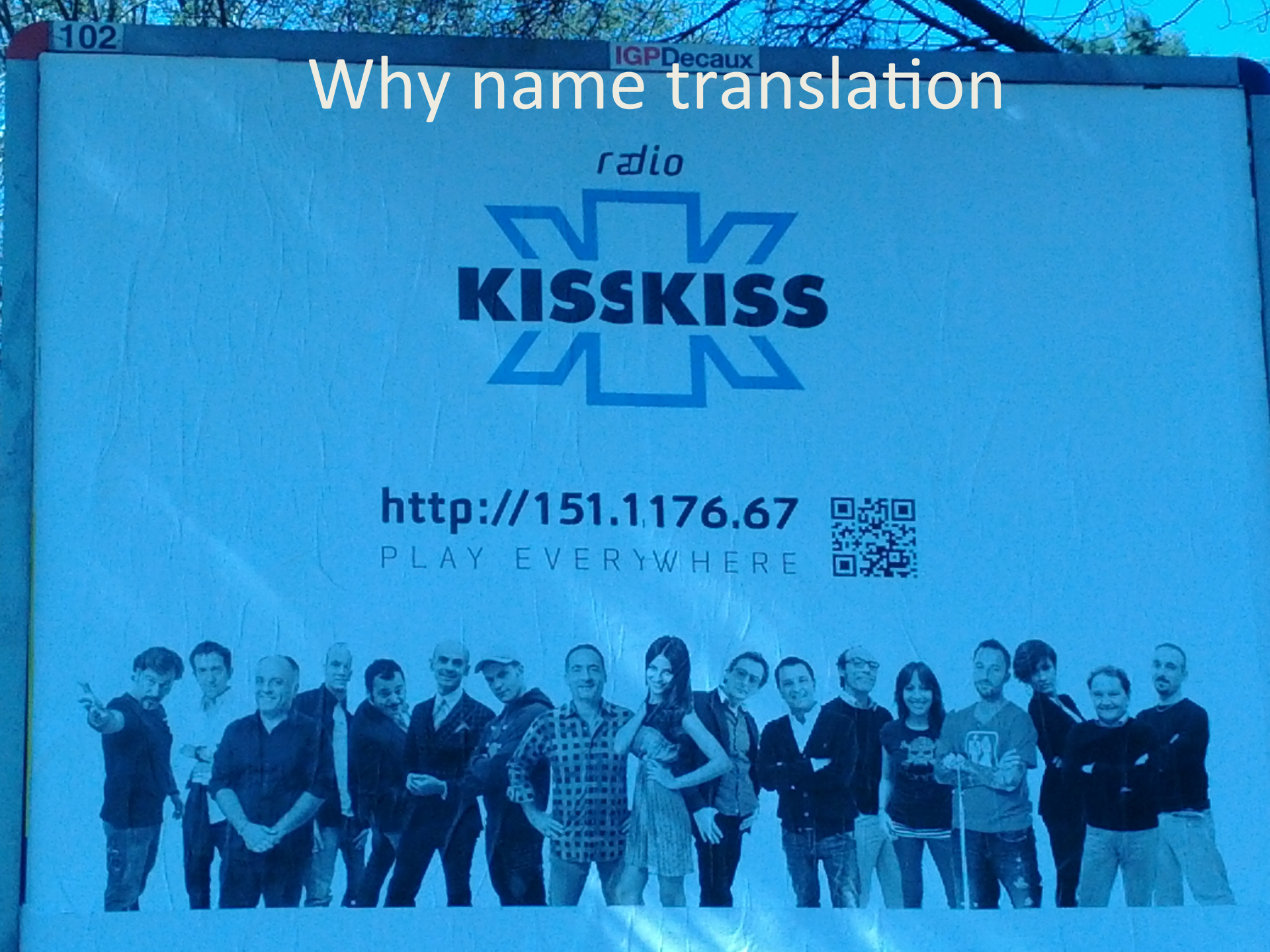
DNS & BIND

Lorenzo Bracciale
Marco Bonola

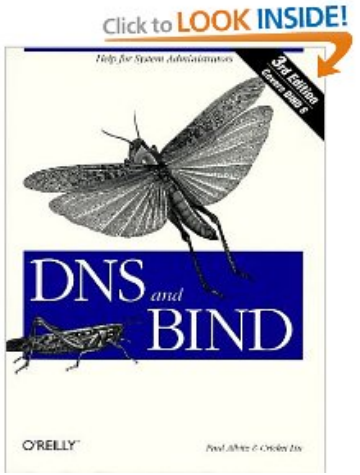
Why name translation



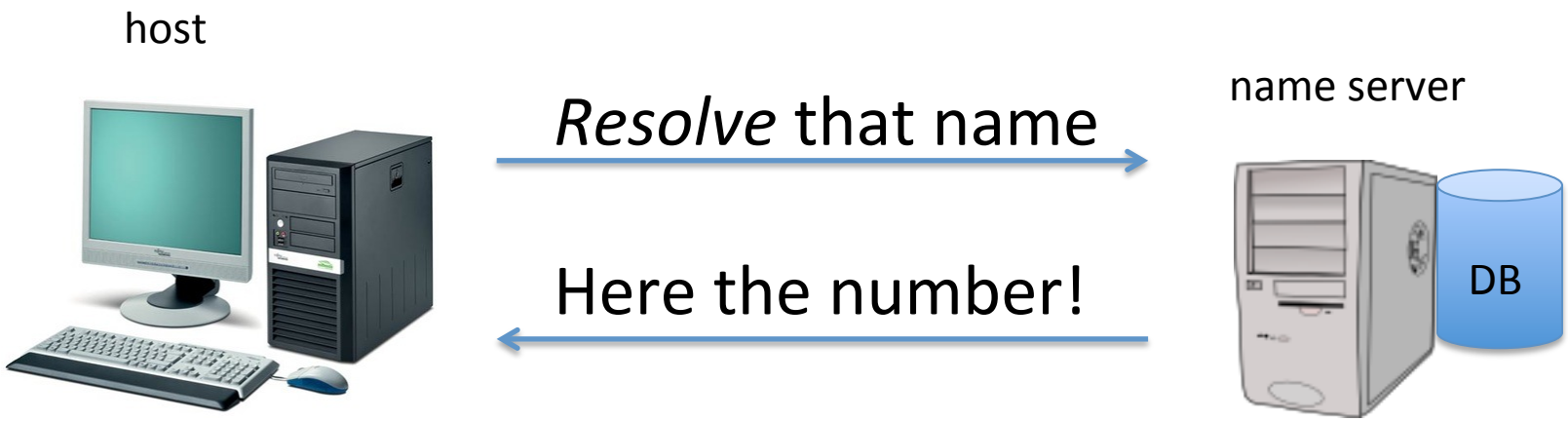
Need for name translation

- 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)



Simple solution



- On Internet
  - need of a *scalable* solution (today > ~140k domains[1])
  - introduce hierarchical names*:* *www.example.com.*
  - Key concept: *authority and delegation*

"silent dot"

[1] http://www.domaintools.com/internet-statistics/

