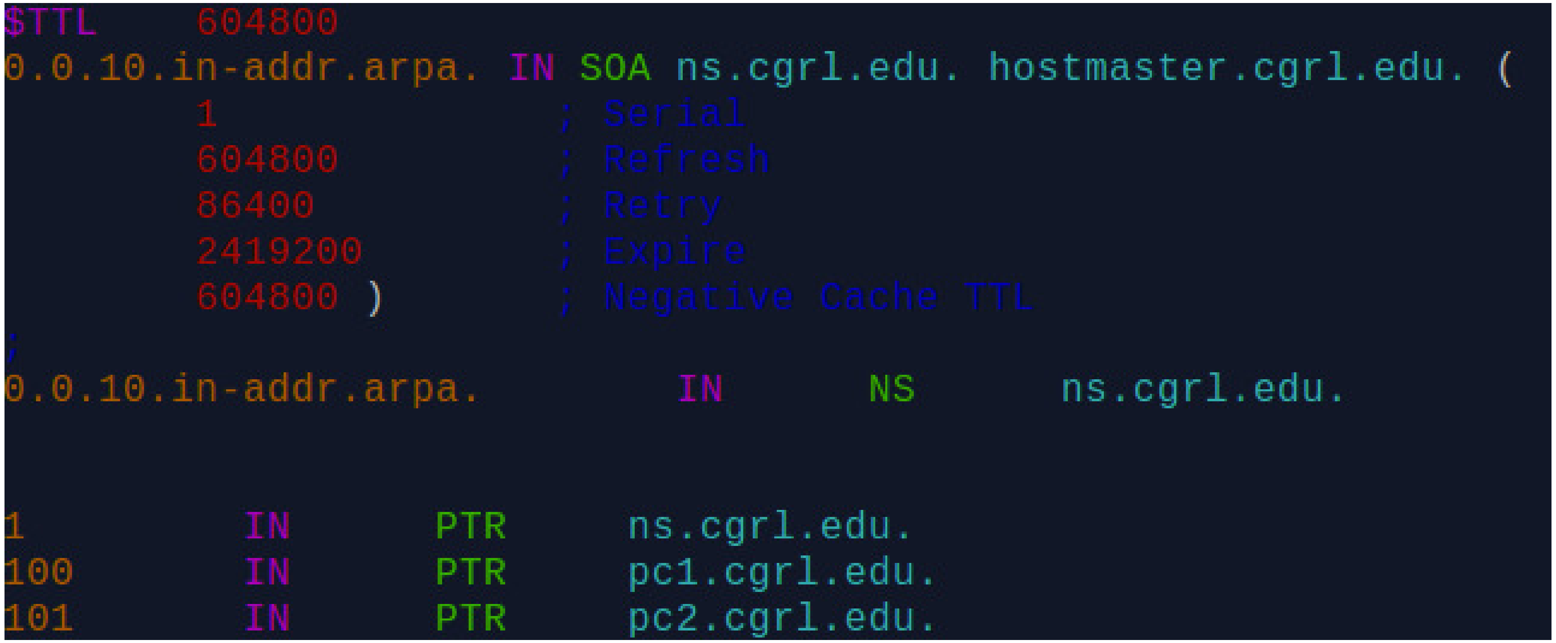
We make ns.cgrl.edu authoritative for the zone: 0.0.10.IN-ADDR.ARPA (10.0.0.0/24)

/etc/bind/named.conf



```
zone "0.0.10.in-addr.arpa" {
        type master;
        file "/etc/bind/db.0.0.10";
};
```

/etc/bind/db.0.0.10



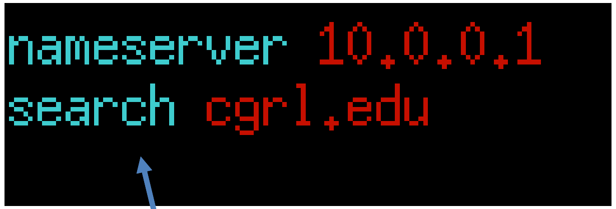
```
$TTL    604800
0.0.10.in-addr.arpa. IN SOA ns.cgrl.edu. hostmaster.cgrl.edu. (
        1               ; Serial
        604800          ; Refresh
        86400           ; Retry
        2419200         ; Expire
        604800 )        ; Negative Cache TTL
;
0.0.10.in-addr.arpa.        IN      NS      ns.cgrl.edu.


1          IN      PTR     ns.cgrl.edu.
100        IN      PTR     pc1.cgrl.edu.
101        IN      PTR     pc2.cgrl.edu.
```

BIND master file

- Master files are used as “db” for the configuration of bind
- Such files always have a domain associated with them, which is called the *origin* and specified as zone in bind.conf.
- Within a master file, you are allowed to specify domain and host names relative to this domain.
  - A name given in a configuration file is considered *absolute* if it ends in a single dot
  - it is considered relative to the origin. The origin all by itself may be referred to using ``@''.
- Wildcard * https://en.wikipedia.org/wiki/Wildcard_DNS_record

