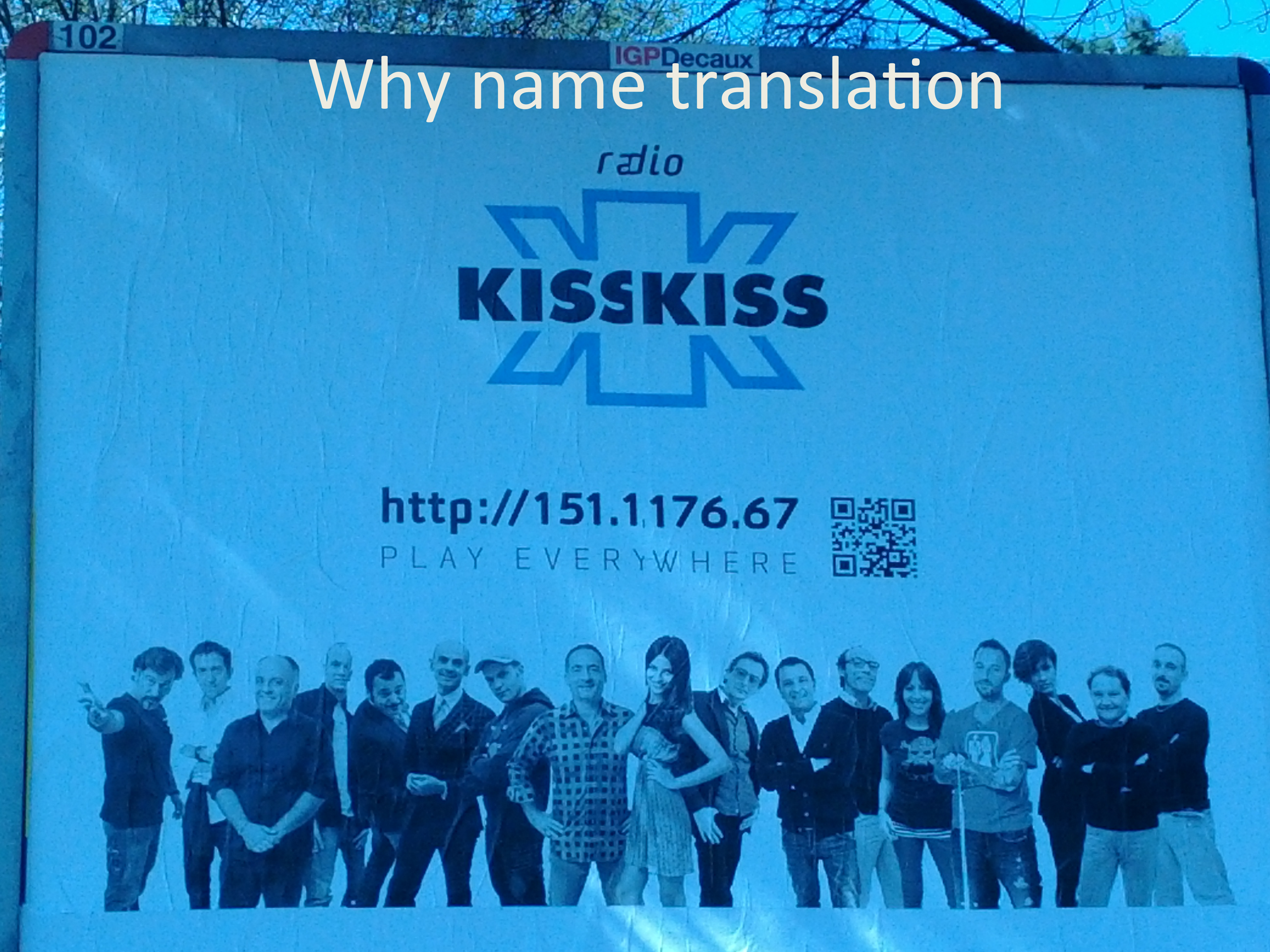
DNS & BIND

Lorenzo Bracciale

Marco Bonola

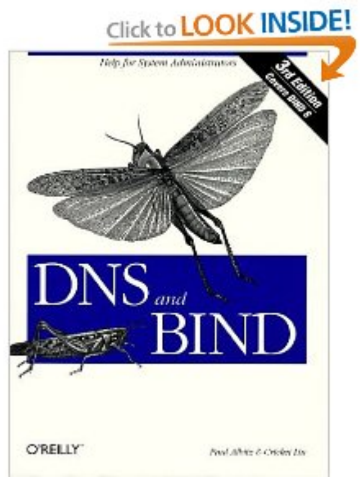
Why name translation



Need for name translation

- initially because tty2 is better than port 21
- …imagine IPV6!
  - 2002:a050:6768:0:e2f8:47ff:fe38:c5cc: (my pc)
- 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)

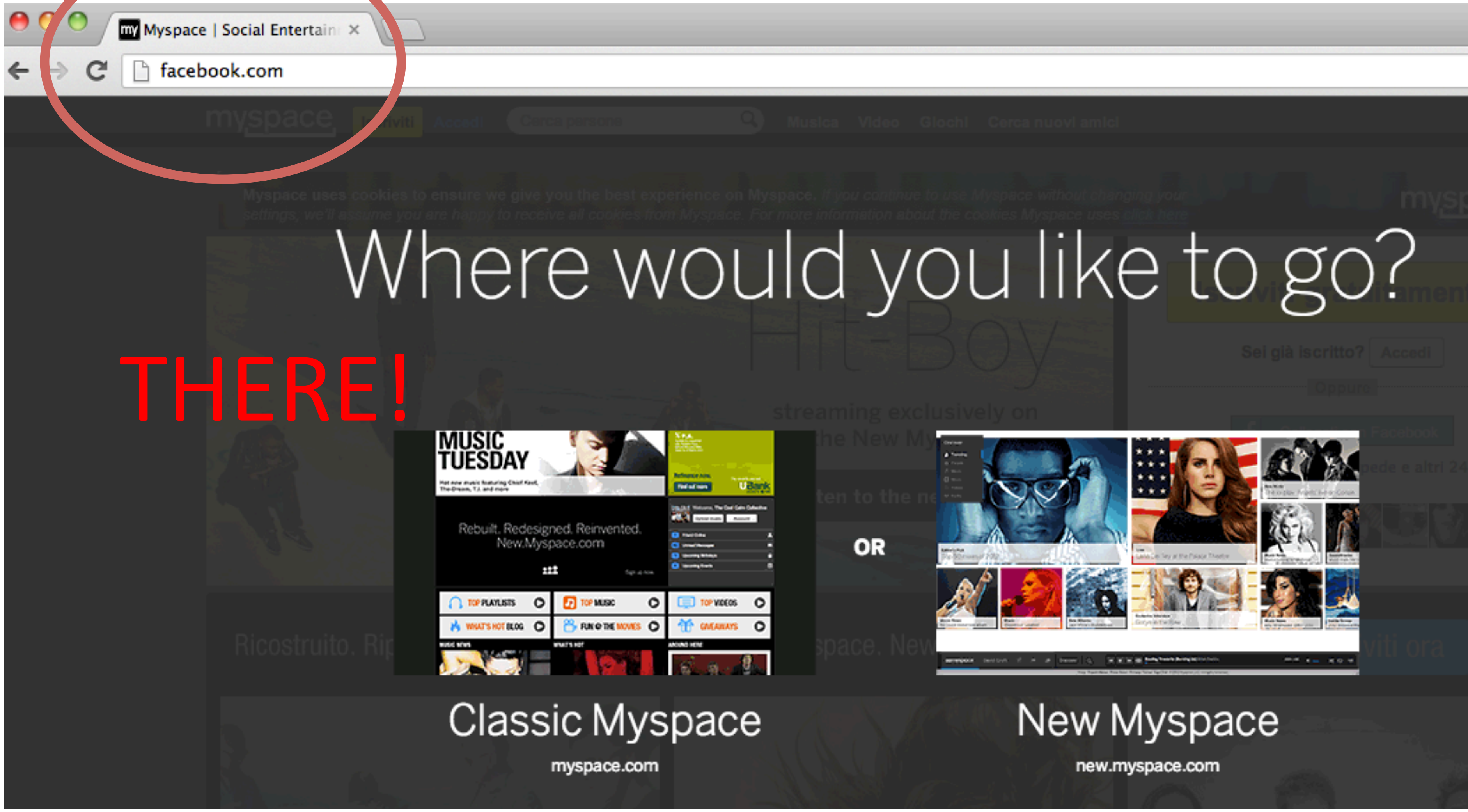


Before DNS…

- Each computer has HOSTS.txt
  - still used in all operating system, check your one!

```
127.0.0.1 localhost
```

- Try to put in /etc/hosts:
  - 63.135.91.11 facebook.com
- Inefficiencies: traffic load, name collisions, consistencies



Myspace | Social Entertain

facebook.com

Myspace uses cookies to ensure we give you the best experience on Myspace. If you continue to use Myspace without changing your settings, we'll assume you are happy to receive all cookies from Myspace. For more information about the cookies Myspace uses click here

Where would you like to go?

THERE!

OR

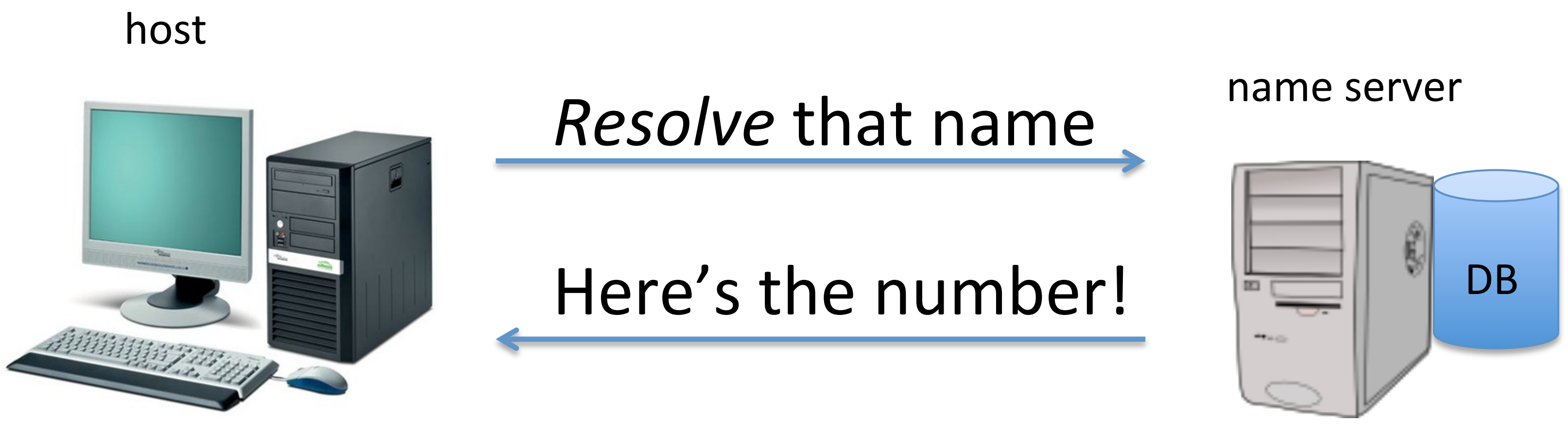
Classic Myspace

myspace.com

New Myspace

new.myspace.com

Simple solution



On Internet

- need of a *scalable* solution (today > ~284M domains[1])
- avoid name collision
- reliability
- introduce hierarchical names*:* *www.example.com.*
- Key concept: *<u>authority</u>* *and* *<u>delegation</u>*

"silent dot"

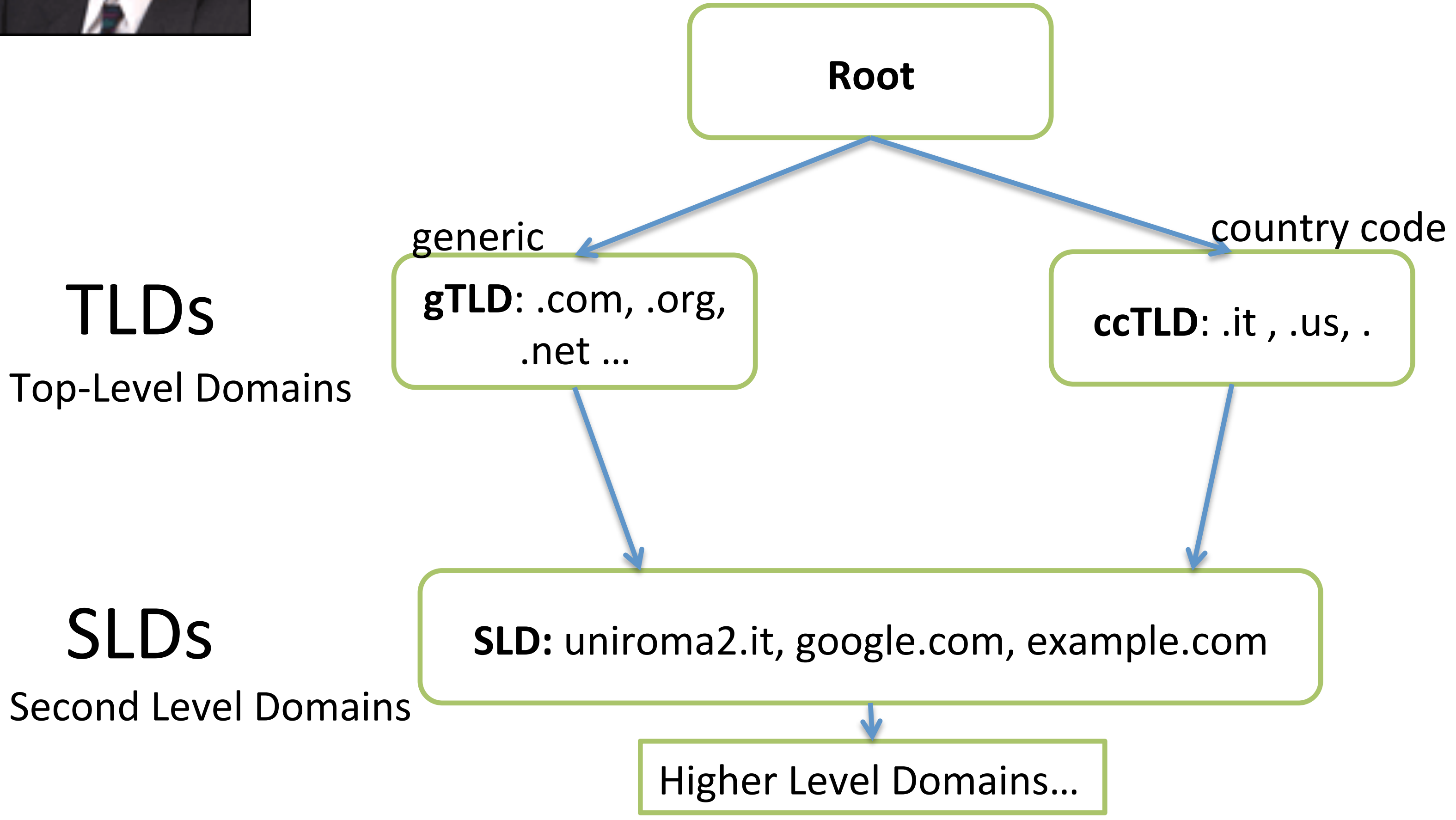
[1] https://investor.verisign.com/releasedetail.cfm?releaseid=892548

Internet Domain Name System

- DNS's distributed database is indexed by domain names
- 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*
- 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 (<u>easier collision avoidance</u>)
- Scalability is reached through <u>DELEGATION</u>