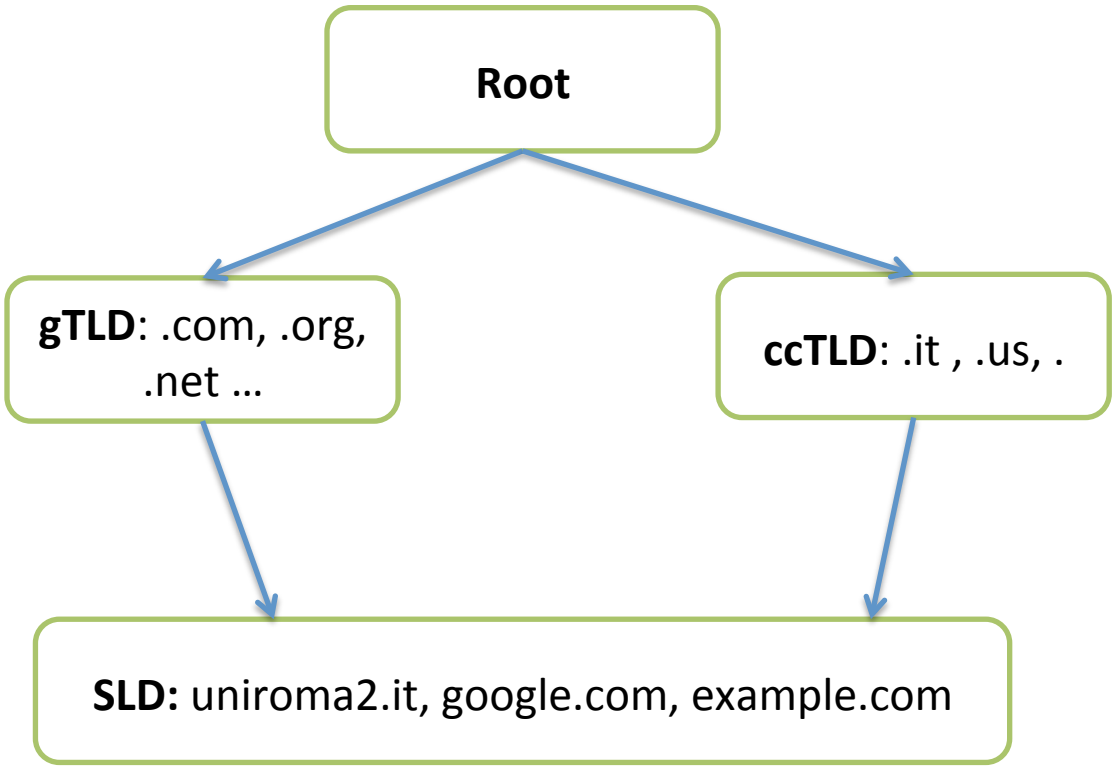
First experiment by Paul Mockapetris 1983

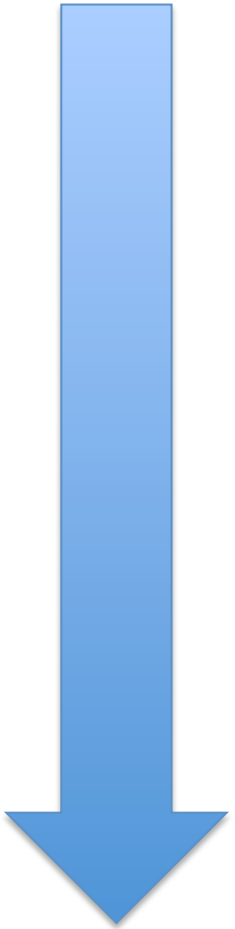
Internet Domain Name System

Root

TLD

SLD





Each domain has an Authority:

Root: Internet Corporation for Assigned Numbers and Names (ICANN—www.icann.org/)

DELEGATION

Zone and Resource Records

- Part of the Domain Name Space can be delegated: this is called a zone
  - The zone can delegate other parts
- Every zone has some Resource Records
  - different types (e.g. A, PTR, MX)

Updating names: let's buy a "domain"