Resolver configuration

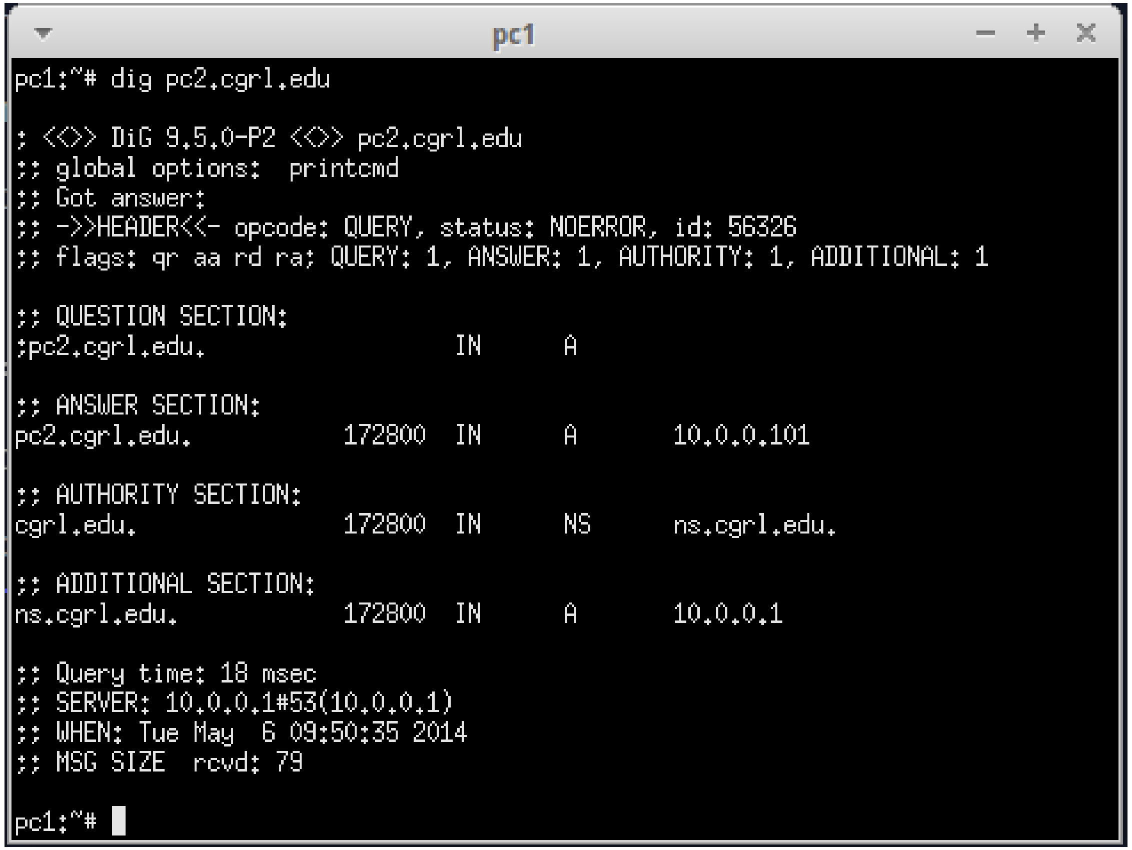
/etc/resolv.conf



```
nameserver 10.0.0.1
search cgrl.edu
```

If something gets to the stub resolver that has no dots in it, the resolver will try appending “search”. Multiple entries are possible, but not efficient.

Simpler approach is to put nothing or domain.



```
pc1:~# dig pc2.cgrl.edu

; <<>> DiG 9.5.0-P2 <<>> pc2.cgrl.edu
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56326
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;pc2.cgrl.edu.                  IN      A

;; ANSWER SECTION:
pc2.cgrl.edu.           172800  IN      A       10.0.0.101

;; AUTHORITY SECTION:
cgrl.edu.               172800  IN      NS      ns.cgrl.edu.

;; ADDITIONAL SECTION:
ns.cgrl.edu.            172800  IN      A       10.0.0.1

;; Query time: 18 msec
;; SERVER: 10.0.0.1#53(10.0.0.1)
;; WHEN: Tue May  6 09:50:35 2014
;; MSG SIZE  rcvd: 79

pc1:~#
```

/etc/resolv.conf

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

primary DNS

secondary DNS

search directive for short names

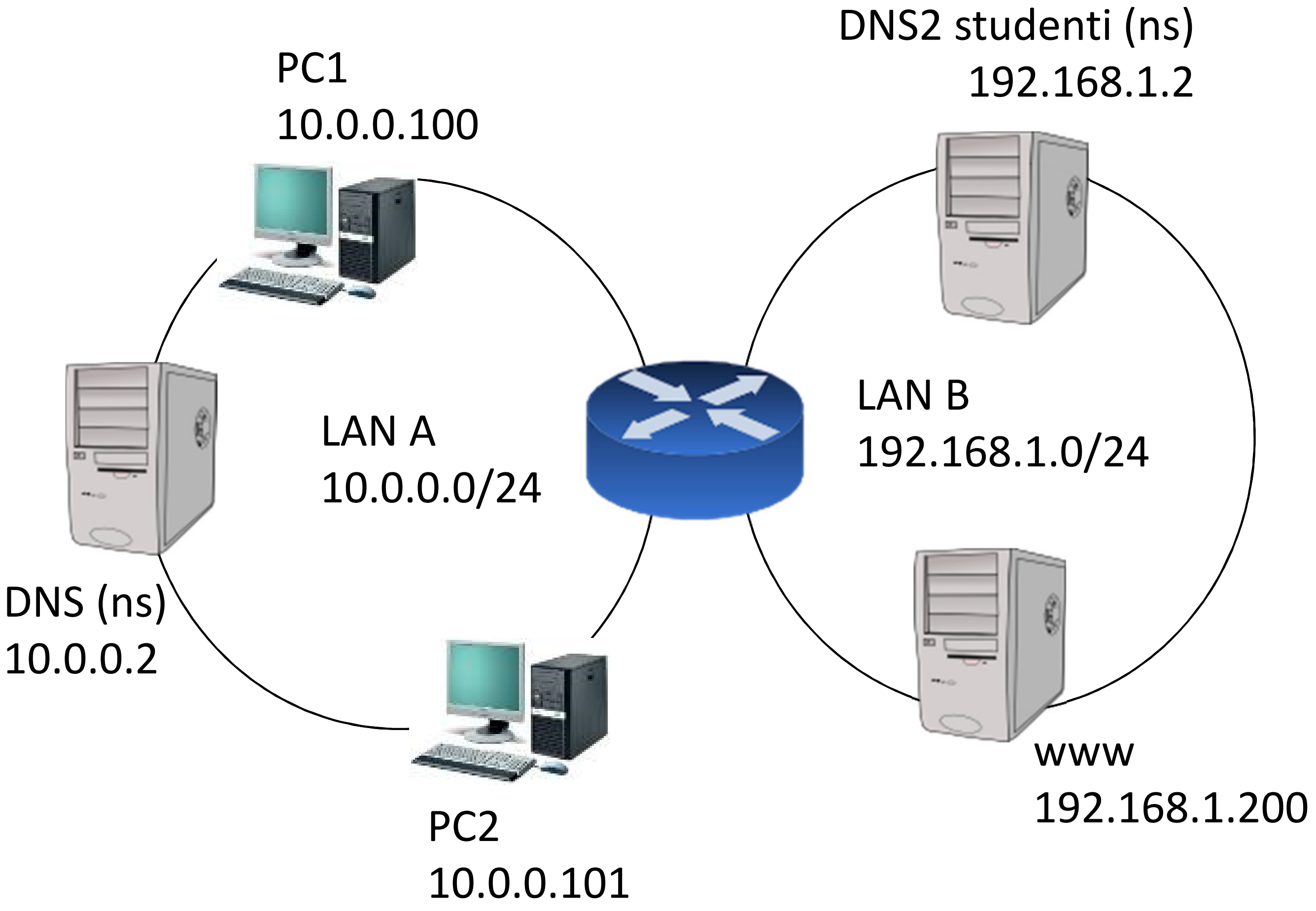
```
domain cgrl.edu
search A.net B.net
```