First experiment by Paul Mockapetris 1983

Internet Domain Name System



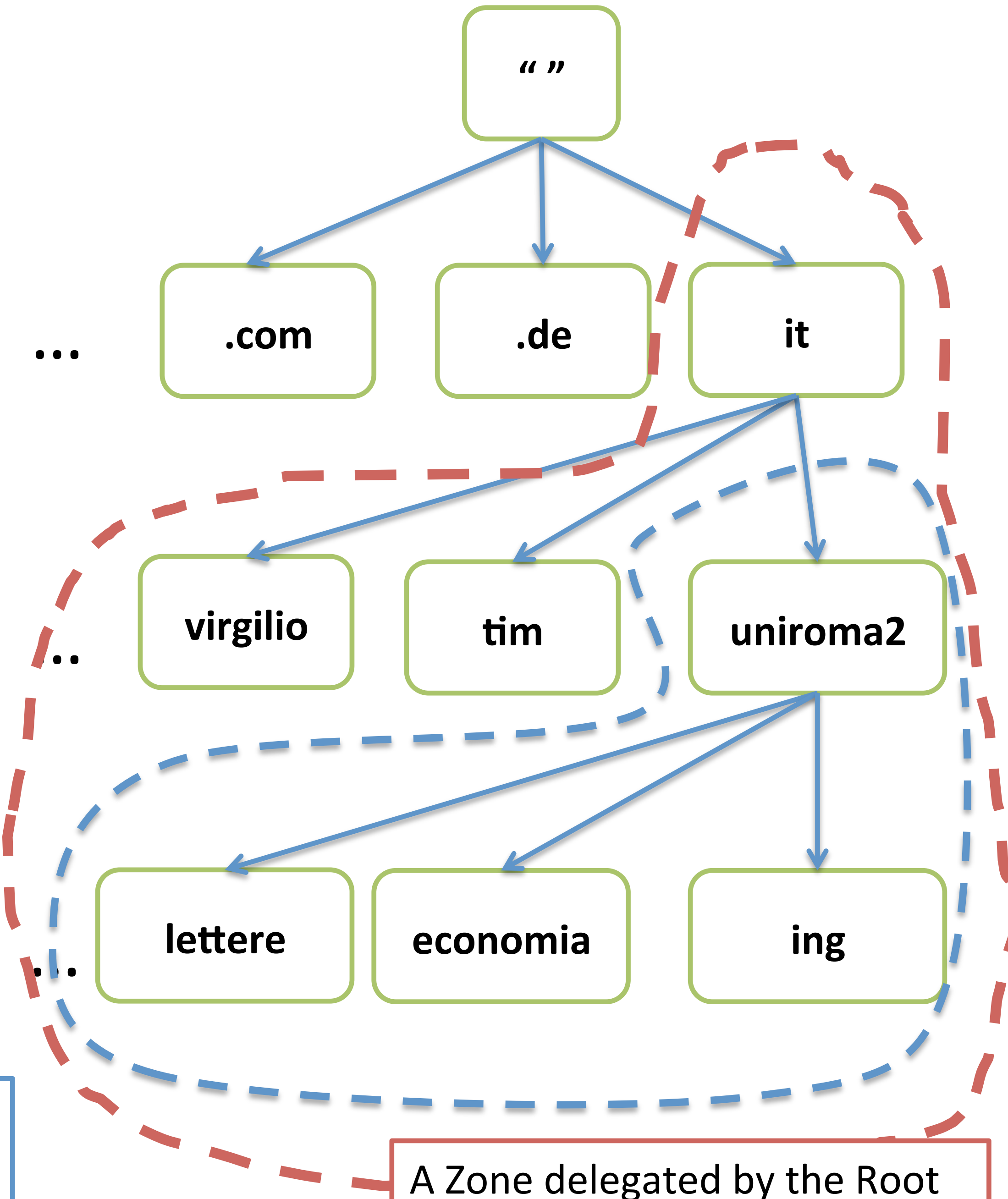
A **Domain** is a string representing the realm of an **Authority**