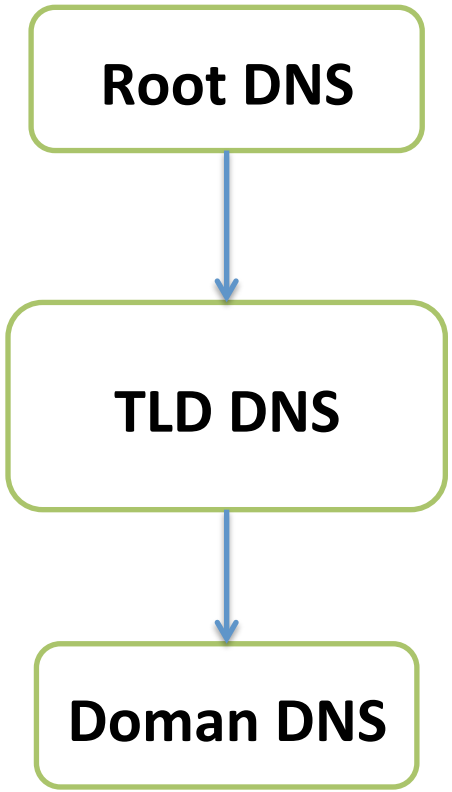
- A registrar interacts with public, store detailed information, and pass a "digest" to registry operator.
- Registry operator build a "zone file" (i.e. Data describing the domain) and pass it to interested TLD
- Periodically, ICANN distribute a "TLD master file" to each Root Server.

www.example.com

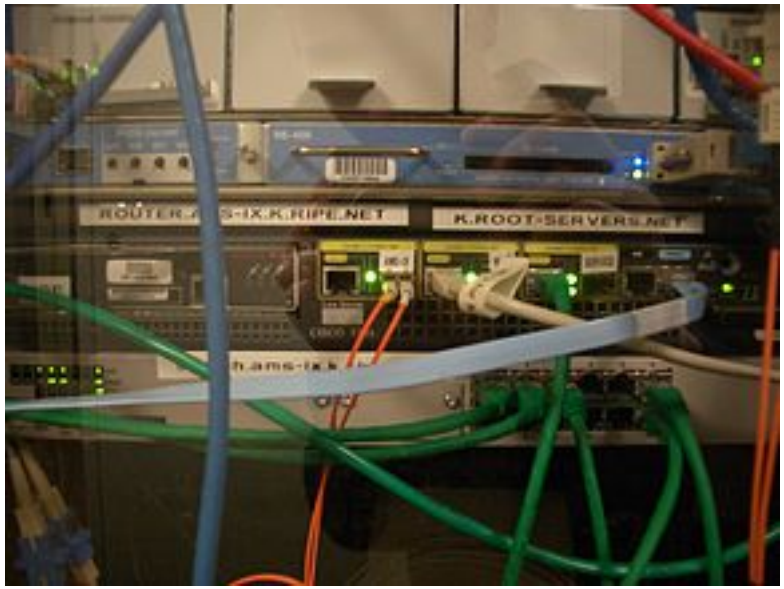
- The domain name example.com part was delegated from a **gTLD registrar**, which in turn was delegated from **ICANN**.
- The owner of the domain chooses the www part (called host name)
- This is a Fully Qualified Domain Name (**FQDN**)
  - specifies an exact location in the DNS tree hierarchy

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 name 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