- Try to resolv “test”, on failure try to resolve test.cgrl.edu (using **gethostname** or *domain* if present)
- If you want that test will be resolved first as test.A,net and then as test.B,net specify search A.net B.net (in case test.A.net fails, resolver will go for test.B.net)
- The **domain** and **search** keywords are mutually exclusive. If using search, first entry is usually the local domain.
- If more than one instance of these keywords is present, the last instance wins.

Questions

- Can we ping just “alias”? Why?
- Is reverse DNS working?
- Can we also ping web.uniroma2.it (160.80.1.246) ?

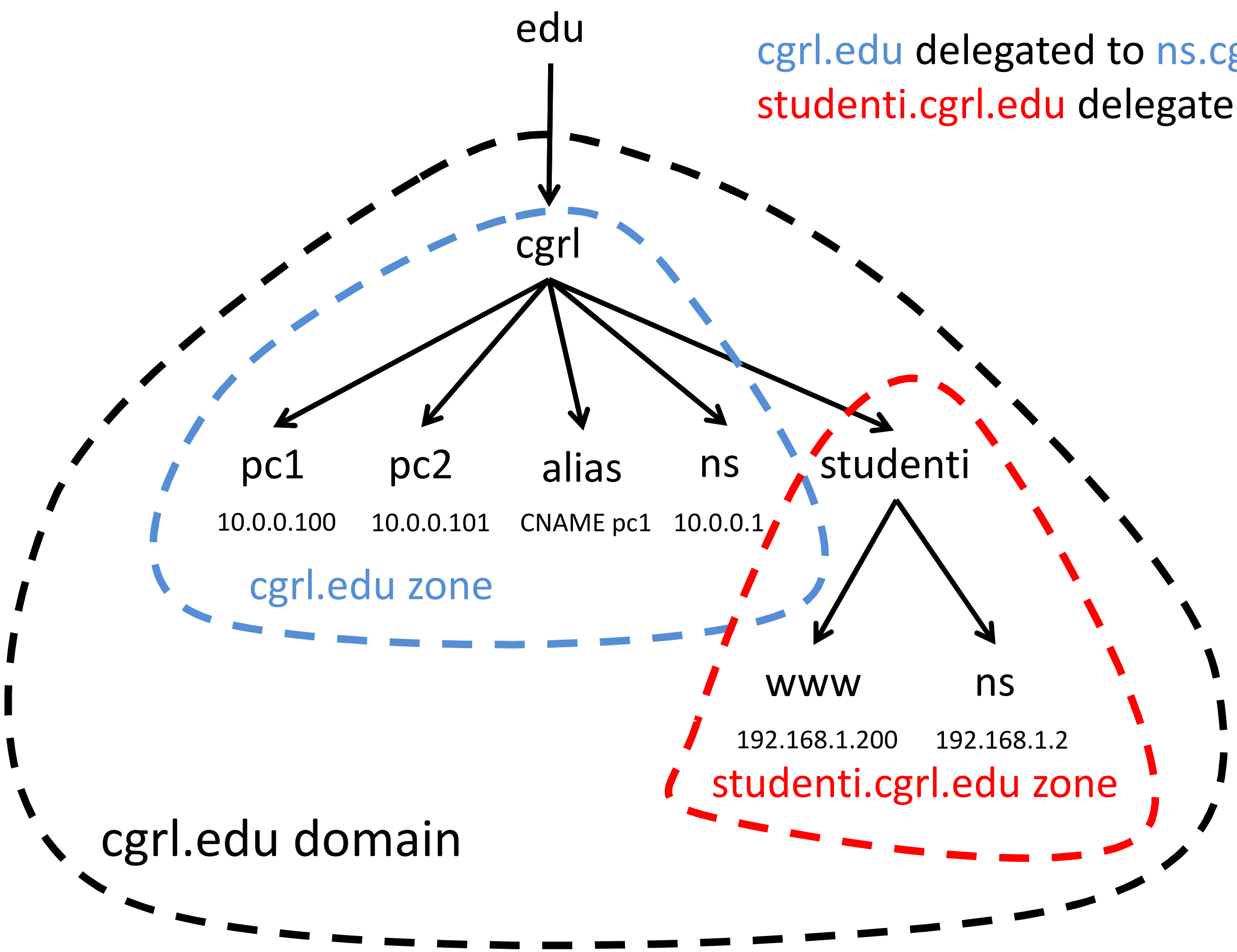
Second simple example: delegation of studenti.cgrl.edu