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

Registrar, Registry, Maintainer

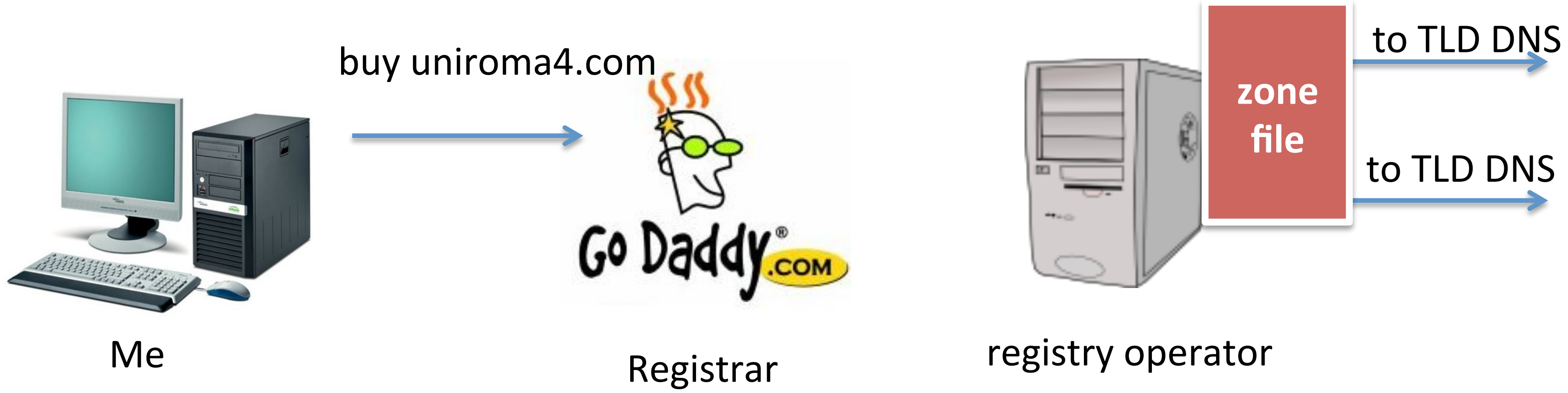
- **Registry**: database of all domain names registered in a top-level domain or second-level domain extension
- **Registrar**: frontend to the public
  - accredited by a gTLD or ccTLD:
    - Example http://www.nic.it/cgi-bin/List/index.cgi
  - Works with “web pages” (*asynchronous*)
- **Maintainer**: 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
Created:          1997-12-03 00:00:00
Last Update:      2013-03-08 12:19:02
Expire Date:      2014-01-14
Registrant
  Name:           Universita' degli Studi di Roma "Tor Vergata"
  Organization:   Universita' degli Studi di Roma "Tor Vergata"
  ContactID:      UNIV86
(  …. )
Admin Contact
(…)
Technical Contacts
 (…)
Registrar
  Organization:   Universita' degli Studi di Roma "Tor Vergata"
  Name:           UNIROMA2-REG
Nameservers
  dns.uniroma2.it
  dns1.uniroma2.it
  ns1.garr.net
```

Updating names: let’s buy a “domain”



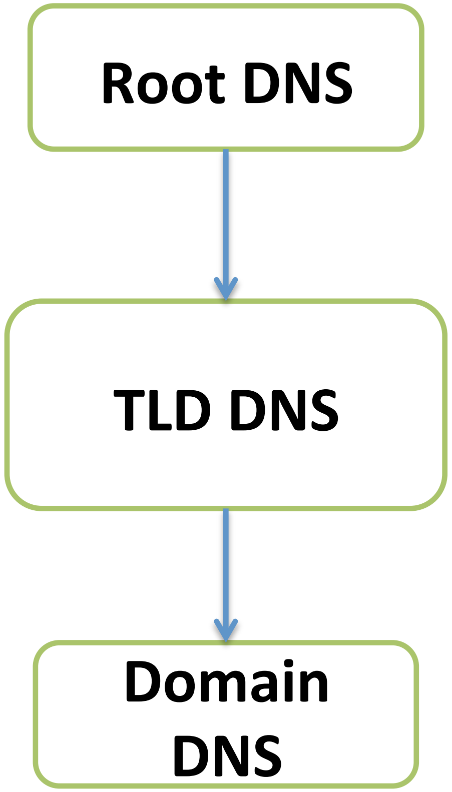
- A registrar interacts with public, store detailed information, and pass a “digest” to registry operator.
- Registry operator build a “zone file” (i.e. Data describing the domain) and pass it to interested TLD
- Periodically, ICANN distribute a “TLD master file” to each Root Server.

www.example.com

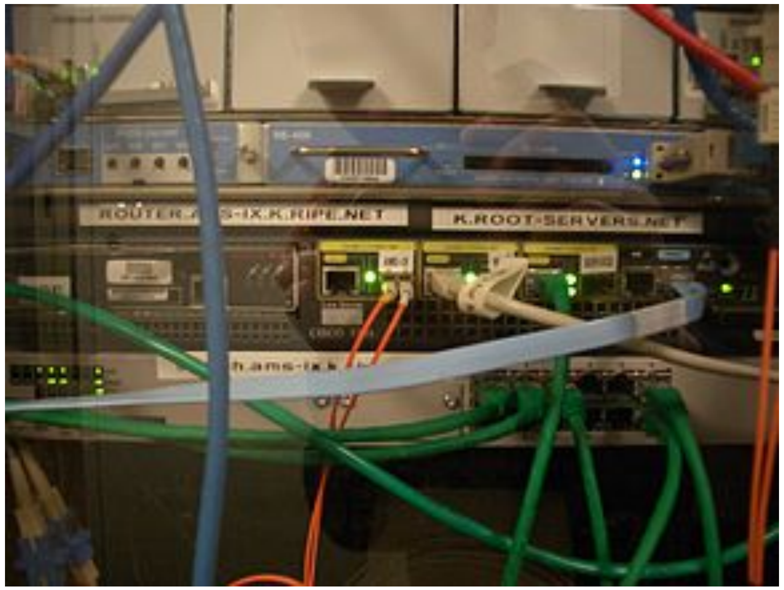
- The domain name example.com was delegated from a **gTLD authority**, which in turn was delegated from **ICANN** (authority for DNS Root Zone)
- The owner of the domain chooses the www part (called host name)
- This is a Fully Qualified Domain Name (**FQDN**)
  - specifies an exact location in the DNS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

DNS Implementation

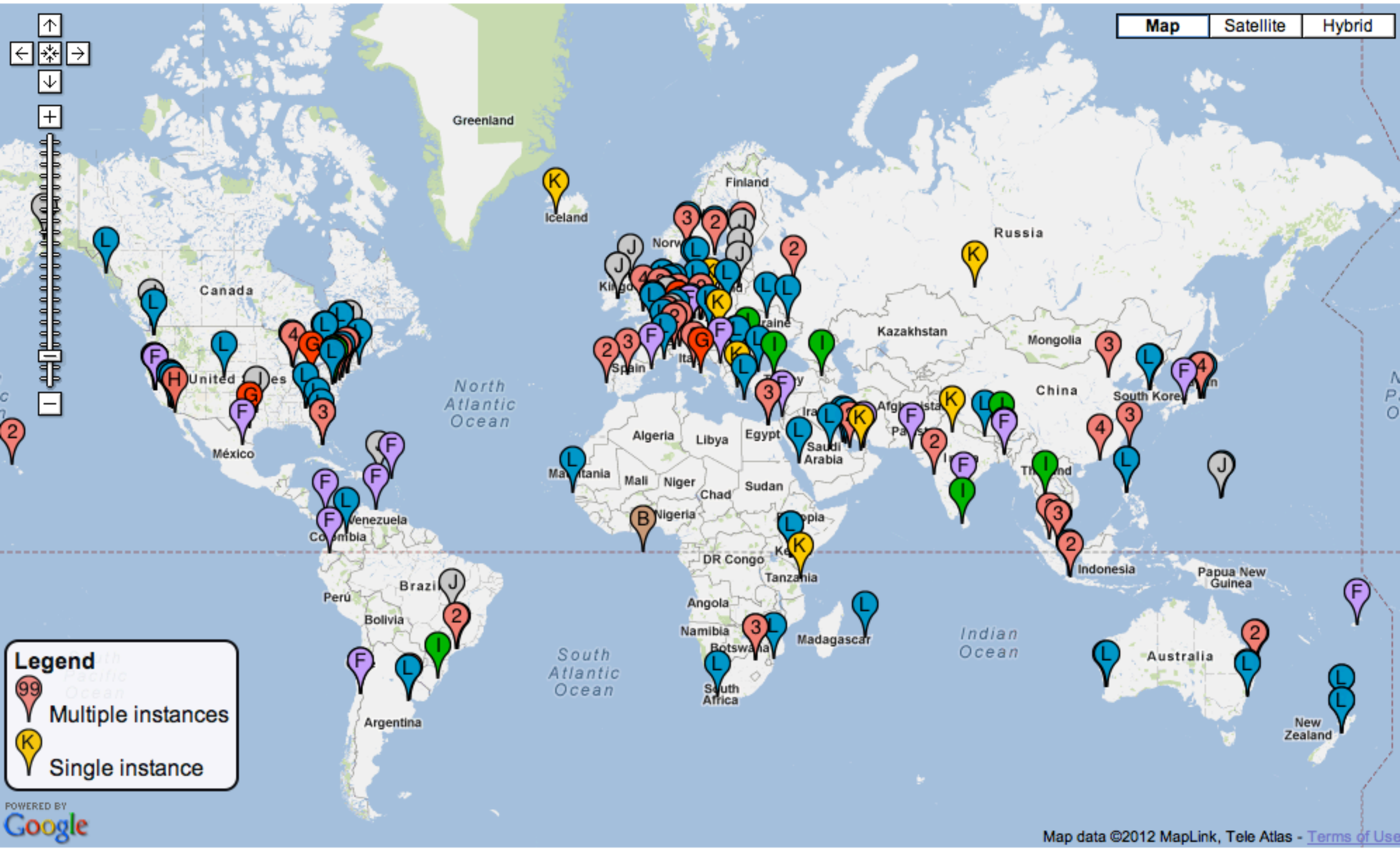
- Exactly maps the domain name delegation structure



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



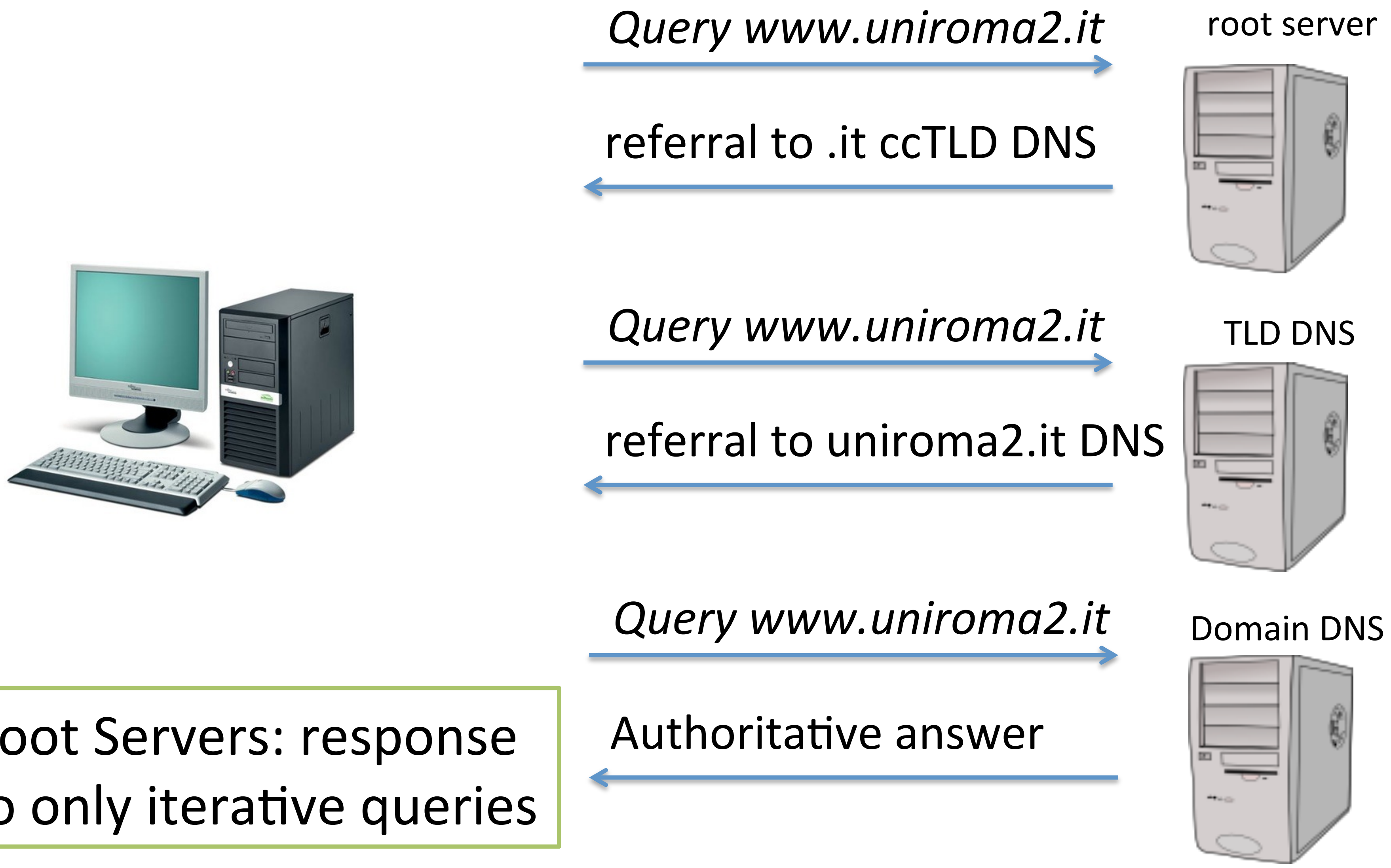
Root servers (anycast)



A DNS comprehends:

1. Zone files
   - translates the domain names into operational entities, such as hosts, mail servers, services for use by DNS software.
   - standard with Resource Records (RFC 1035, so portable!)
2. DNS program
3. Resolver library (ask the questions)

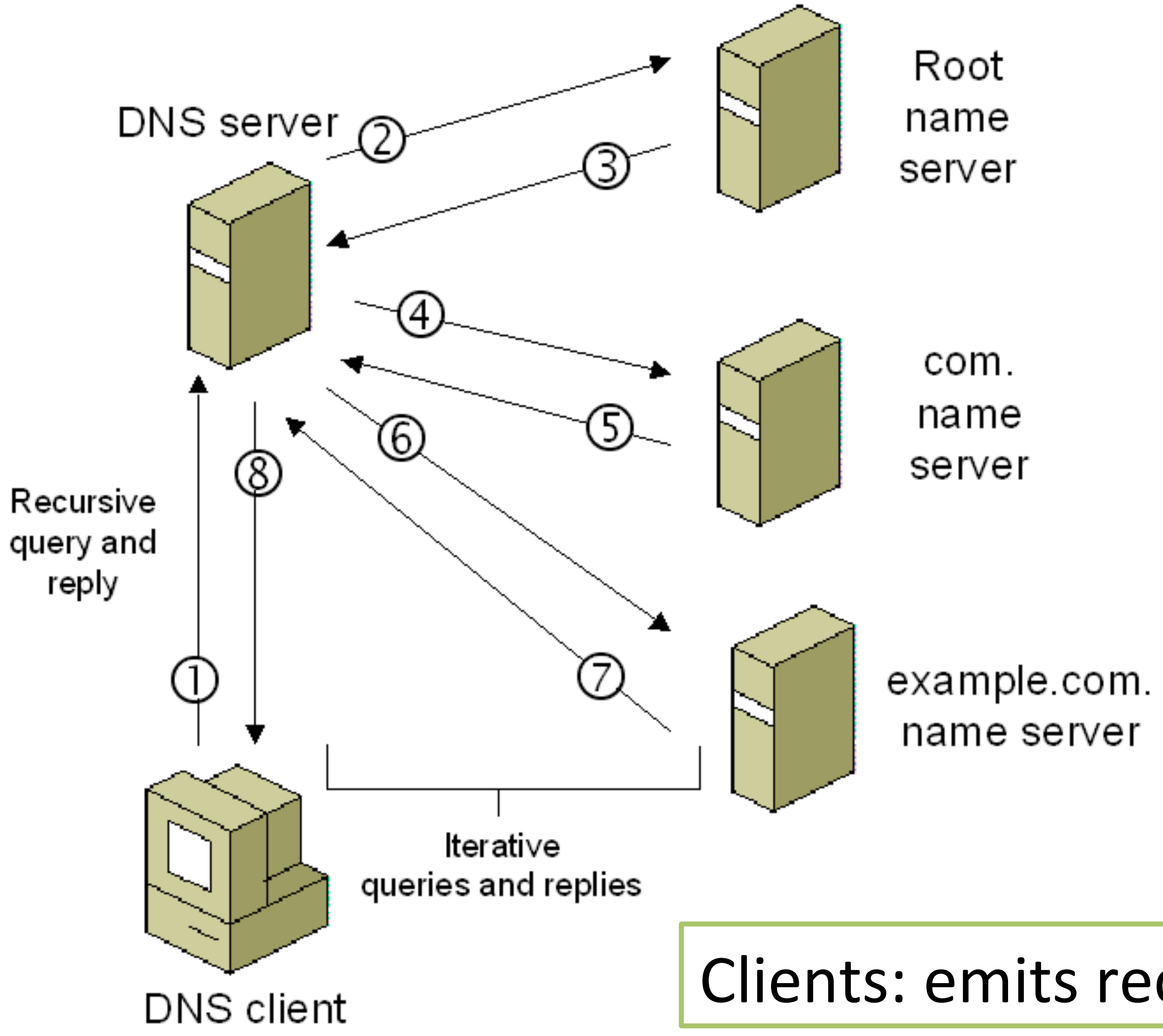
DNS Queries: iterative vs recursive

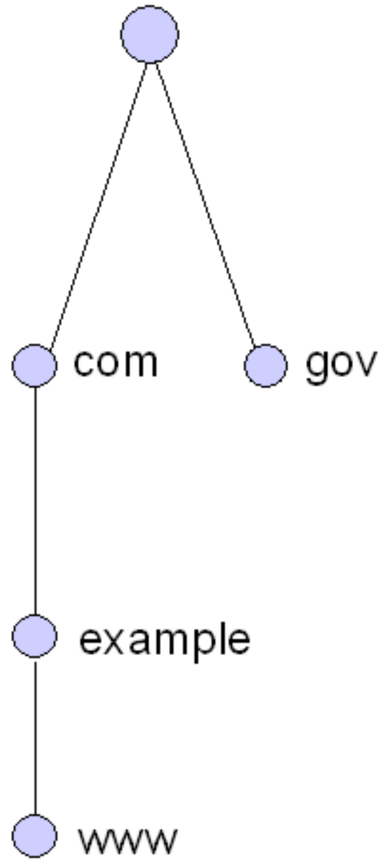


Root Servers: response to only iterative queries

DNS Queries: iterative vs recursive

i.e. find an answer

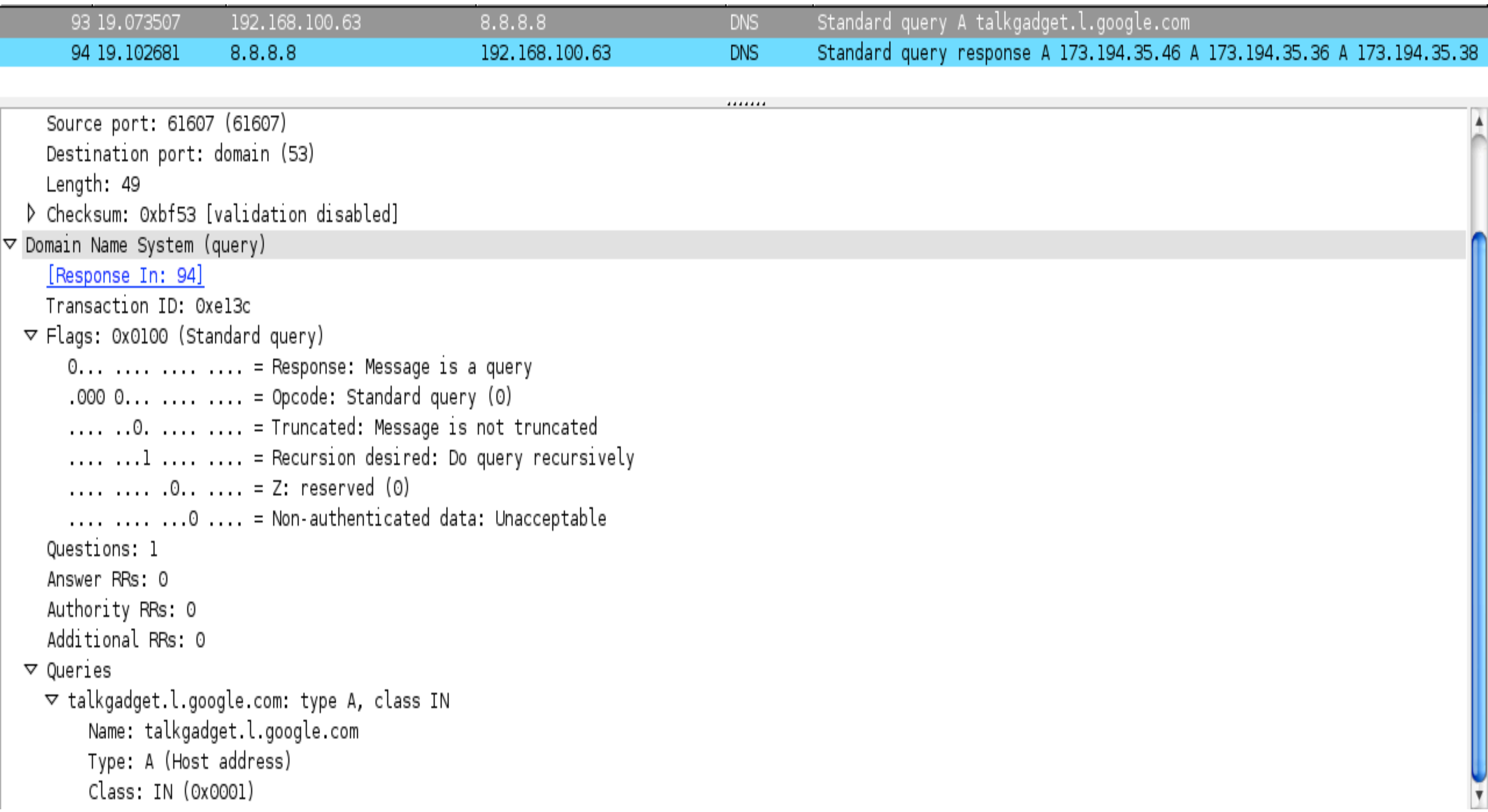




Clients: emits recursive queries

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

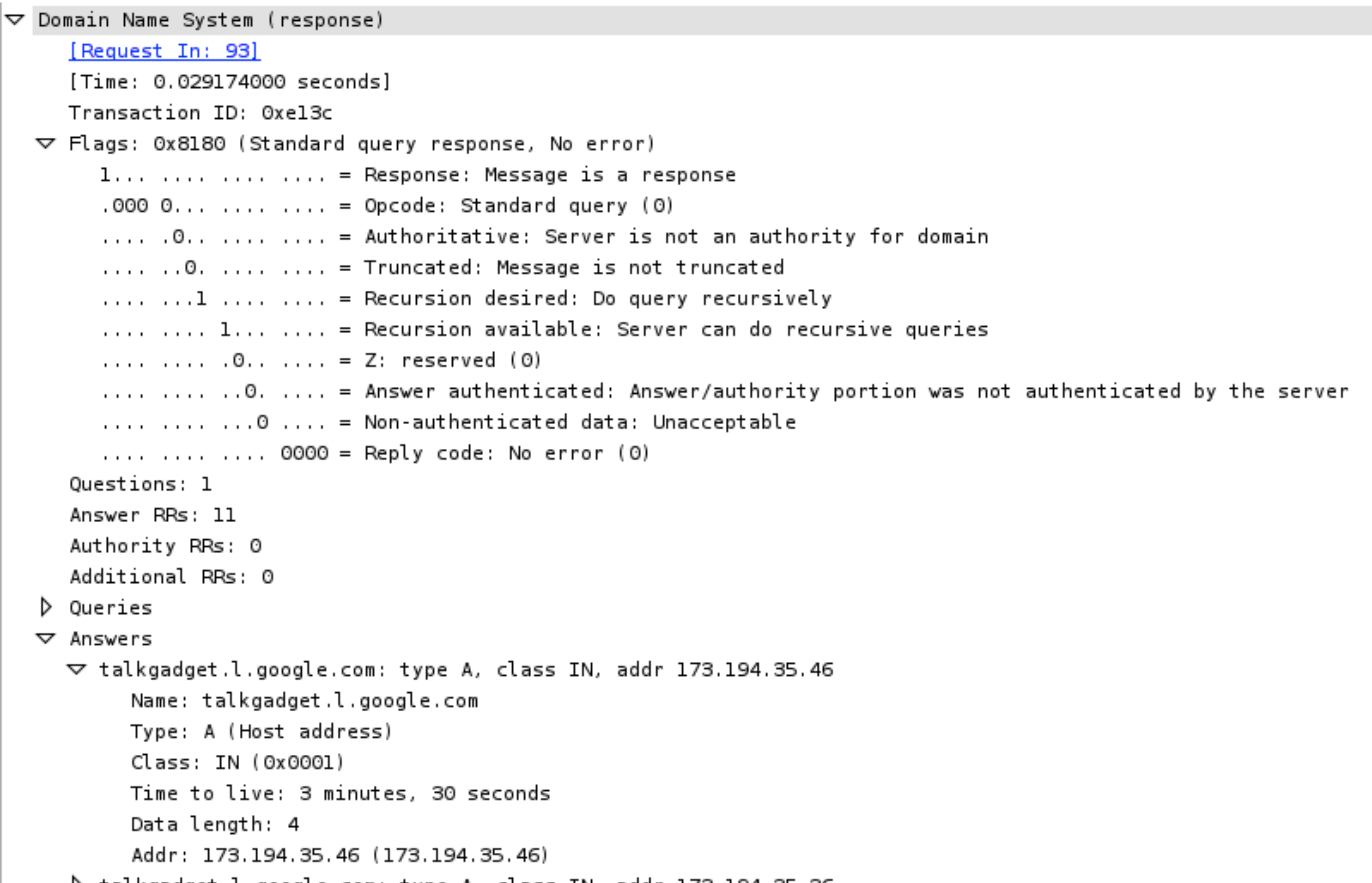
DNS Queries



```
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)
    ▷ talkgadget.l.google.com: type A, class IN, addr 173.194.35.36
```

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 DNS
- Browser use *gethostbyname* or *gethostbyaddr* methods to invoke name/ip resolution
  - functions provided by the stub resolver

debian package: *dnsutils*

Dig

```
root@ale:~# dig www.uniroma2.it

; <<>> DiG 9.7.3 <<>> www.uniroma2.it
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31347
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 0

;; QUESTION SECTION:
;www.uniroma2.it.        IN    A

;; ANSWER SECTION:
www.uniroma2.it.   3600       IN    CNAME    webhouse01.ccd.uniroma2.it.
webhouse01.ccd.uniroma2.it. 3600 IN  A       160.80.2.46

;; AUTHORITY SECTION:
ccd.uniroma2.it.     3600       IN    NS    dns1.uniroma2.it.
ccd.uniroma2.it.     3600       IN    NS    dns.uniroma2.it.