DNS Queries: iterative vs recursive

Query www.uniroma2.it

root server

referral to .it ccTLD DNS

Query www.uniroma2.it

TLD DNS

referral to uniroma2.it DNS

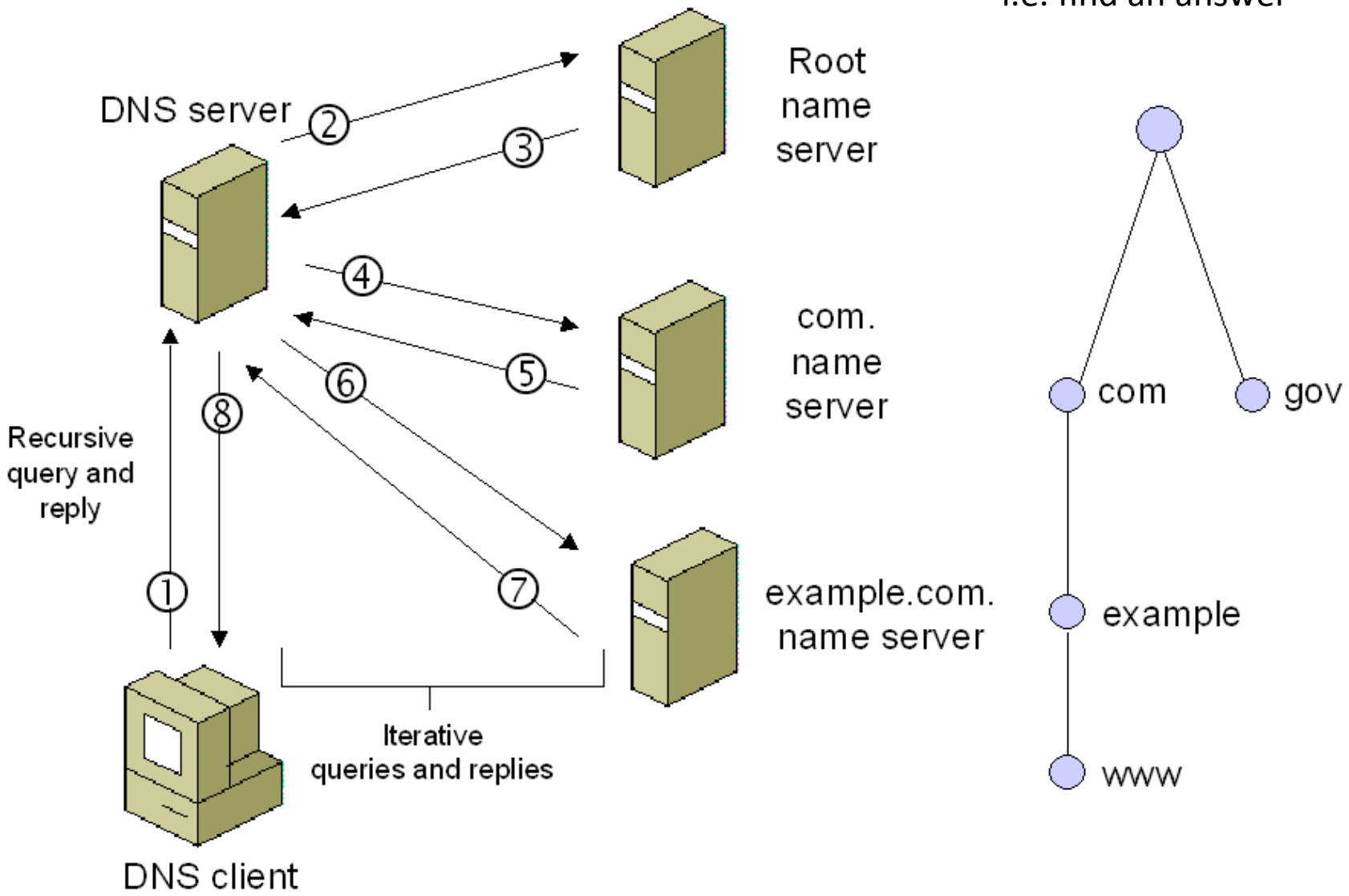
Query www.uniroma2.it

Domain DNS

Authoritative answer

DNS Queries: iterative vs recursive

i.e. find an answer



trace in http://stud.netgroup.uniroma2.it/cgrl/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
```

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

DNS Resolver

- The client-side of the DNS is usually called a DNS resolver.
- These simple resolvers (called "stub resolvers") cannot follow referrals
  - Need a recursive DNS
- Browser use *gethostbyname* or *gethostbyaddr* methods to invoke name/ip resolution

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

debian package: *dnsutils*

```
; <<>> 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
```

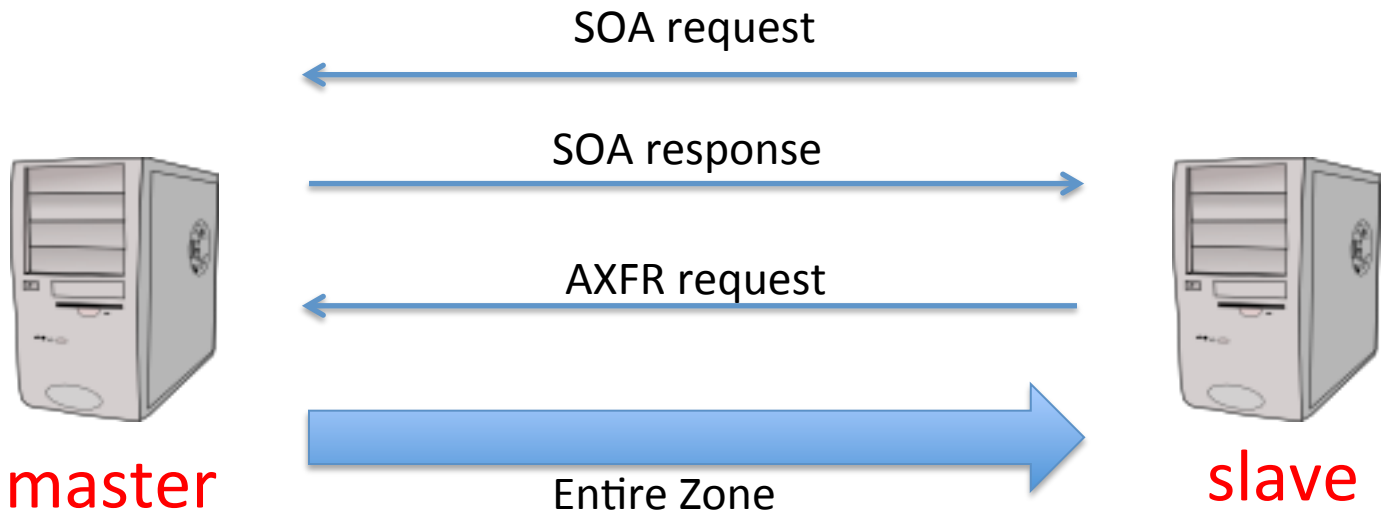
Dig

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

Master Slave configuration



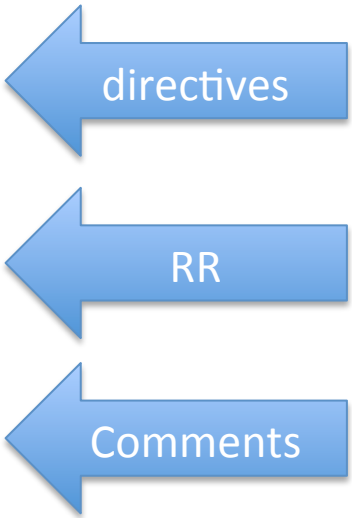
- redundancy for load balancing and fault resilience
- zones are passed from master to slave
  - full or partial zone transfer
- timing?

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)

Zone File: Example