Second simple example: delegation of studenti.cgrl.edu

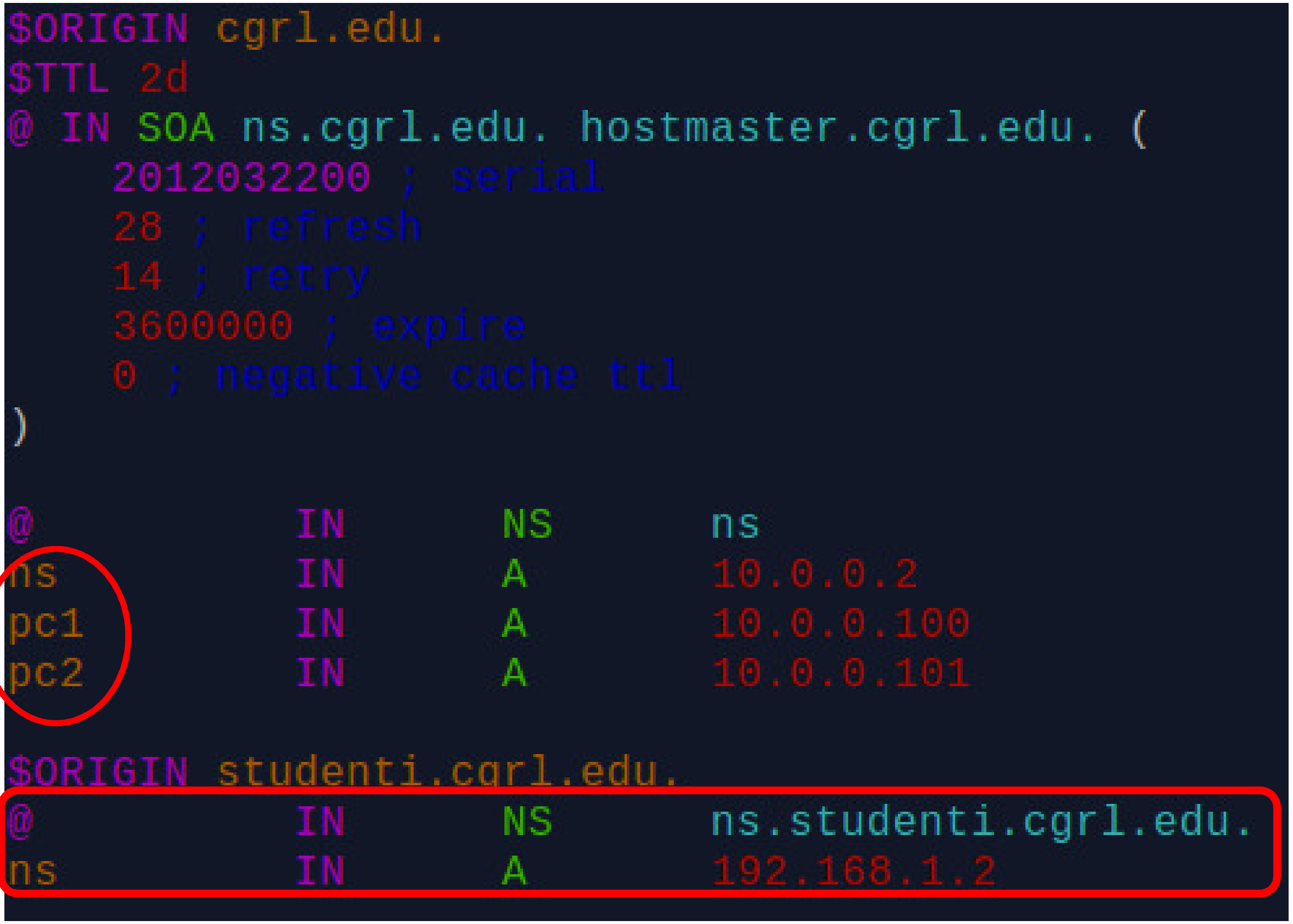
cgrl.edu delegated to ns.cgrl.edu

studenti.cgrl.edu delegated to ns.studenti.cgrl.edu



BIND configuration – dns

dns#/etc/bind/db.edu.cgrl



```
$ORIGIN cgrl.edu.
$TTL 2d
@ IN SOA ns.cgrl.edu. hostmaster.cgrl.edu. (
    2012032200 ; serial
    28 ; refresh
    14 ; retry
    3600000 ; expire
    0 ; negative cache ttl
)

@           IN      NS      ns
ns          IN      A       10.0.0.2
pc1         IN      A       10.0.0.100
pc2         IN      A       10.0.0.101

$ORIGIN studenti.cgrl.edu.
@           IN      NS      ns.studenti.cgrl.edu.
ns          IN      A       192.168.1.2
```