;; Query time: 53 msec
;; SERVER: 213.133.99.99#53(213.133.99.99)
;; WHEN: Thu Mar 22 18:35:15 2012
;; MSG SIZE  rcvd: 115
```

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 +short
  - show in short form all the ns fields of root servers
- dig -x 204.152.184.167 +short
  - reverse lookup

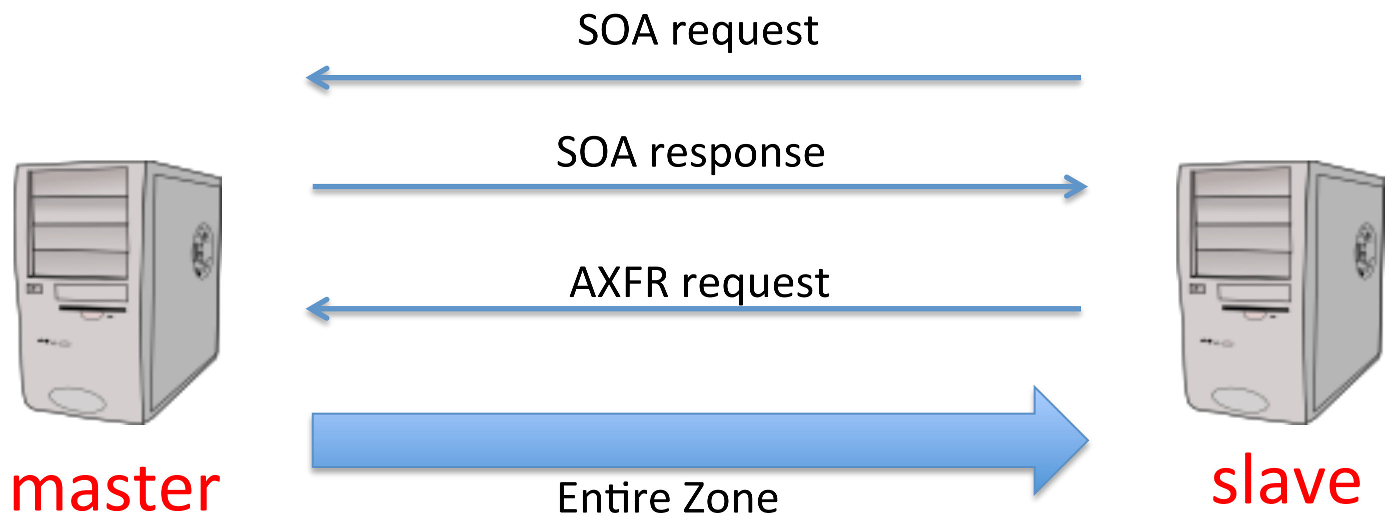
tcpdump for dns

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

Master Slave configuration



- redundancy for load balancing and fault resilience
- zones are passed from master to slave
  - full or partial zone transfer
- timing?

RFC 1035

Zone File: Example

Comments

directives

```
$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
```

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, AAAA, 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

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
- Invert the IP and search in the IN-ADDR.ARPA domain

Reverse Mapping: zone file

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

PTR RR

192.168.254.17

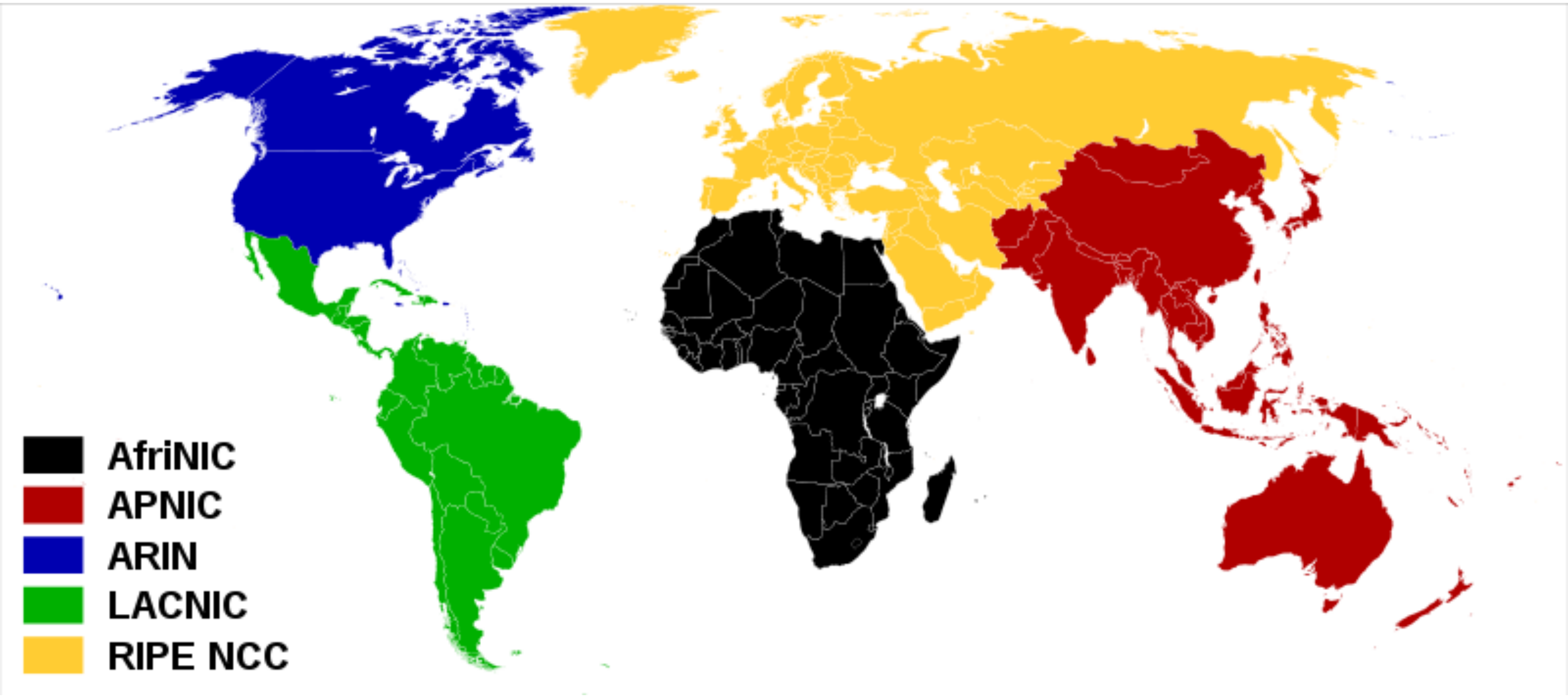
Try with:
dig -x 204.152.184.167 +short

Reverse Mapping

- 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

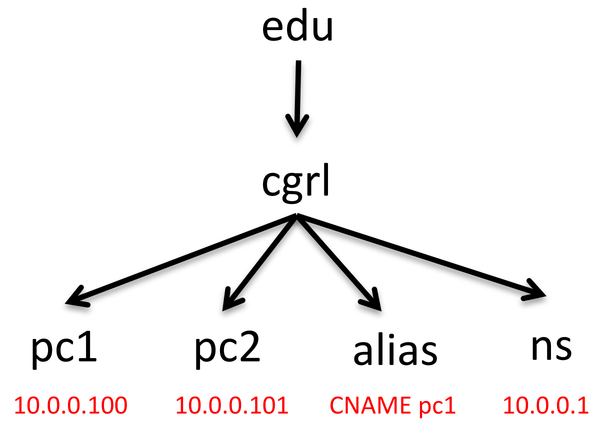
Interested? Search for **Internet Governance**
http://en.wikipedia.org/wiki/Internet_governance

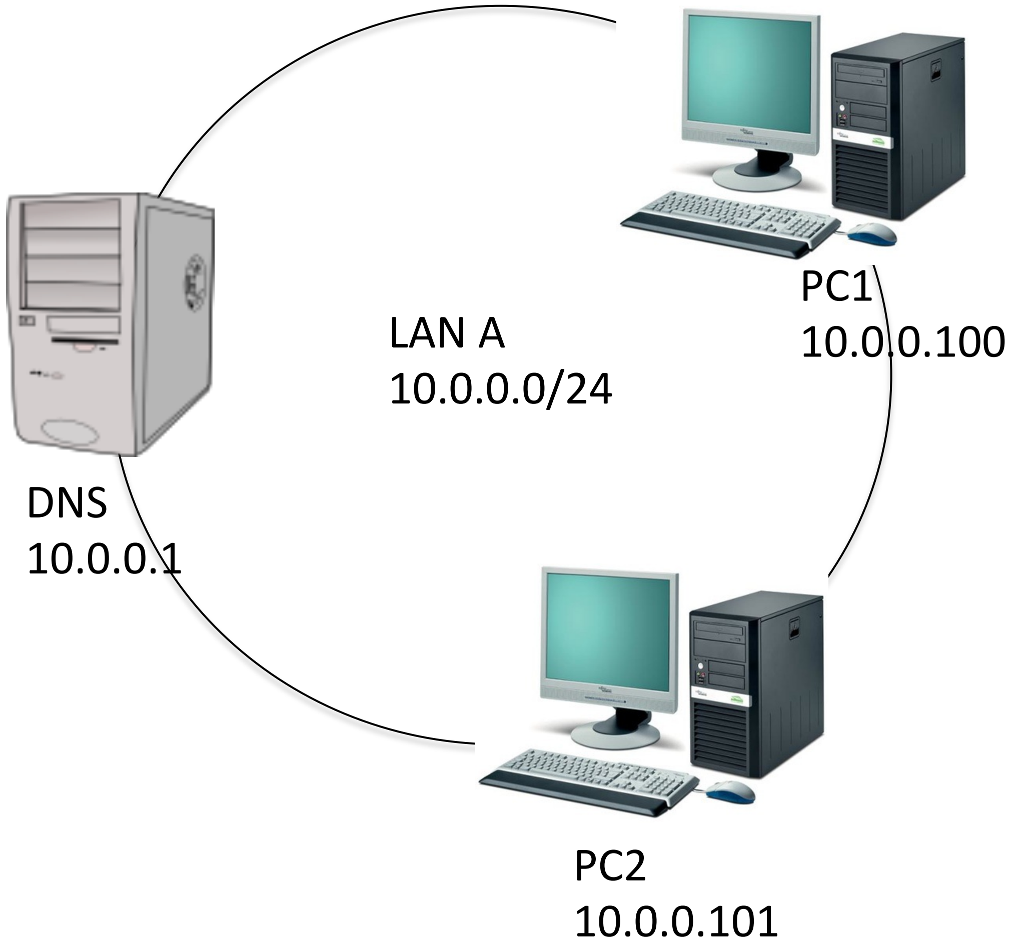
Things are getting serious!

BIND

First simple example: cgrl.edu

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

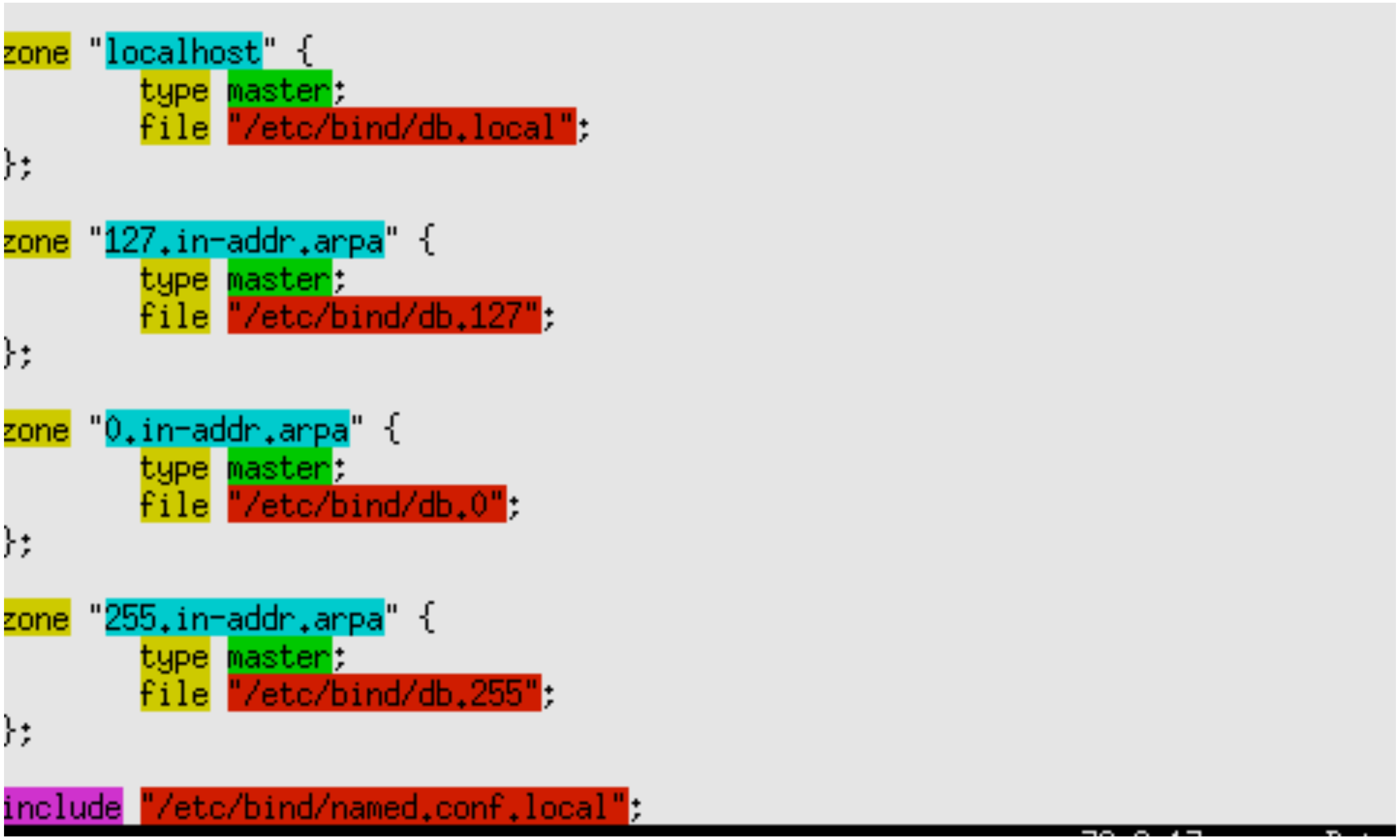




Bind

- 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

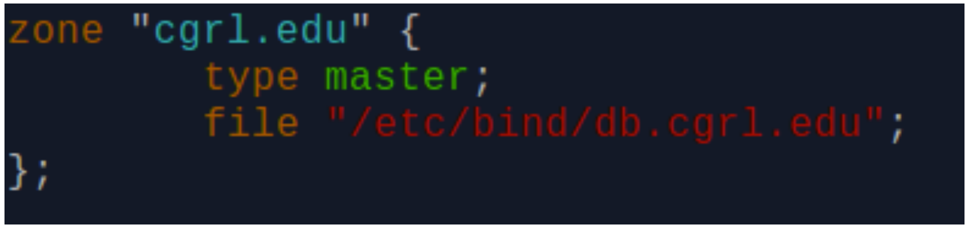


```
zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};