```
$ORIGIN example.com.     ; designates the start of this zone file in the namespace
$TTL 1h                  ; default expiration time TTL value
example.com.  IN  SOA  ns.example.com. username.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. ; ns.somewhere.example is a backup nameserver for
example.com
example.com.  MX    10 mail.example.com.  ; mail.example.com is 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
www           CNAME example.com.          ; www.example.com is an alias for example.com
mail          A     192.0.2.3             ; IPv4 address for mail.example.com, (MX records must address record- RFC 2181)
mail2         A     192.0.2.4             ; IPv4 address for mail2.example.com
mail3         A     192.0.2.5             ; IPv4 address for mail3.example.com
```



Syntax: SOA RR

- Specifies authoritative information about a DNS zone
- 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
- Defined in RFC 1035

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 unqualfied 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.

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
  - Query: 1.IN-ADDR.ARPA
  - Query: 2.1.IN-ADDR.ARPA
  - Query: 3.2.1.IN-ADDR.ARPA
  - Query: 4.3.2.1.IN-ADDR.ARPA

Reverse Mapping: zone file

…

$ORIGIN 254.168.192.IN-ADDR.ARPA.

…

17 IN PTR www.example.org

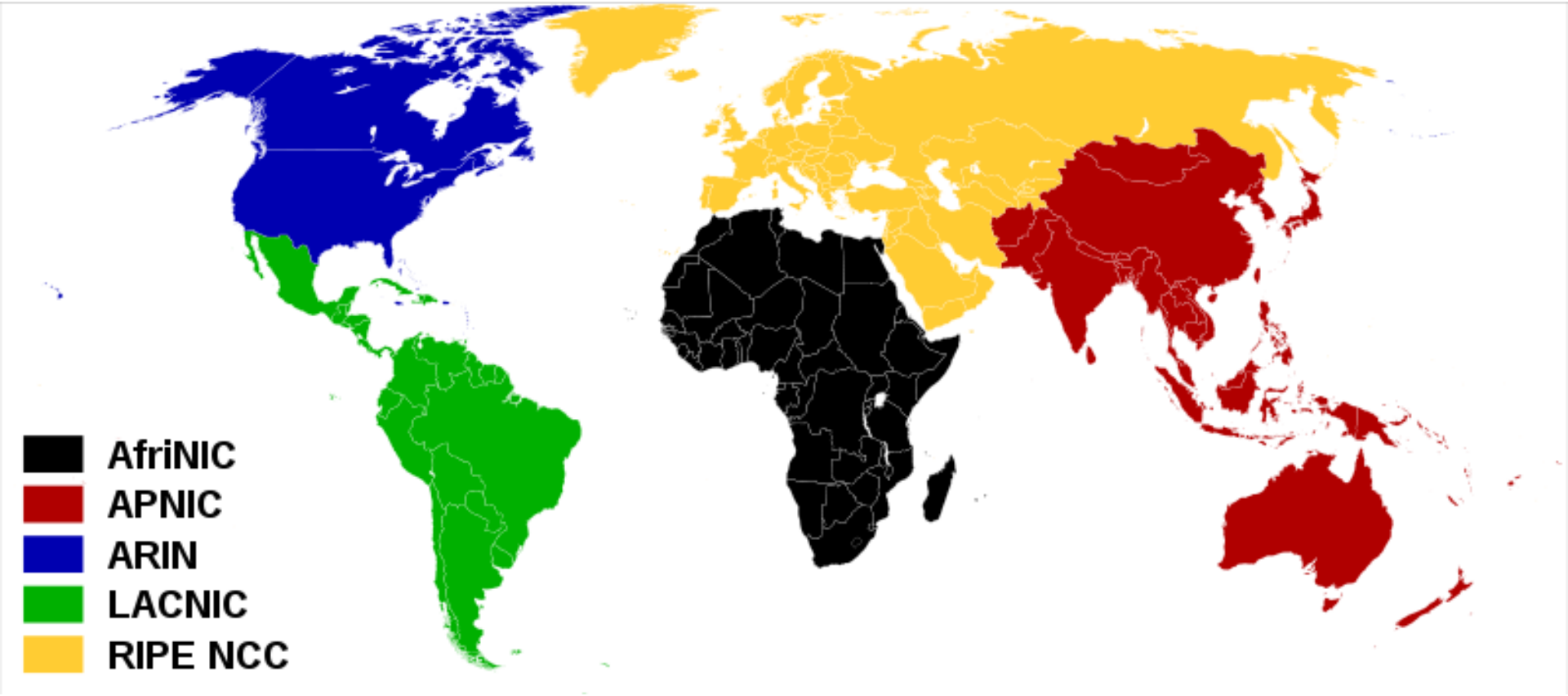
192.168.254.17

Try with:
dig -x 204.152.184.167 +short

Reverse Mapping

- IPv4 addresses are allocated in netblocks by the RIRs ….

RIRs

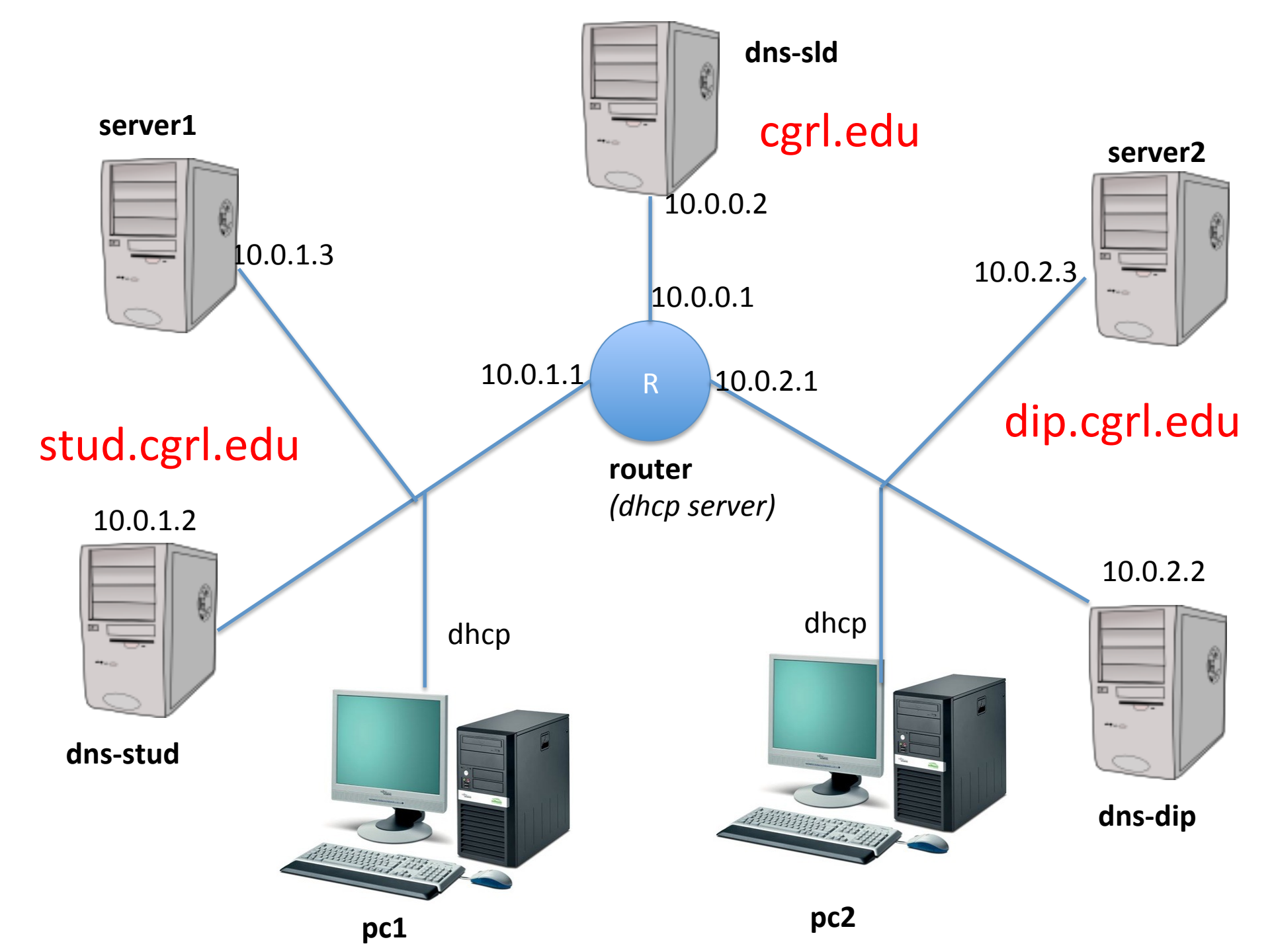


Reverse Mapping

- 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 if static IPv4 addresses are involved

Things are getting serious!

BIND


dns-sld
cgrl.edu
server1
server2
10.0.0.2
10.0.1.3
10.0.2.3
10.0.0.1
10.0.1.1
R
10.0.2.1
stud.cgrl.edu
dip.cgrl.edu
router
(dhcp server)
10.0.1.2
10.0.2.2
dhcp
dhcp
dns-stud
dns-dip
pc1
pc2

/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 we put 127.0.0.1 to test our new dns server!!*

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/
    - 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";
```