@ substitutes the current value of $ORIGIN

Relative names appended to current zone

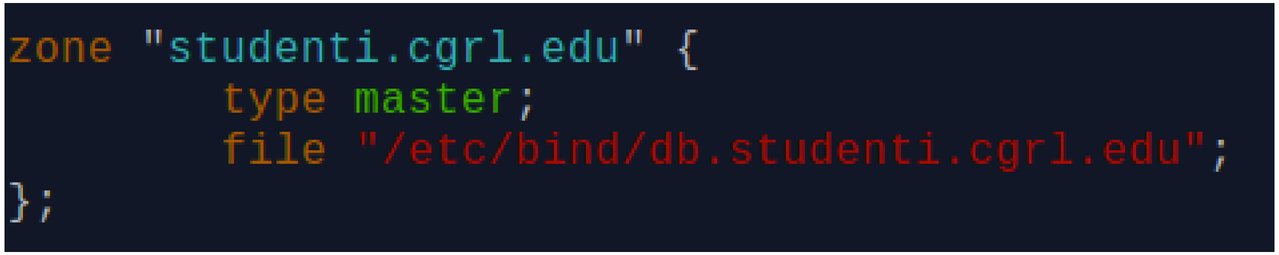
delegation

Glue record

- How we can resolve ns.studenti.cgrl.edu?
  - if that was exactly the dns responsible to resolve *.studenti.cgrl.edu!!
- A glue record is an A record for the name server that is authoritative for the delegated zone
  - ns.studenti.cgrl.edu   IN   A   192.168.1.2

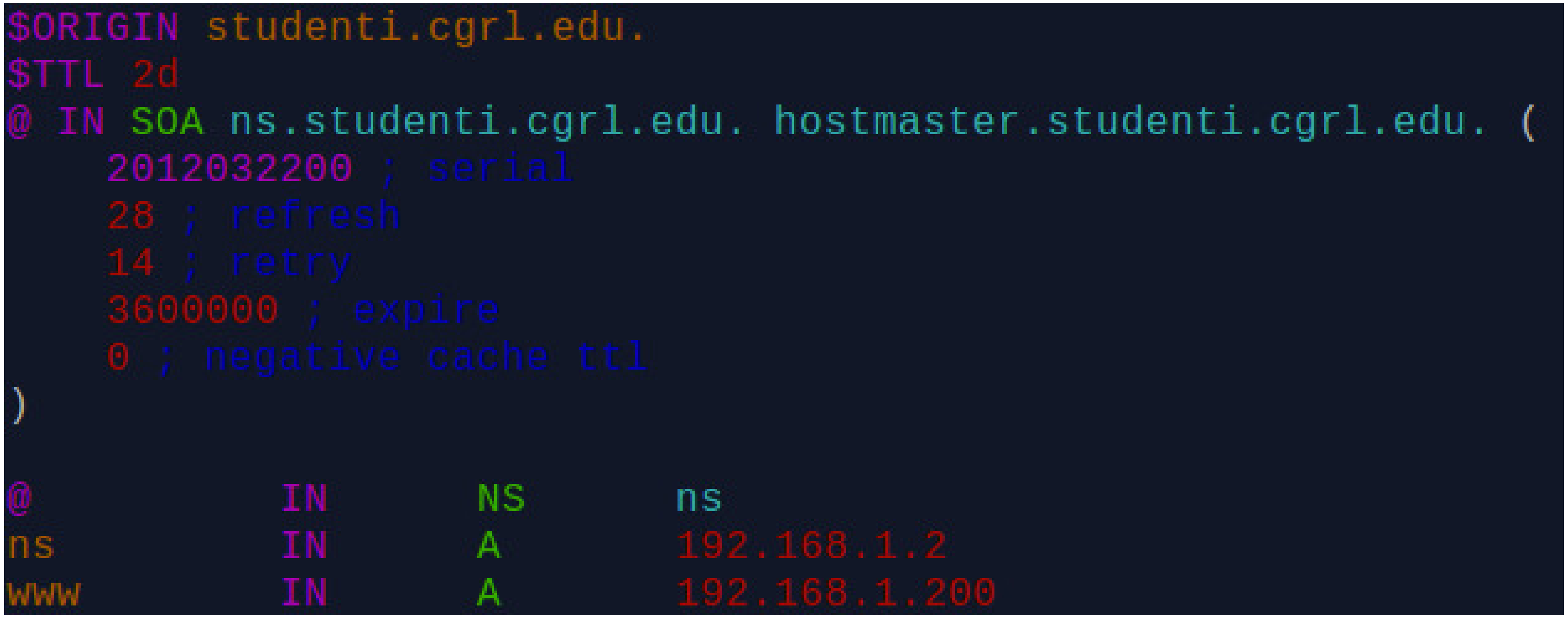
BIND configuration – dns2

Add to dns2#/etc/bind/named.conf



```
zone "studenti.cgrl.edu" {
        type master;
        file "/etc/bind/db.studenti.cgrl.edu";
};
```

dns2#/etc/bind/db.studenti.cgrl.edu



```
$ORIGIN studenti.cgrl.edu.
$TTL 2d
@ IN SOA ns.studenti.cgrl.edu. hostmaster.studenti.cgrl.edu. (
    2012032200 ; serial
    28 ; refresh
    14 ; retry
    3600000 ; expire
    0 ; negative cache ttl
)

@          IN      NS      ns
ns         IN      A       192.168.1.2
www        IN      A       192.168.1.200
```

Questions

- From PC1 we can now ping www.studenti.cgrl.edu on a different "domain"
- Do we get authoritative or non-auth DNS response?
  - Can we get an authoritative answer?
  - Why ping so slooow??

MX records and load Balancing

- **Mail eXchanger record** (**MX record**) – Each domain (and sub-domain) need to have an MX-record in order to receive email.
- DNS server will report all entries of its SNMP servers (request is performed by the sender, a Mail/message Transfer Agent (MTA)).
- A MTA relay, on multiple servers available will chose one → Round robin at "sender"

IN MX 10 mail.cgrl.edu.