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

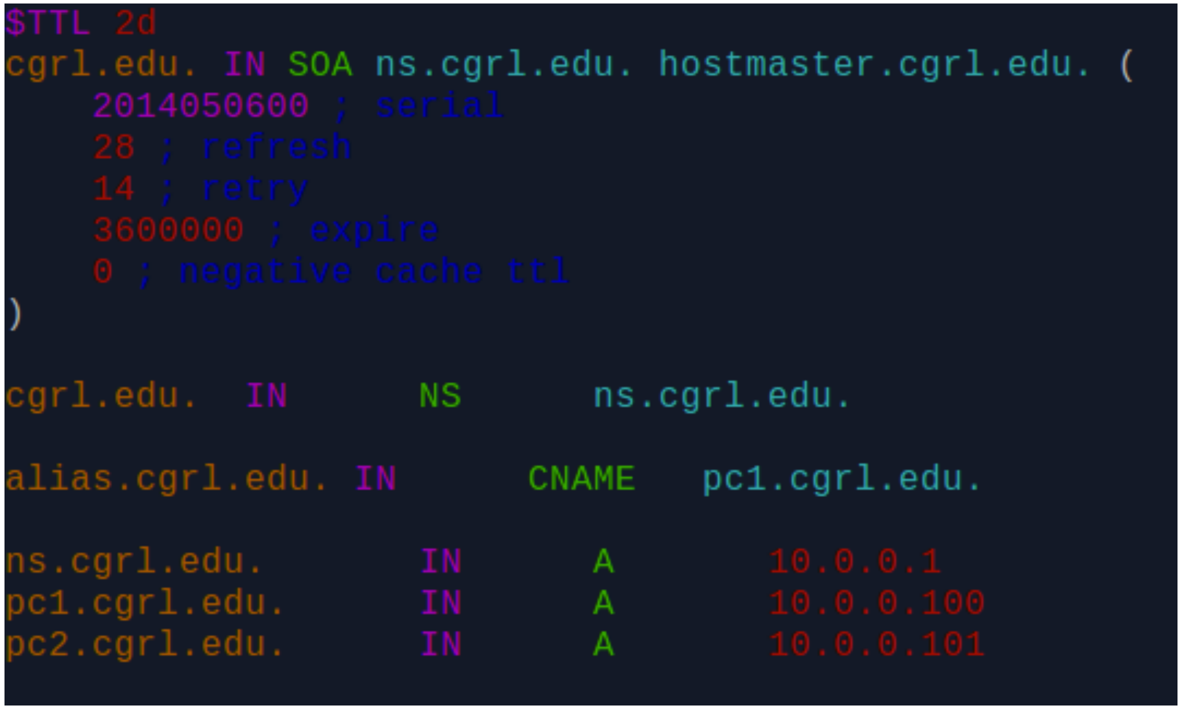
BIND configuration

/etc/bind/named.conf



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

/etc/bind/db.edu.cgrl

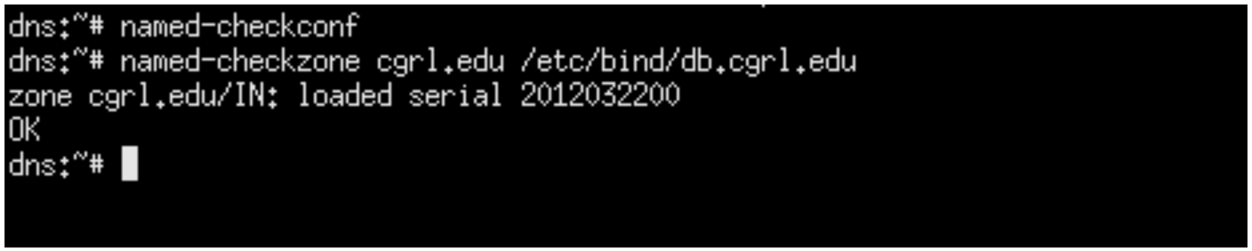


```
$TTL 2d
cgrl.edu. IN SOA ns.cgrl.edu. hostmaster.cgrl.edu. (
    2014050600 ; serial
    28 ; refresh
    14 ; retry
    3600000 ; expire
    0 ; negative cache ttl
)

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)

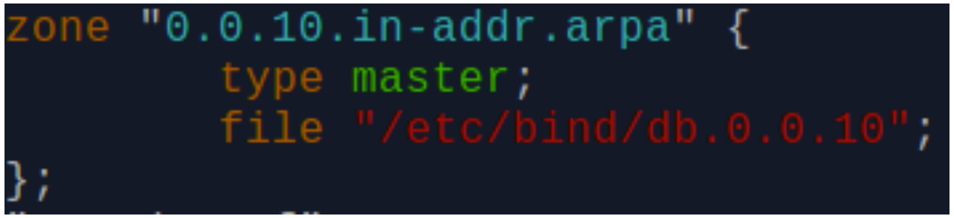


```
dns:~# named-checkconf
dns:~# named-checkzone cgrl.edu /etc/bind/db.cgrl.edu
zone cgrl.edu/IN: loaded serial 2012032200
OK
dns:~# 
```

And for reverse address mapping?

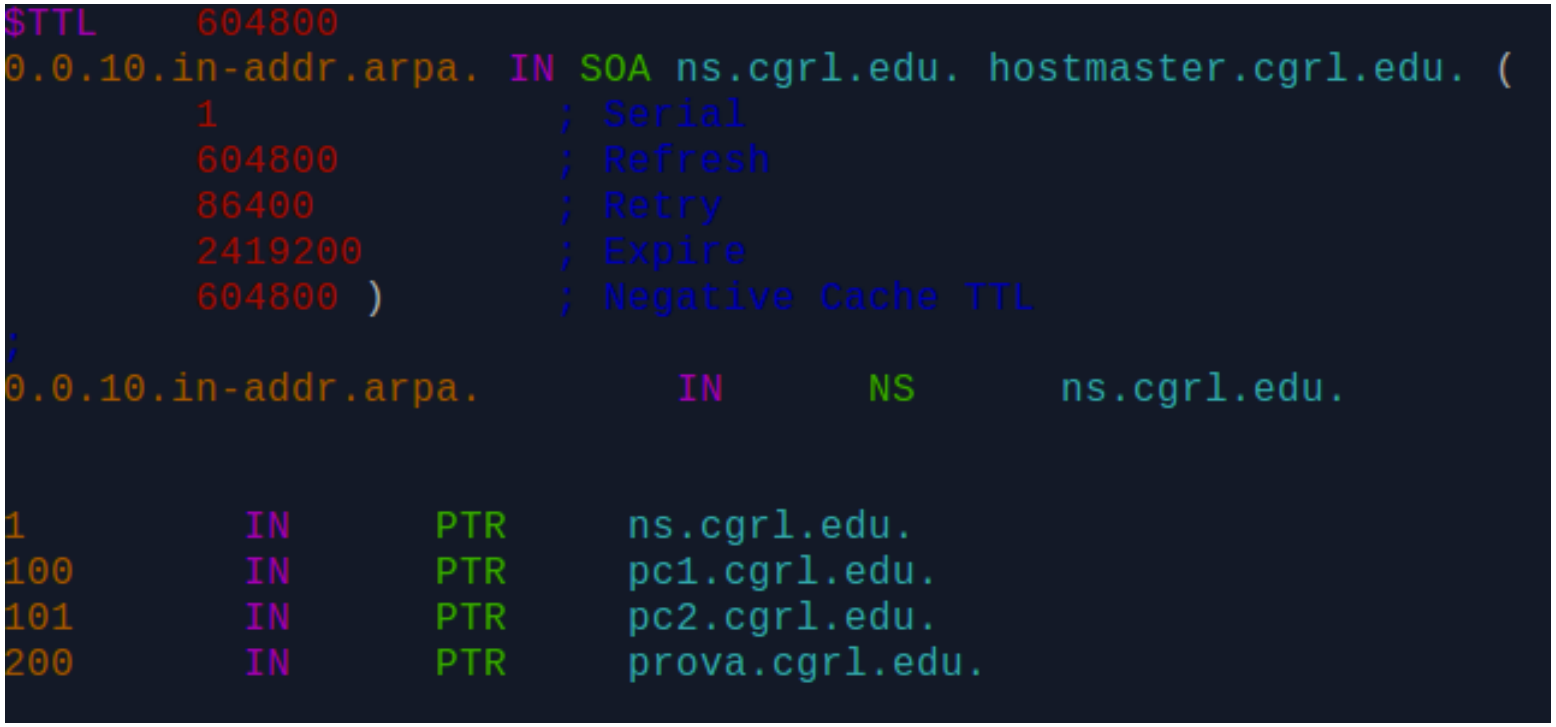
We simply make ns.cgrl.edu authoritative for the zone: 0.0.10.IN-ADDR.ARPA

/etc/bind/named.conf



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

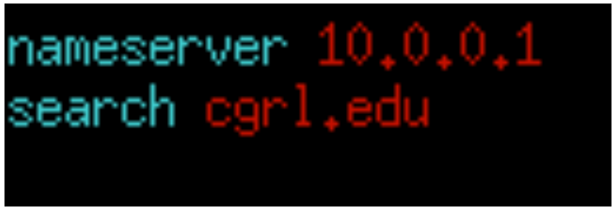
/etc/bind/db.0.0.10



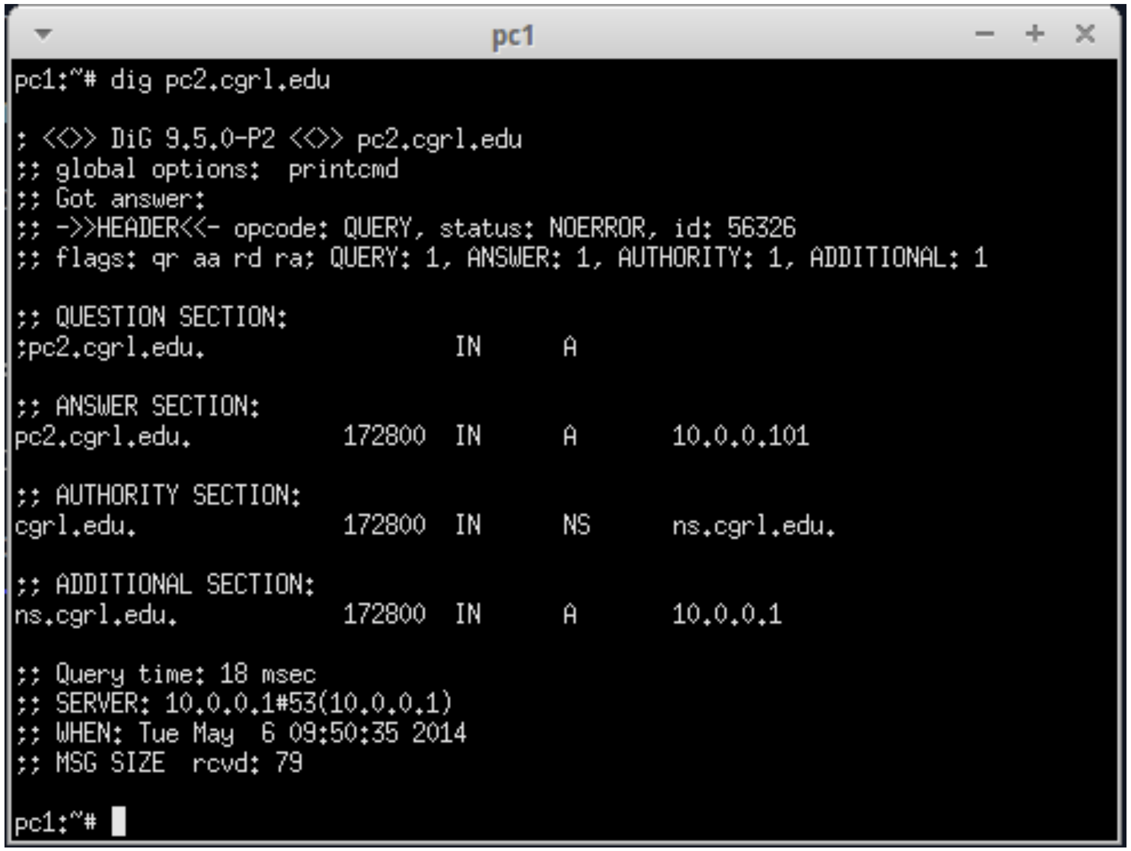
```
$TTL    604800
0.0.10.in-addr.arpa. IN SOA ns.cgrl.edu. hostmaster.cgrl.edu. (
        1               ; Serial
        604800          ; Refresh
        86400           ; Retry
        2419200         ; Expire
        604800 )        ; Negative Cache TTL
;
0.0.10.in-addr.arpa.        IN      NS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


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

Resolver configuration

/etc/resolv.conf



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



```
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
```

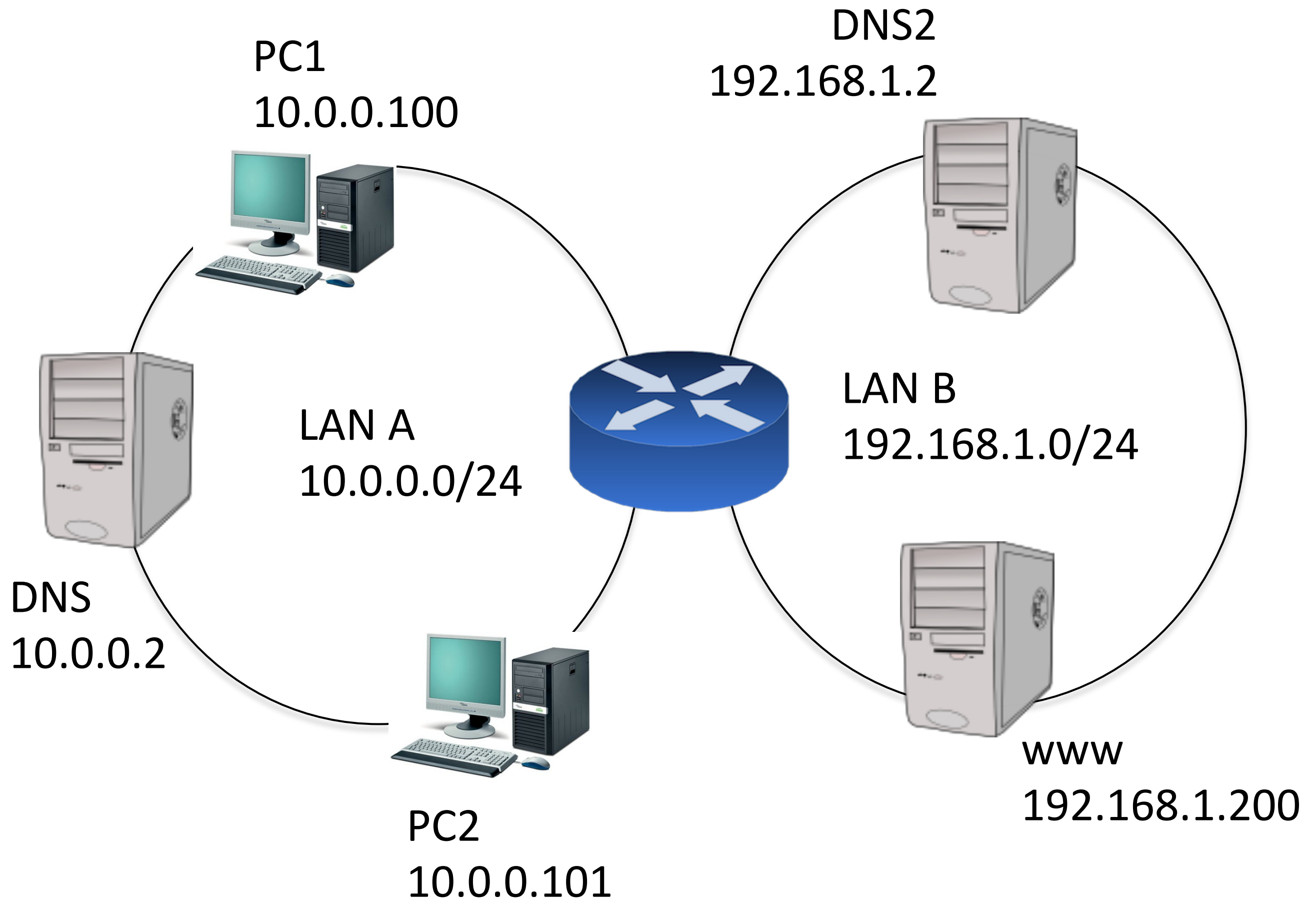
primary DNS
secondary DNS

```
domain mydomain.com
search mysearch.com d2.com
```