- Add a zone for cgrl.com to /etc/bind/db.com.cgrl

Troubleshooting BIND configuration

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

Statements: BIND

- many!
- allow-transfer {192.168.1.2;}; (default yes)
- or selective:

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

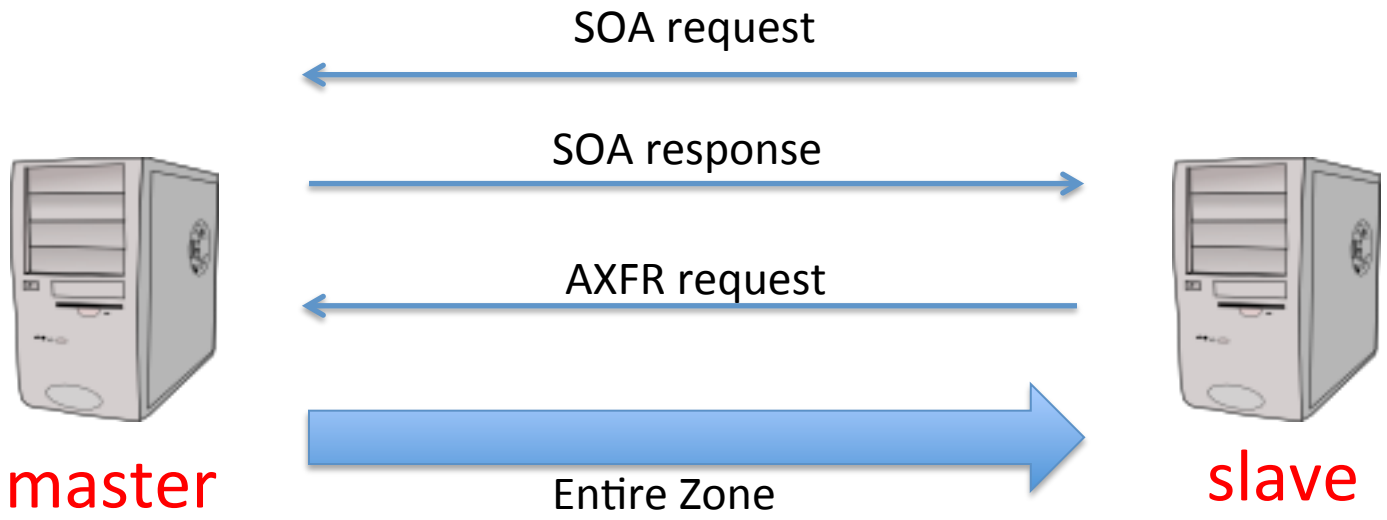
- The allow-notify {192.168.254.2;}; statement disables NOTIFY messages from any host except the zone master to minimize possible malicious action.