IN MX 10 mail2.cgrl.edu.

IN MX 10 mail3.cgrl.edu.

…

mail IN A 192.168.0.4

mail2 IN A 192.168.0.5

mail3 IN A 192.168.0.6

SNMP servers

MX records returned

```
$> dig uniroma2.it mx

uniroma2.it.        3599      IN    MX  20 mx-01.uniroma2.it.
uniroma2.it.        3599      IN    MX  20 mx-02.uniroma2.it.
uniroma2.it.        3599      IN    MX  20 mx-05.uniroma2.it.
uniroma2.it.        3599      IN    MX  25 mx-03.uniroma2.it.
uniroma2.it.        3599      IN    MX  25 mx-04.uniroma2.it.
```

Exercise in class

Add two PCs for mail service.

Add an example MX entries in our DNS for cgrl.edu

Mail server failover

```
; zone file fragment
IN MX 10 mail.cgrl.edu.
IN MX 20 mail2.cgrl.edu.
IN MX 20 mail3.cgrl.edu.
....
mail   IN A    192.168.0.4 ....
mail2 IN A    192.168.0.5 ....
mail3 IN A    192.168.0.6 ....
```

- If the most preferred mail server, the one with the lowest number (10), is not available, mail must be sent to the second most preferred server (one with 20, randomly selected)

RR records and load Balancing

- DNS server support many records of the same type (ie., A, MX) for the same name→ Round robin by the server!

```
IN MX 10 mail.cgrl.edu.
…
mail    IN   A   192.168.0.4
             A   192.168.0.5
             A   192.168.0.6
```

Load Balancing

- The name server will deliver all the IP addresses defined for the given name in answer to a query for the RRs; works for MX, A, NS….NOT CNAME …. PTR?.
- The order of IP addresses in the returned list is variable, and defined by the rrset-order statement in BIND’s named.conf file.
- Example cofig: default is cyclic. It can be overridden as random for some types/names:
  - *rrset-order {type MX name "example.com" order random;*
    order cyclic;};
- Clients normally use only the first IP received!
- DNS local caching can significantly distort the effectiveness of any DNS IP address balancing algorithm. A TTL value of 0 may be used to inhibit

Sender Policy Framework (SPF)

- The design intent of the SPF record is to allow a receiving Message Transfer Agent (MTA) to verify that the originating IP (the source-ip) of an e-mail from a sender is authorized to send mail for the sender’s domain. RFC 7208
- Sender: foo@bar.it sIP=10.0.3.4. SNMP server: let’s ask DNS of “bar.it” if 10.0.2.4 belongs to its network.
- TXT RR (BIND releases from 9.4.0 support the SPF RR type)
- v=spf1 [pre] type [[pre] type] ... [mod]” where:
  - pre: + = pass (default), - = fail, ~ = softfail (indeterminate result), ? = neutral
  - type: This defines the mechanism type to use for verification of the sender.

SPF: SMTP Conversation Example

```
==> 220 teamits105.teamITS.net ESMTP Sendmail 8.13.6.20060614/8.13.6; Wed, 6 Dec 2007 14:27:47 -0600 (CST)
<-- HELO teamits104.teamITS.net
==> 250 teamits105.teamITS.net Hello py-in-f99.google.com [64.233.167.99], pleased to meet you
<-- mail from: sender@teamITS.com
==> 250 2.1.0 sender@teamITS.com... Sender ok
<-- rcpt to: steve@teamITS.com
==> 250 2.1.5 steve@teamITS.com... Recipient ok
<-- Data
==> 354 Please start mail input.
<-- From: sender@teamITS.com
<-- To: steve@teamITS.com
<-- Subject: Want to buy a widget?
<--
<-- Body text of message.
<-- .
==> 250 Mail queued for delivery.
<-- Quit
==> 221 Closing connection. Good bye.
```