search directive for short names

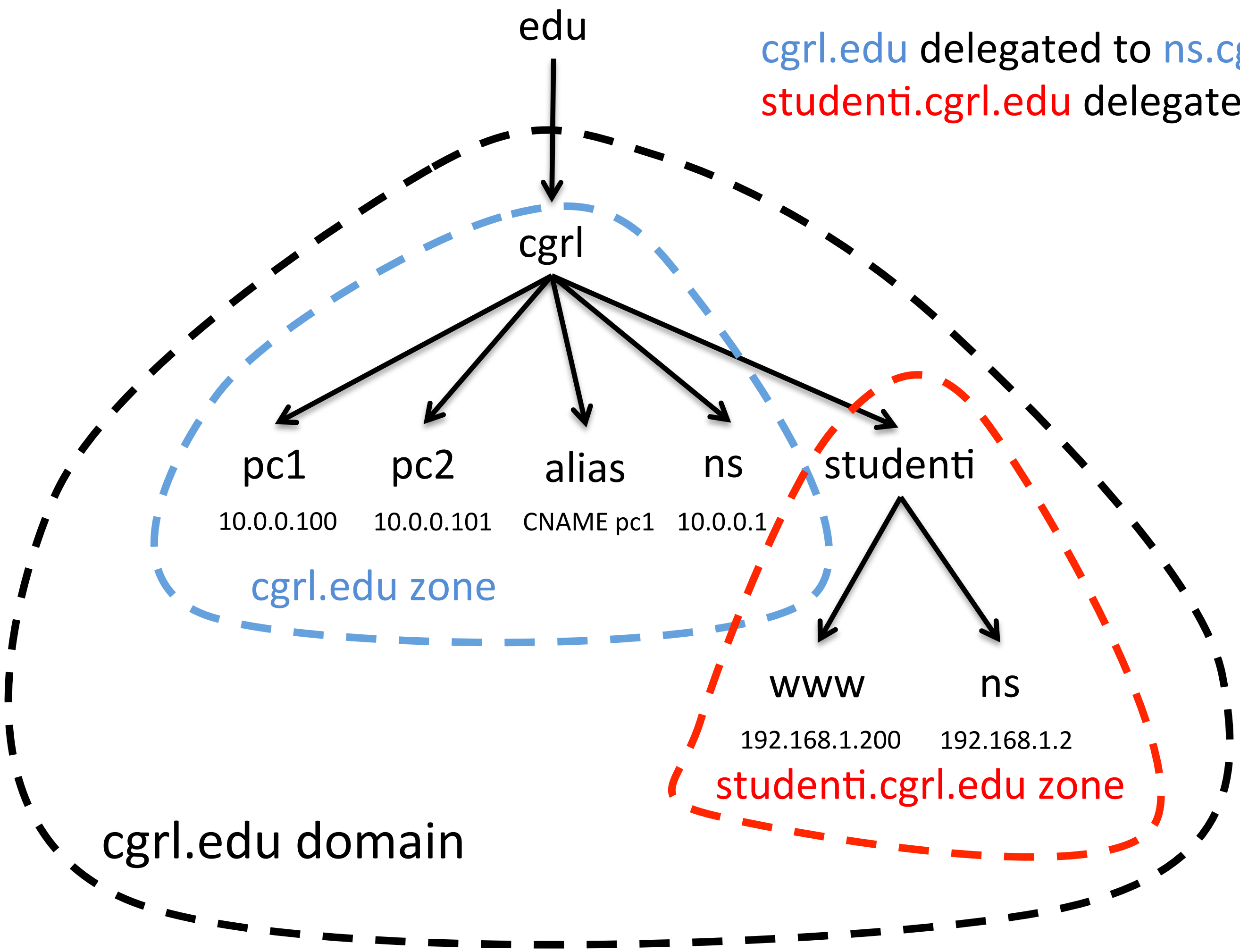
- When try to resolv “test” it resolve test.mydomain.com (using **gethostname** or *domain* if present)
- If you want that test will be resolved as test.A and test.B specify search A B. (in case test.A fails, resolver will go for test.B)
- The **domain** and **search** keywords are mutually exclusive. If more than one instance of these keywords is present, the last instance wins.
- *Let’s put 127.0.0.1 to test our new dns server!!*

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



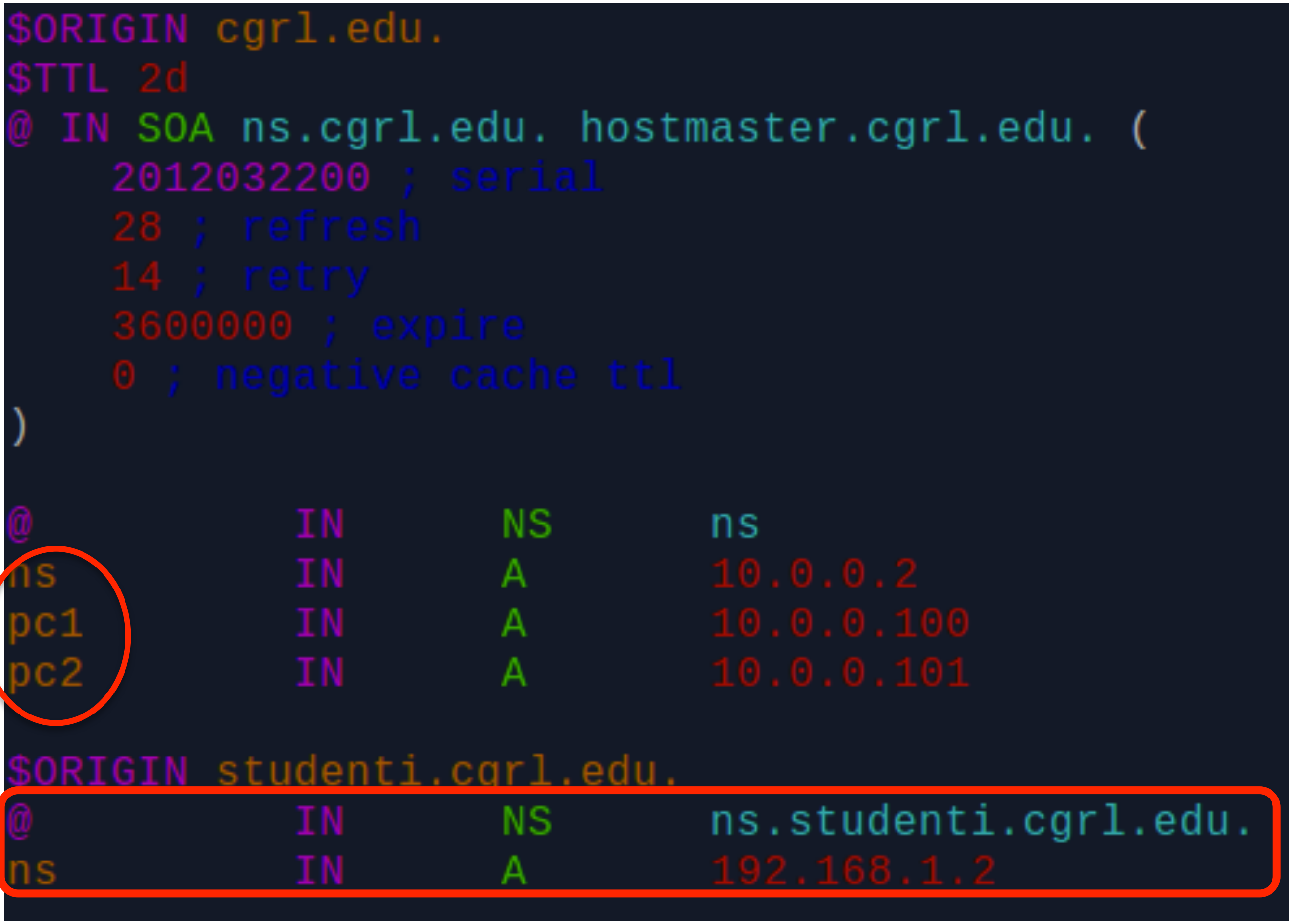
BIND configuration – dns

@ substitutes the current value of $ORIGIN

Relative names appended to current zone

delegation

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



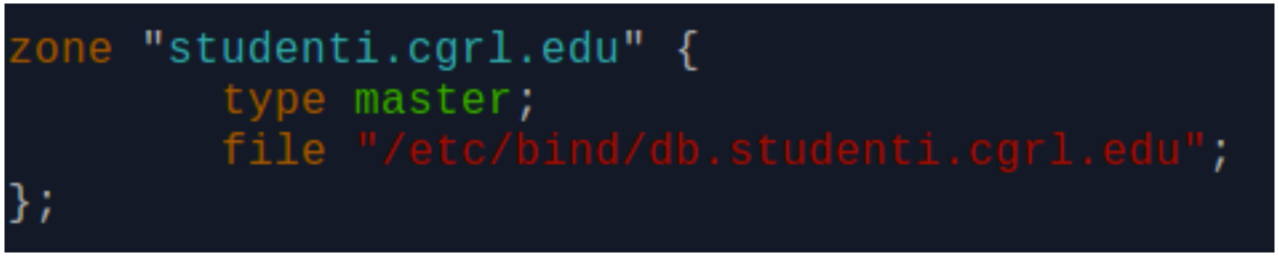
```
$ORIGIN cgrl.edu.
$TTL 2d
@ IN SOA ns.cgrl.edu. hostmaster.cgrl.edu. (
    2012032200 ; serial
    28 ; refresh
    14 ; retry
    3600000 ; expire
    0 ; negative cache ttl
)

@           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
```

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

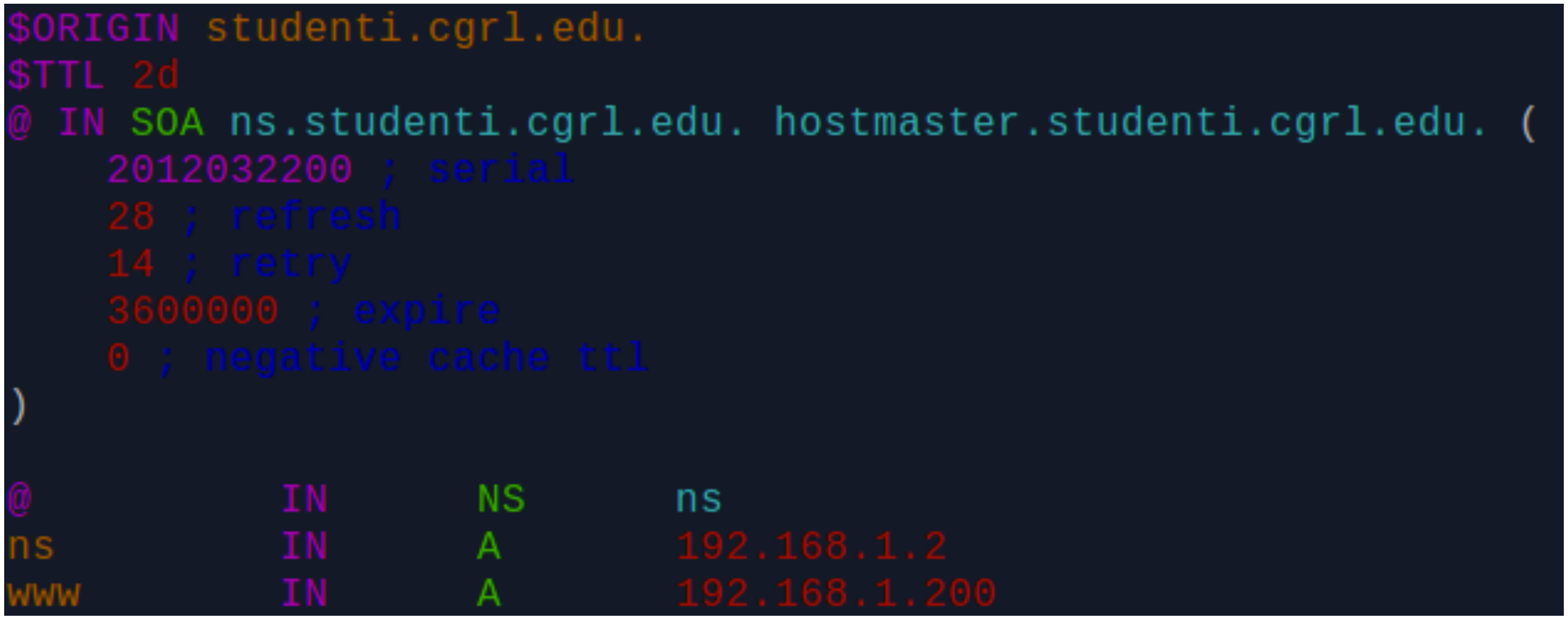
BIND configuration – dns2

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
```

MX records and load Balancing

- in most used MTA clients, if equal DNS preferences → Round robin!

```
IN MX 10 mail.example.com
IN MX 10 mail2.example.com
IN MX 10 mail3.example.com

mail IN A 192.168.0.4
mail2 IN A 192.168.0.5
mail3 IN 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 A RRs;
- the order of IP addresses in the returned list is defined by the rrset-order statement in BIND’s named.conf file.
  - *rrset-order {type MX name "example.com" order random; order cyclic};*
- Caching can significantly distort the effectiveness of any DNS IP address allocation algorithm. A TTL value of 0 may be used to inhibit

Mail server failover

```
; zone file fragment
IN MX 10 mail.example.com.
IN MX 20 mail.example.net.
.... mail IN A     192.168.0.4 ....
```

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

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.