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";
};
```

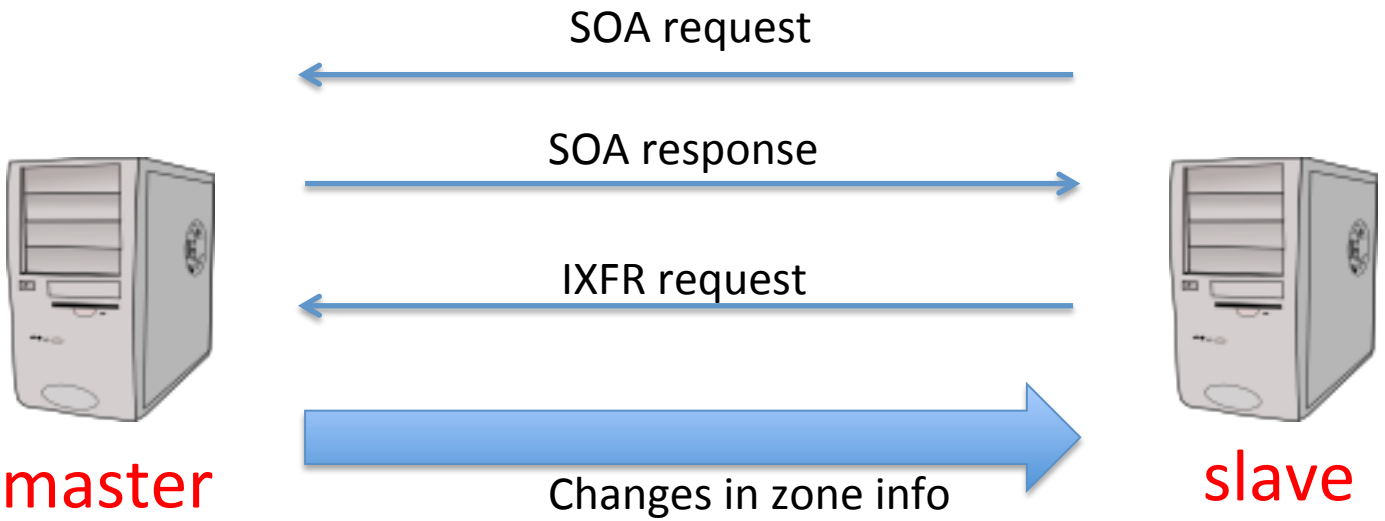
Master Slave configuration: AXFR



Full Zone Transfer

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

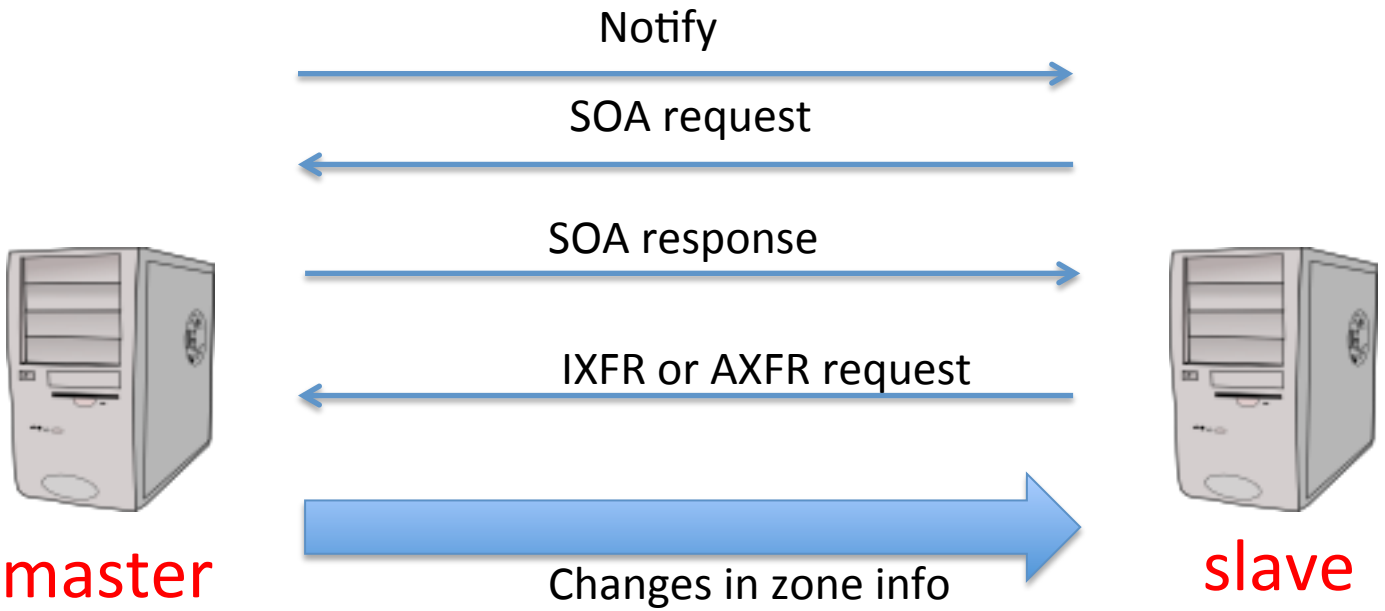
Master Slave configuration: IXFR



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



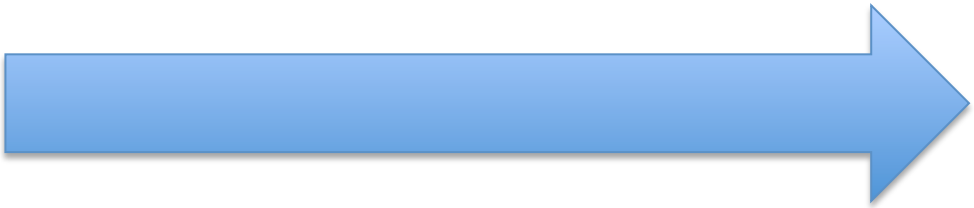
servers can send a NOTIFY message to clients to signal changes

Notify decrease latency and propagation time of zone changes

Example: delegation and redundancy

Master for : example.com
Slave for us.example.com

Master for subdomain:
us.example.com



Delegation of subdomain: us.example.com

Bind: Delegate a Subdomain (Subzone)

```
zone "example.com" in{
    type master;
    file "master.example.com";
}; // optional - example.com acts as the slave (secondary) for the delegated subdomain zone
"us.example.com" IN {
    type slave;
    file "slave.us.example.com";
    masters {10.10.0.24;};
};
```

Delegation with redundancy

Bind: Delegate a Subdomain (Subzone)