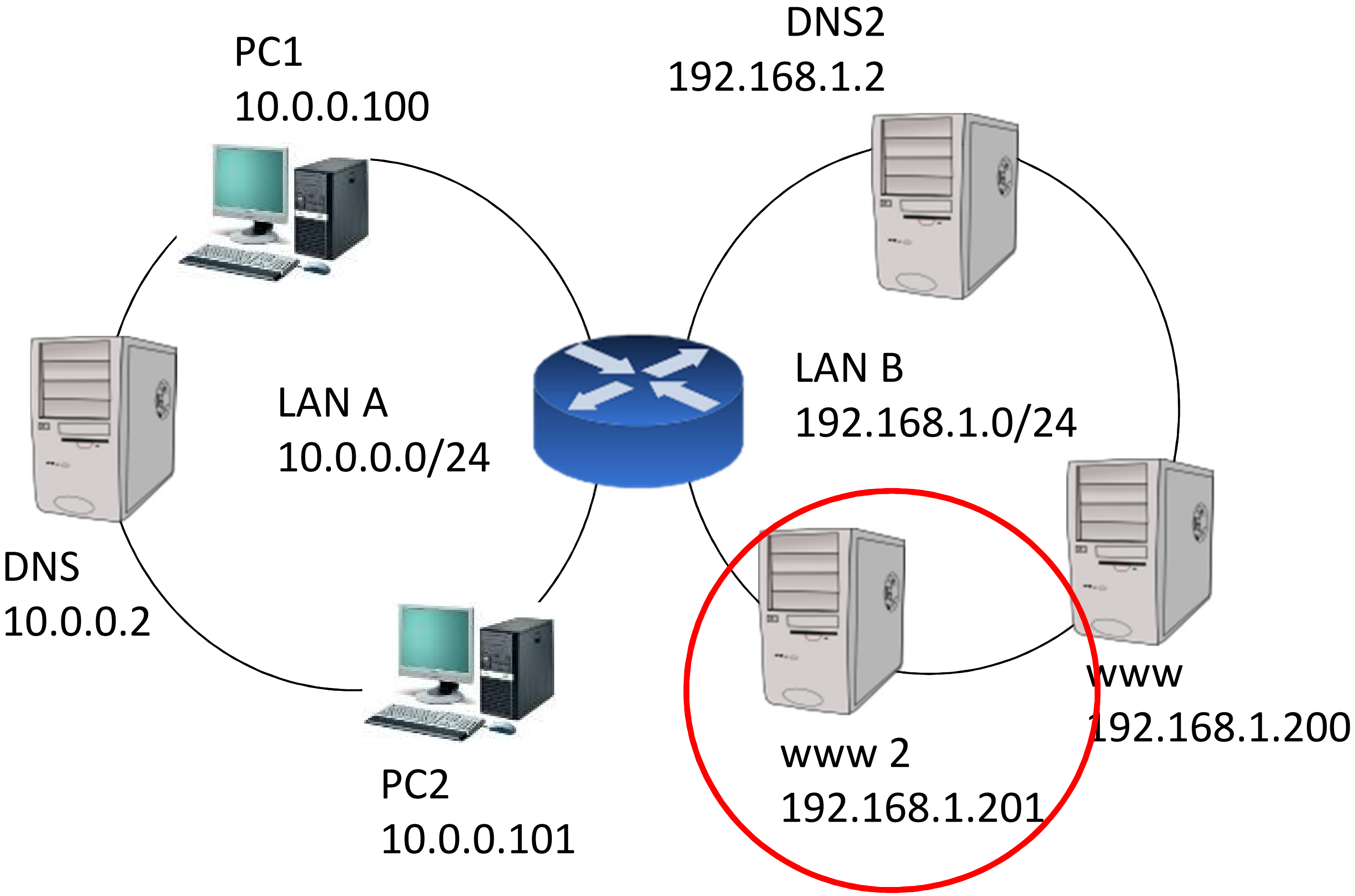
SPF Examples

- mail.acme.example.net. TXT "v=spf1 a –all”
  - The only host that can announce itself as mail.acme.example.net *is* mail.acme.example.net (indicated by the "a")

- @   IN TXT "v=spf1 a:mail.example.com/27 -all”
  - or: @   IN SPF "v=spf1 a:mail.example.com/27 –all
  - We can use slash notation to specify a CIDR range

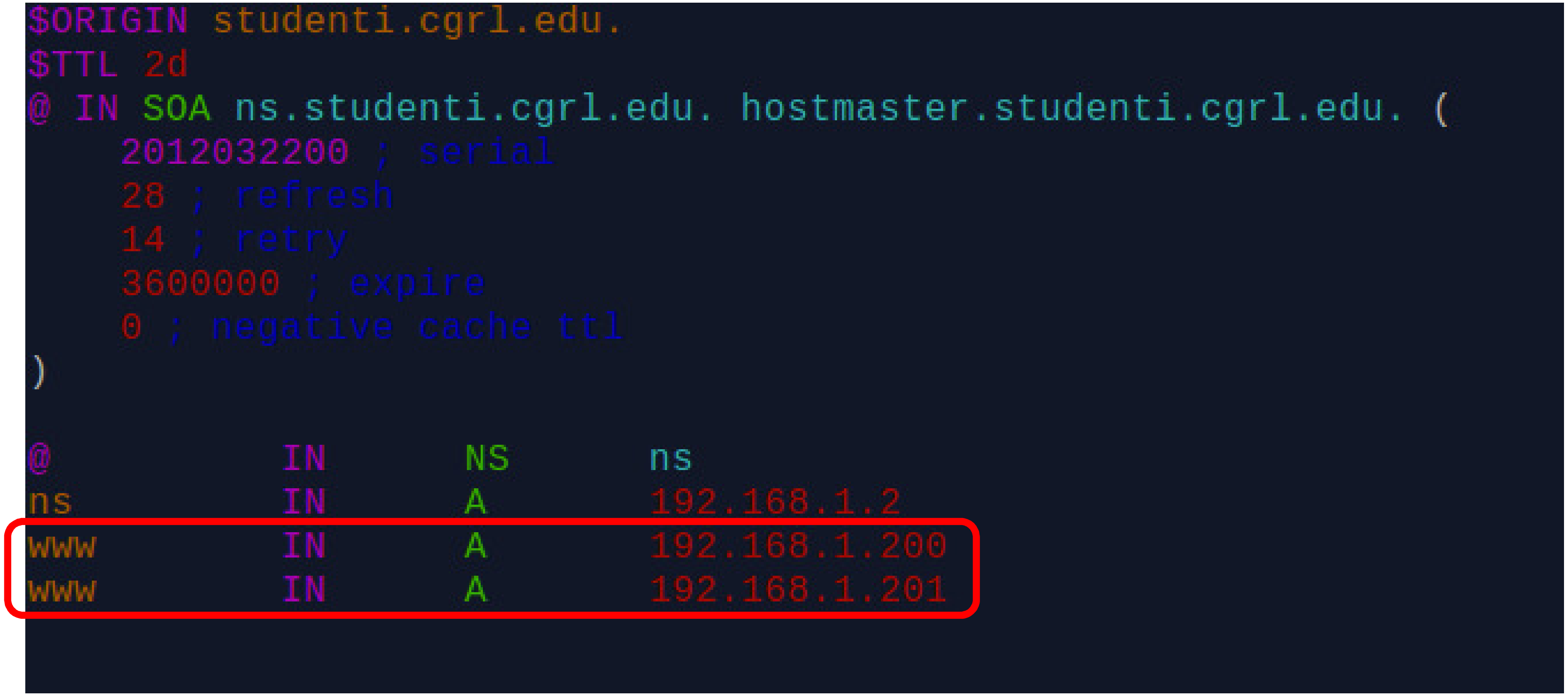
Exercise in class

Add www VM and load balance www.studenti.cgrl.edu between www and www “2”