- 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

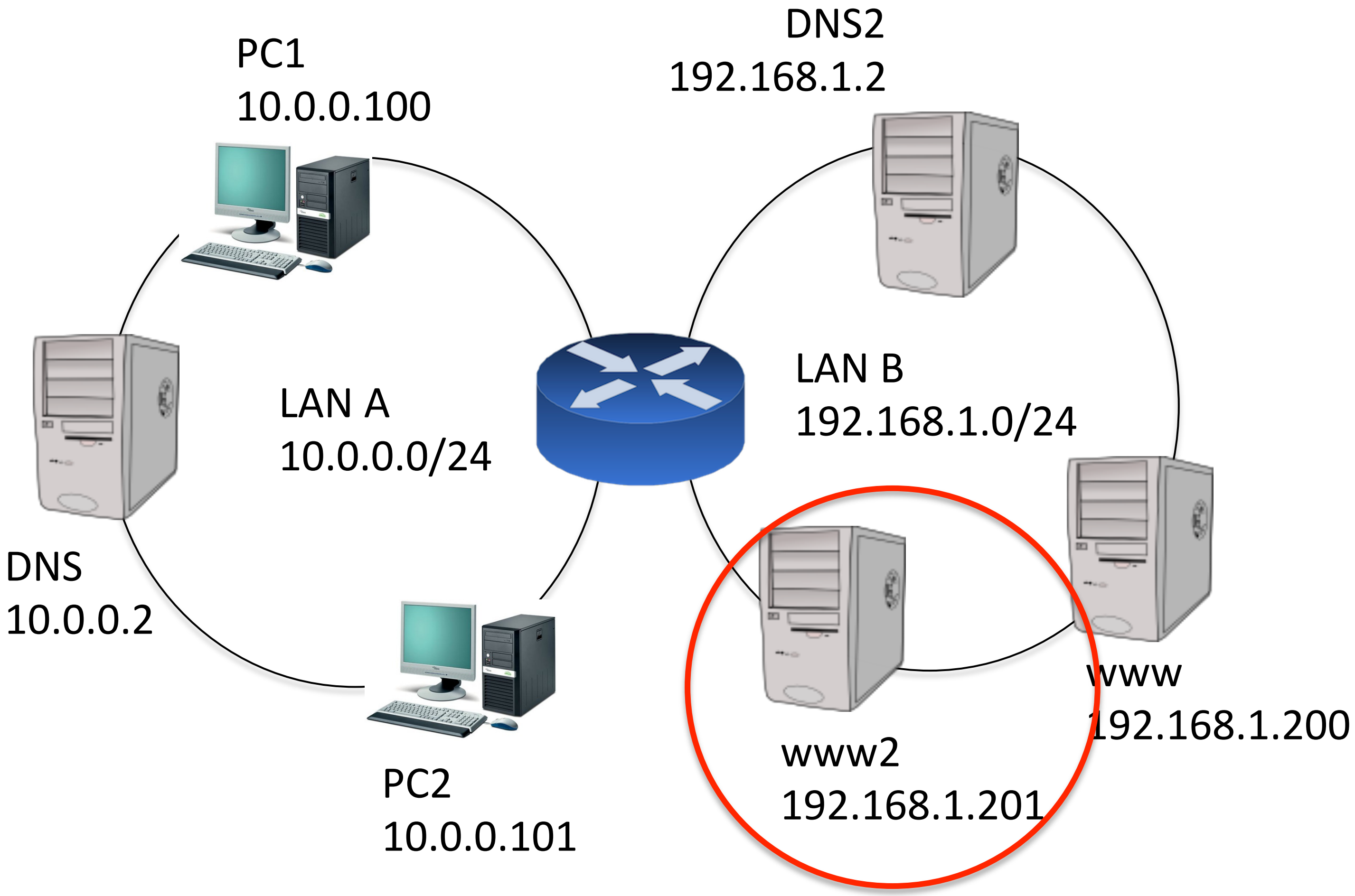
```
==> 220 teamits105.teamITS.net ESMTP Sendmail 8.13.6.20060614/8.13.6; Wed, 6 Dec 2007 14:27:47
-0600 (CST)
<-- HELO teamits104.teamITS.net
==> 250 teamits105.teamITS.net Hello py-in-f99.google.com [64.233.167.99], pleased to meet you
<-- mail from: sender@teamITS.com
==> 250 2.1.0 sender@teamITS.com... 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

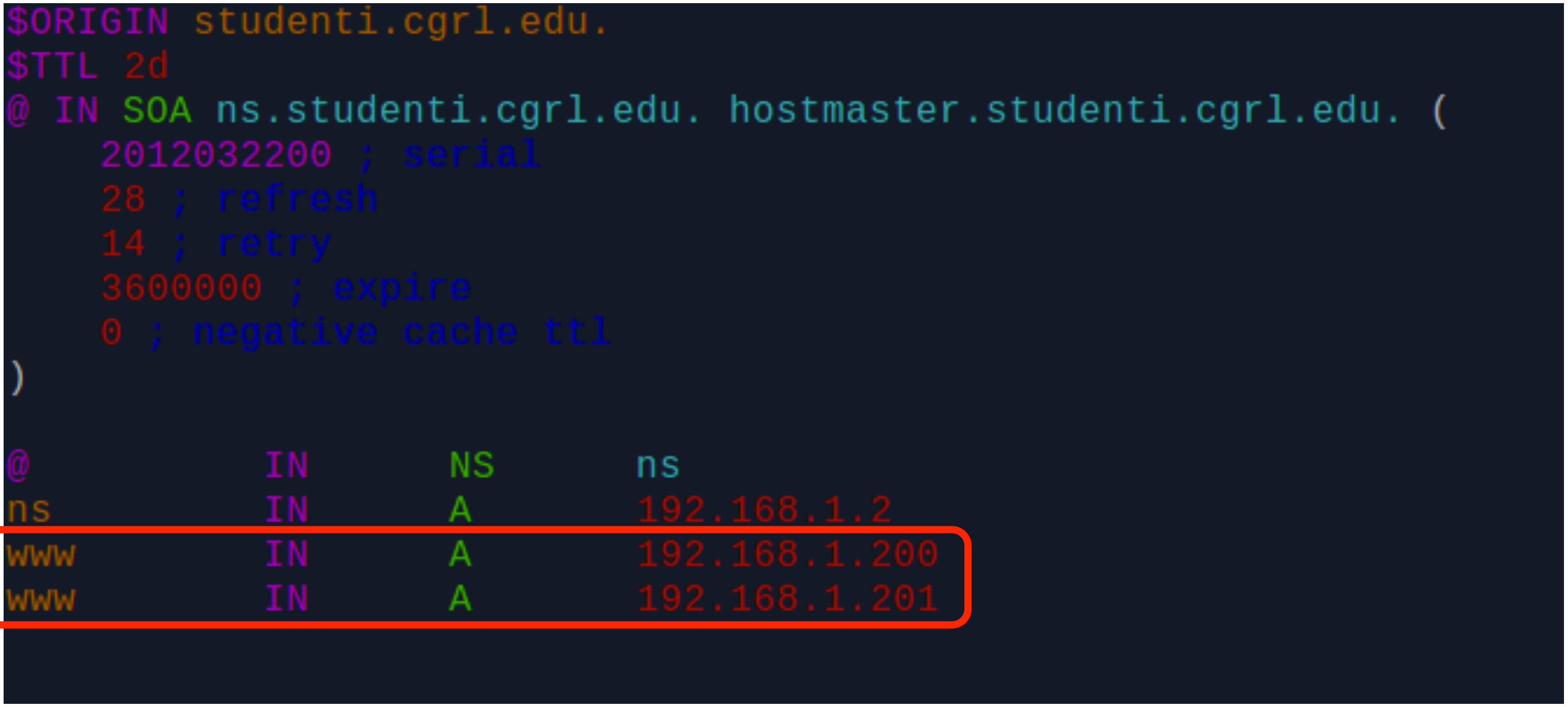
Exercise in class

Add www2 VM and load balance www.studenti.cgrl.edu between www and www2



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 throoguh 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
```

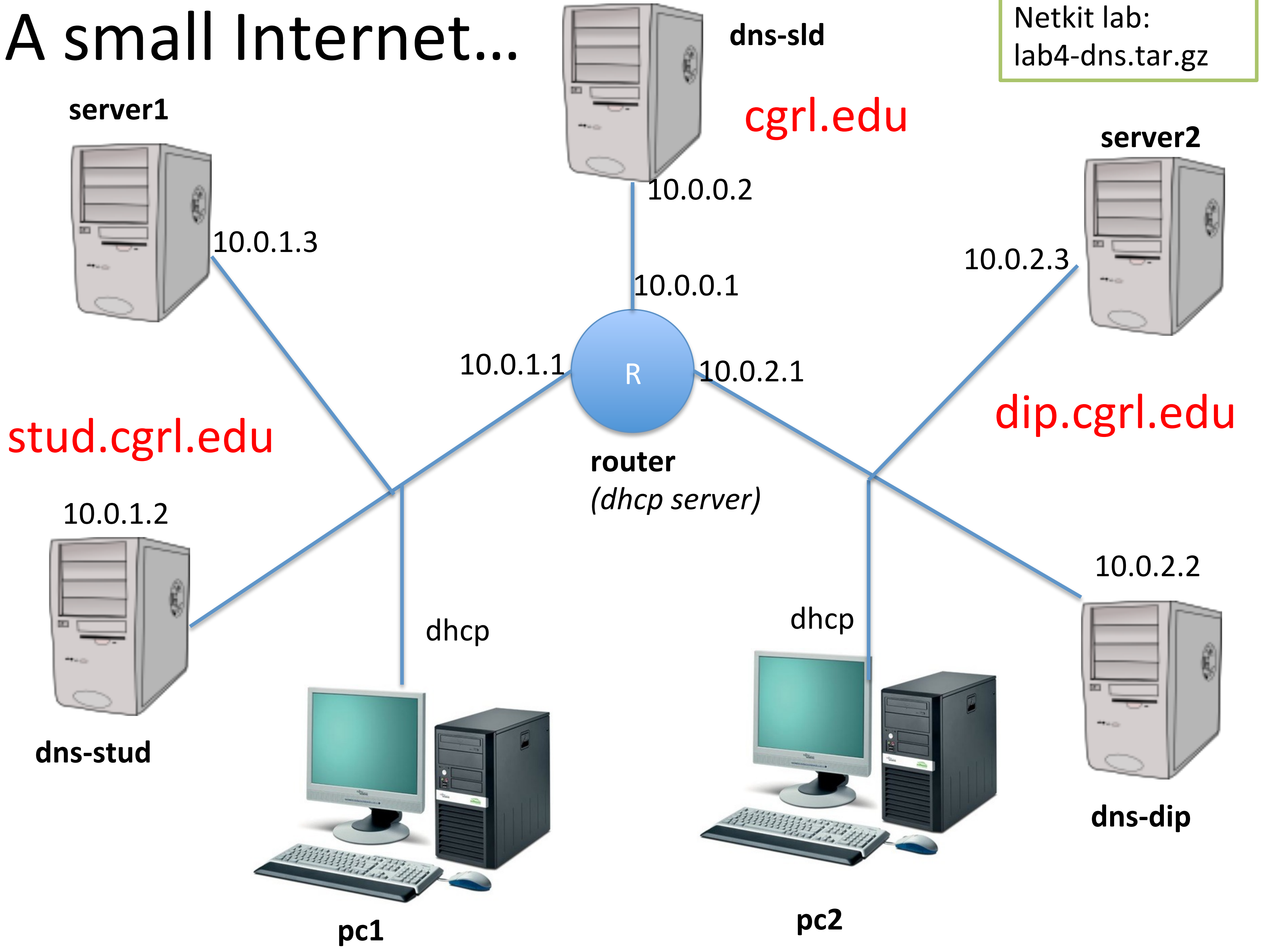
Question

Why www can’t resolve, for example, pc1.cgrl.edu?

Solution?

A small Internet…

Netkit lab:
lab4-dns.tar.gz



Statements: BIND

- many!
  - http://www.zytrax.com/books/dns/ch7/statements.html
- allow-transfer {192.168.1.2;}; (default yes)
- or selective:

```
zone "example.com" in {
.... allow-transfer {192.168.1.2;}; ....
};
```

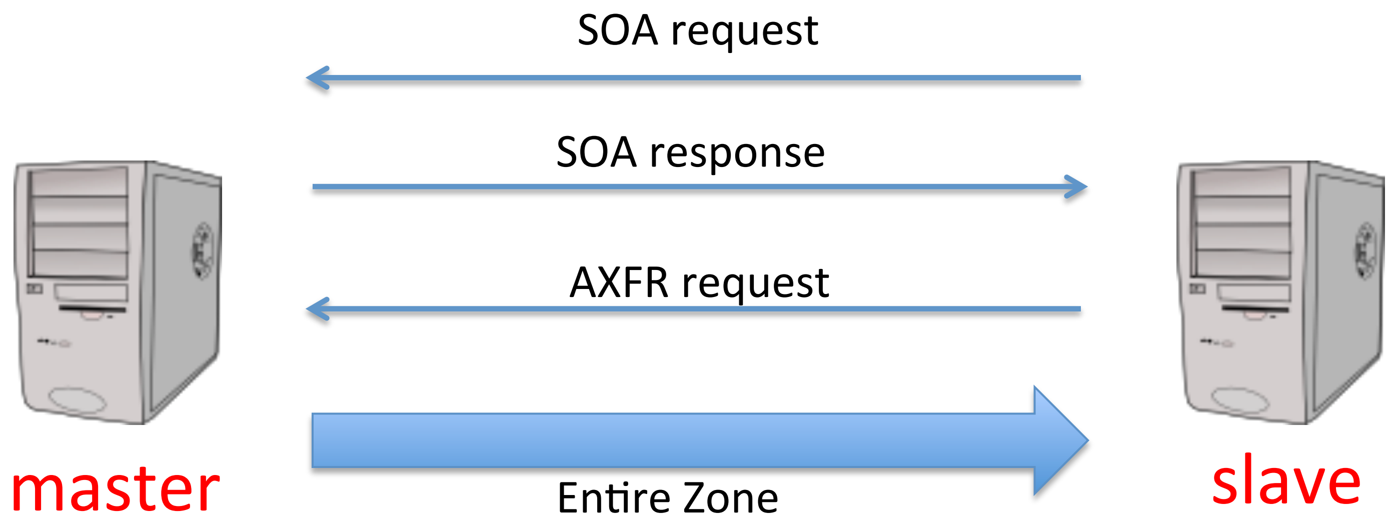
- The allow-notify {192.168.254.2;}; statement disables NOTIFY messages from any host except the zone master to minimize possible malicious action.

View clause

- To offer different services to different clients (e.g. inside and outside our company)
- The view statement can take a serious number of statements

```
view "goodguys" {
match-clients { 192.168.254.0/24; }; // the example.com network
recursion yes; // required zone for recursive queries zone
"." {
        type hint;
        file "root.servers";
};
```

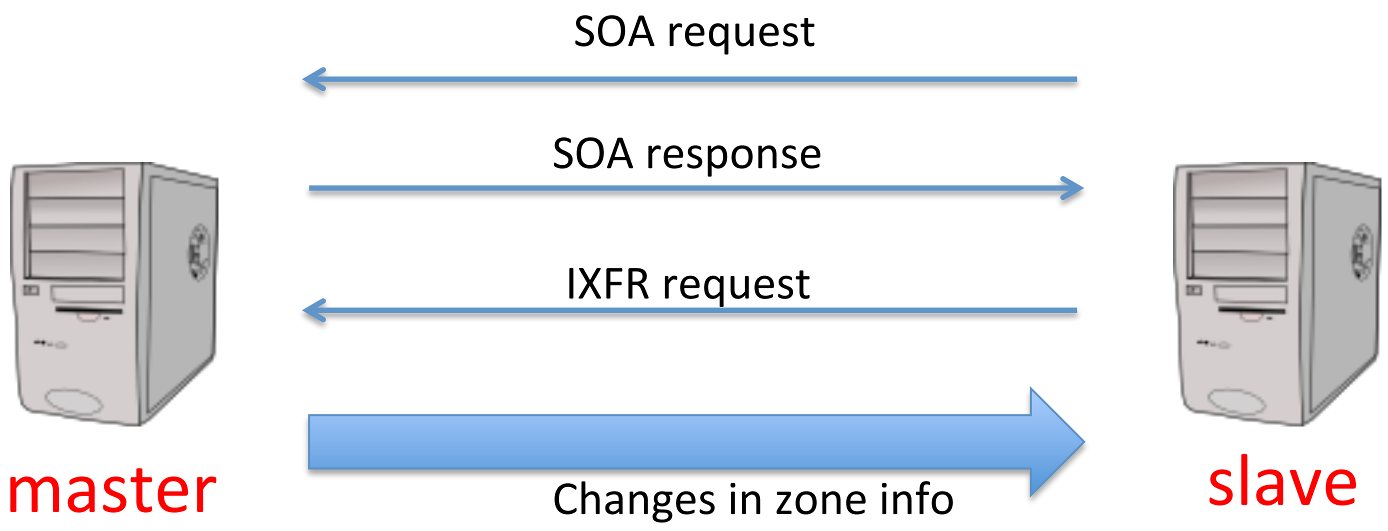
Master Slave configuration: AXFR



Full Zone Transfer

- Master: the zone file will be read from the local filestore
- Slave: obtains the zone records using zone transfer
- Everything done using TCP, zone transfer are always started by clients

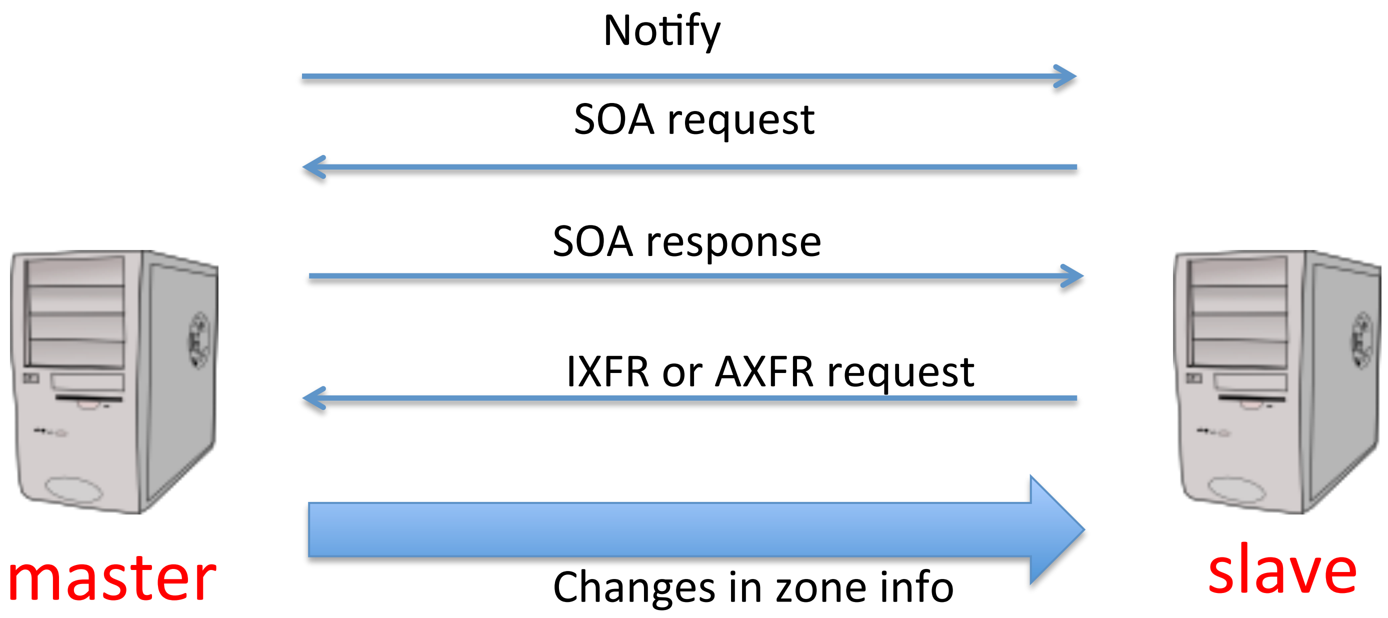
Master Slave configuration: IXFR



Incremental zone transfer

- Requests a zone transfer of the given zone but only differences from a previous serial number.
- AXFR can be sent if the authoritative server is unable to fulfill the request due to configuration or lack of required deltas.