```
; zone fragment for 'zone name' example.com is 2 days
$TTL 2d      ; default TTL
 $ORIGIN example.com.
@      .IN SOA ns1.example.com. hostmaster.example.com. (
        2003080800  ; serial number
        12h ; refresh = 12 hours
        15m ; update retry = 15 minutes ;
        3w12h; expiry = 3 weeks + 12 hours
        2h20m ; minimum = 2 hours + 20 minutes
        ; main domain
        ; mail domain
)
; name servers
        IN      NS      ns1.example.com.
        IN      NS      ns2.example.com.
; mail servers
         IN      MX      mail.example.com.
; A records for preceding name servers
ns1    IN      A       192.168.0.3
ns2    IN      A       192.168.0.4
; A record for preceding mail server mail
mail   IN      A       192.168.0.5
```

CONTINUE…

Bind: Delegate a Subdomain (Subzone)

```
; subdomain definitions in the same zone file ;
$ORIGIN us.example.com.
; all subsequent RRs use this ORIGIN
; two name servers for the subdomain
@ IN     NS     ns3.us.example.com.
    IN     NS     ns1.example.com.
```

Do we miss something?

Glue record

- How we can resolve ns3.us.example.com?
  - if that was exactly the dns responsible to resolve *.us.example.com!!
- A glue record is an A record for the name server that is authoritative for the delegated zone.
- Easy! Just add a “glue record”:
  ns3 IN A 10.10.0.24 ; glue record

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.
....
```

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

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.

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

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

Out-of-Sequence Serial Numbers

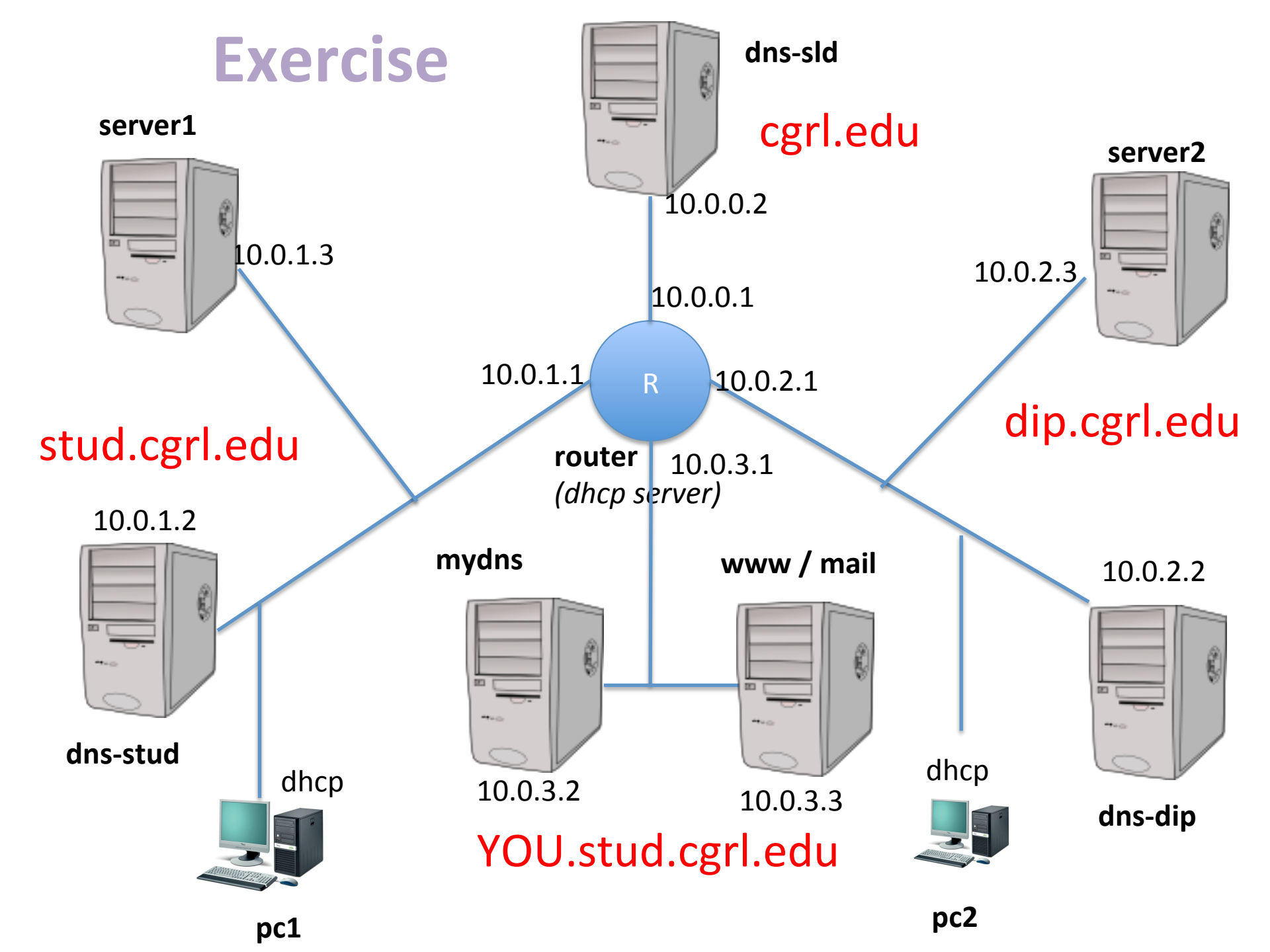
- SN = 4 byte int and setted as a date (convention)
  - bigger SN, newer the data
- what if we make a mistake and put a data in the future?
  - what till the future will come to correct the error
  - increment by 2^31 the value, push to all the slaves, and then put the right value (wrapped through zero)

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
server1
server2
10.0.0.2
10.0.1.3
10.0.0.1
10.0.2.3
10.0.1.1
R
10.0.2.1
stud.cgrl.edu
dip.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