Load Balancing of www server on lan B

- Simply add an other A RR in /etc/bind/db.studenti.cgrl.edu
- BIND will automatically round robin through the n addresses bound to the same name

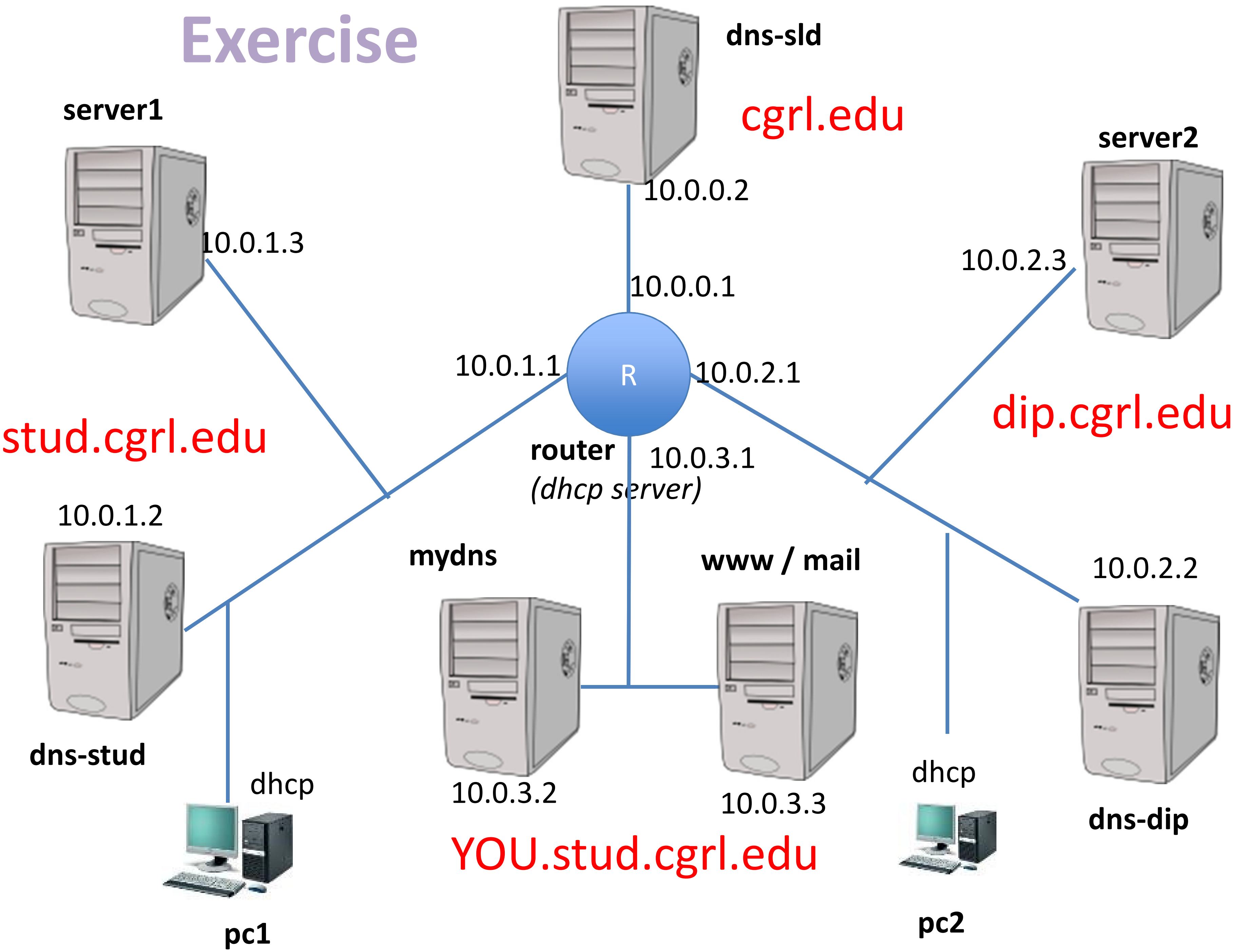


```
$ORIGIN studenti.cgrl.edu.
$TTL 2d
@ IN SOA ns.studenti.cgrl.edu. hostmaster.studenti.cgrl.edu. (
        2012032200 ; serial
        28 ; refresh
        14 ; retry
        3600000 ; expire
        0 ; negative cache ttl
)

@               IN      NS      ns
ns              IN      A       192.168.1.2
www             IN      A       192.168.1.200
www             IN      A       192.168.1.201
```

Question

- Test load balancing by ping. Does it work? Check also DNS reponses (dig/nsloockup).
- Why www can’t resolve, for example, pc1.cgrl.edu?
  - Solution?

Exercise


dns-sld
cgrl.edu
10.0.0.2
server1
10.0.1.3
server2
10.0.2.3
10.0.0.1
10.0.1.1
R
10.0.2.1
dip.cgrl.edu
stud.cgrl.edu
router
(dhcp server)
10.0.3.1
10.0.1.2
mydns
www / mail
10.0.2.2
dns-stud
dhcp
10.0.3.2
10.0.3.3
dhcp
dns-dip
YOU.stud.cgrl.edu
pc1
pc2