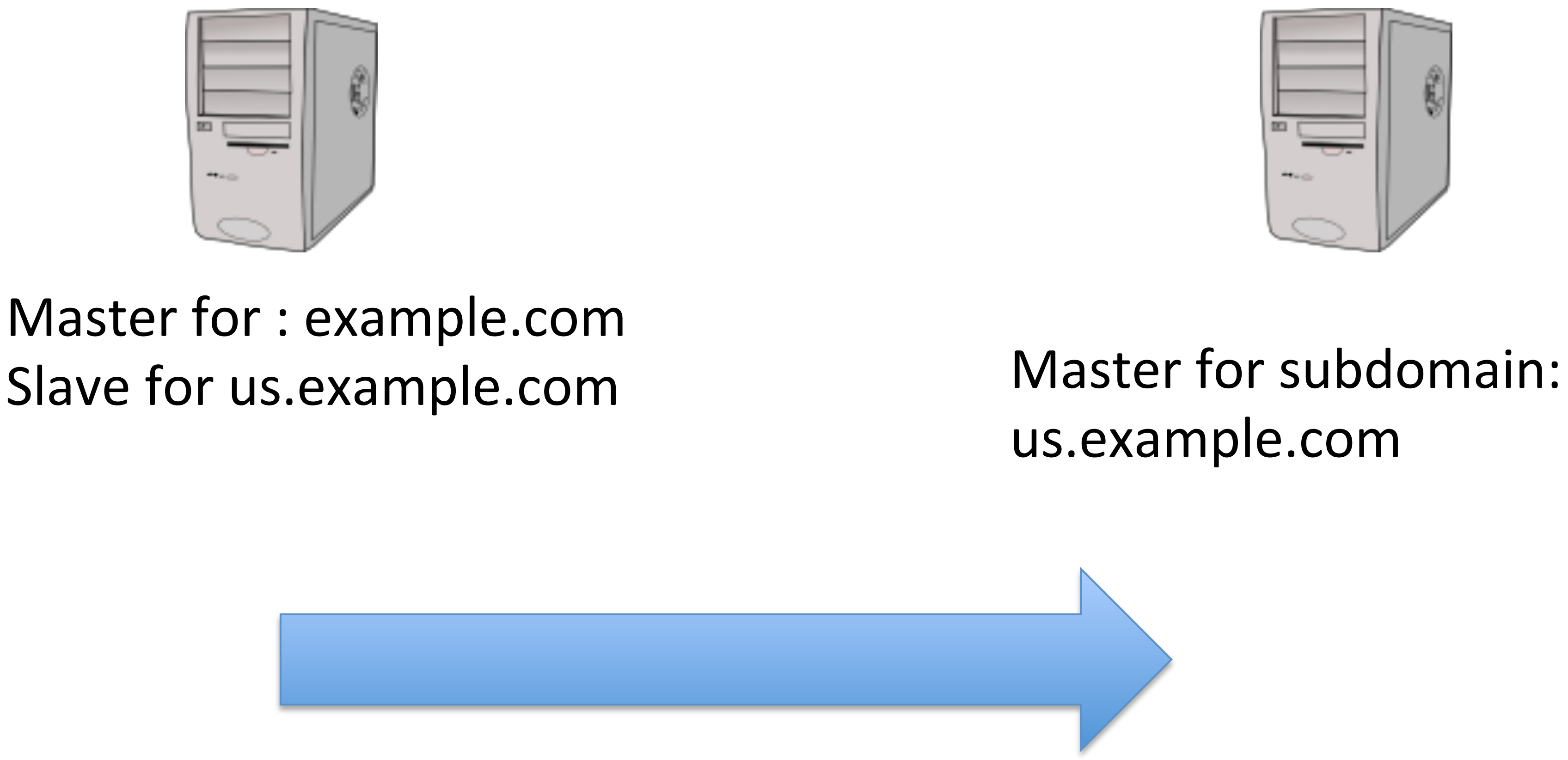
Master Slave configuration: Notify



servers can send a NOTIFY message to clients to signal changes

Notify decrease latency and propagation time of zone changes

Example: delegation and redundancy



Delegation of subdomain: us.example.com

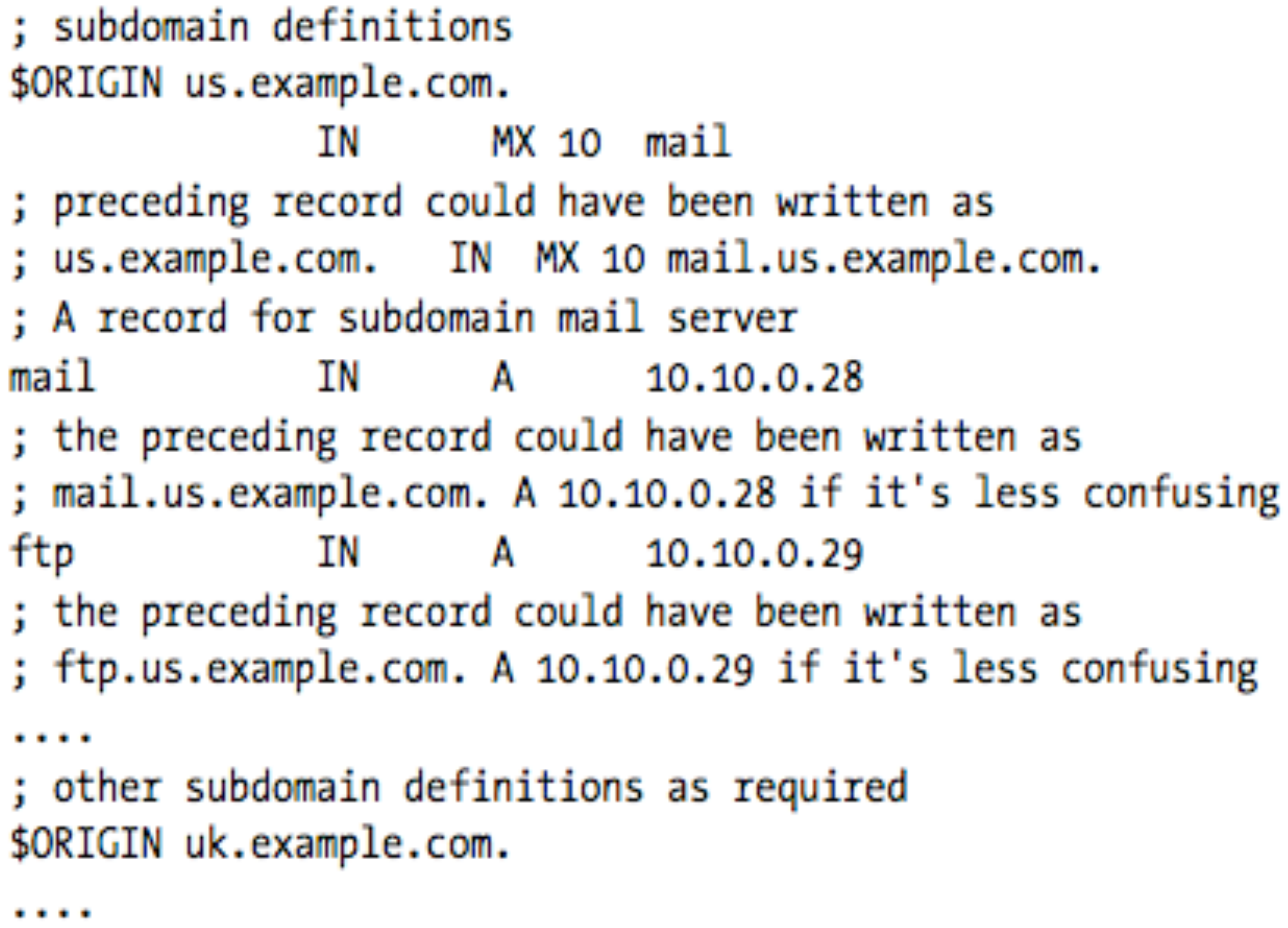
Bind: Delegate a Subdomain (Subzone)

```
zone "example.com" in{
     type master;
     file "master.example.com";
};
"us.example.com" IN {
    type slave;
    file "slave.us.example.com";
    masters {10.10.0.24;};
};
```

Delegation with redundancy

don’t require any additional name servers!

Virtual subdomain



```
; subdomain definitions
$ORIGIN us.example.com.
              IN      MX 10   mail
; preceding record could have been written as
; us.example.com.   IN  MX 10 mail.us.example.com.
; A record for subdomain mail server
mail          IN      A       10.10.0.28
; the preceding record could have been written as
; mail.us.example.com. A 10.10.0.28 if it's less confusing
ftp           IN      A       10.10.0.29
; the preceding record could have been written as
; ftp.us.example.com. A 10.10.0.29 if it's less confusing
....
; other subdomain definitions as required
$ORIGIN uk.example.com.
....
```

Assignee

Reverse delegation

- Example: how to reverse delegate subnet < /24:
  – RFC 2317

```
64/26           IN  NS  ns1.example.com.
64/26           IN  NS  ns2.example.com.
; the preceding could have been written as
; 64/26.199.168.192.IN-ARDDR.ARPA.  IN NS ns2.example.com.
; IPs addresses in the subnet - all need to be defined
; except 64 and 127 since they are the subnets multicast
; and broadcast addresses not hosts/nodes
65              IN  CNAME   65.64/26.199.168.192.IN_ADDR.ARPA. ;qualified
66              IN  CNAME   66.64/26 ;unqualified name
67              IN  CNAME   67.64/26
....
125             IN  CNAME   125.64/26
126             IN  CNAME   126.64/26
; end of 192.168.199.64/26 subnet
```

Assignor

(End-user) Zone File

- Simple!

```
65 IN PTR fred.example.com.
66 IN PTR joe.example.com.
67 IN PTR bill.example.com.
```

Out-of-Sequence Serial Numbers

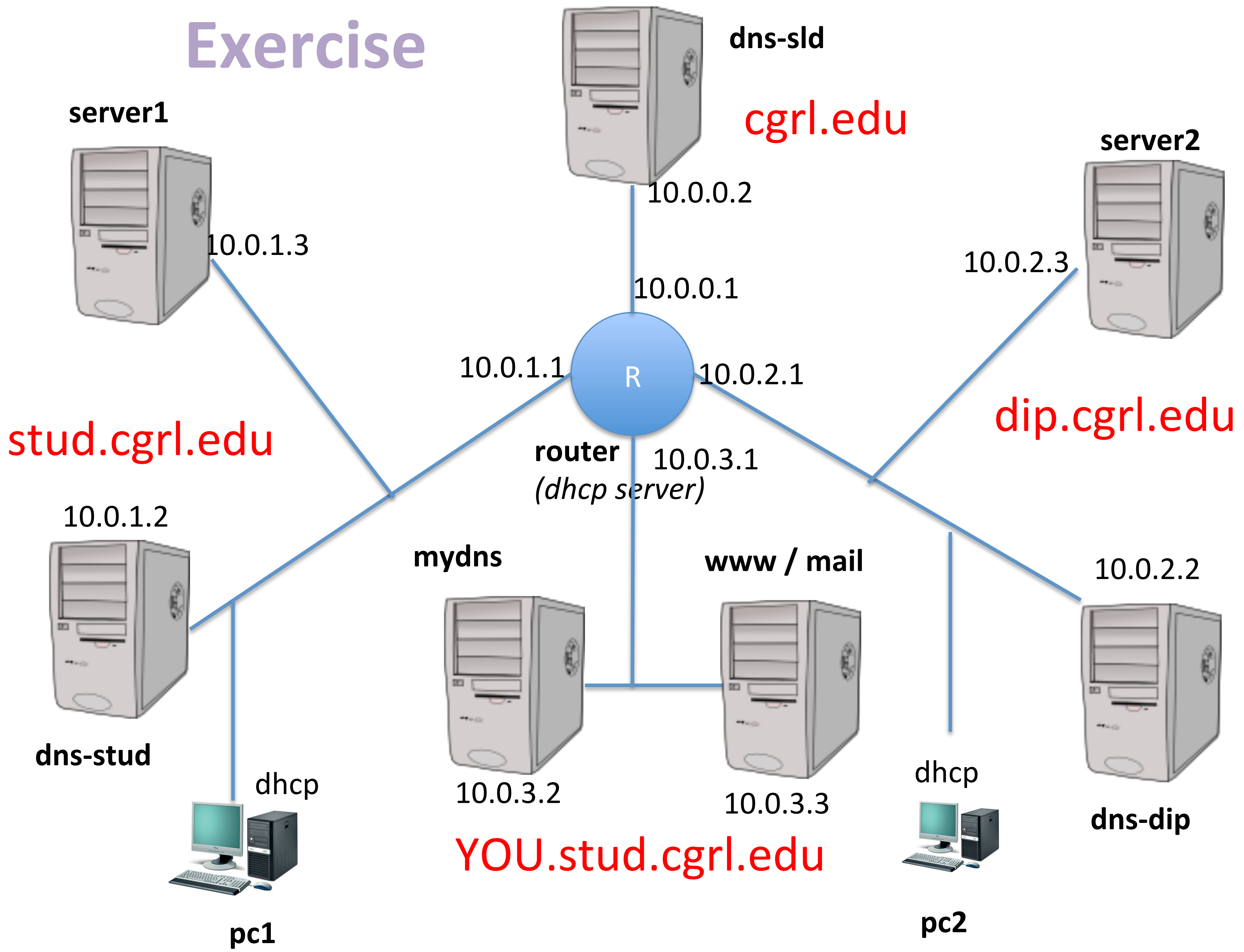
- SN = 4 byte int and set as a date (convention)
  - bigger SN, newer the data
- what if we make a mistake and put a data in the future?
  - what till the future will come to correct the error
  - increment by 2^31 the value, push to all the slaves, and then put the right value (wrapped through zero)

Wildcard

```
@ IN MX 10 mail.example.com.
*   IN MX 10 mail.example.com.
```

- an MX query everythingelse.example.com will return the host mail.example.com.

Exercise



dns-sld
cgrl.edu
10.0.0.2
server1
10.0.1.3
10.0.0.1
server2
10.0.2.3
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