Networks 13: DNS

i.g.batten@bham.ac.uk

What is DNS?

- Absolutely fundamental to Internet: maps names to IP numbers (v4 and v6), numbers to names, locates resources
- One of the oldest protocols in regular use (RFC 882, November 1983, current protocols pretty much as they now are in RFC1034/RFC1035, November 1987)
- Hideously insecure, complex implementations

DNS Concept

From RFC882

```
                    |
       +------------------+------------------+
       |                  |                  |
     COLORS            FLAVORS              TRUTH
       |                  |
 +-----+-----+            |
 |     |     |         NATURAL
RED  BLUE  GREEN          |
                          |
          +---------------+---------------+
          |               |               |
      CHOCOLATE        VANILLA        STRAWBERRY
```

Outline

- Everyone that needs one can get a domain name and operate a nameserver at that point in the hierarchy.
- Once they have that domain name, they can create **resource records** within the domain, to an arbitrary depth
    - Not quite: DNS names are limited to 255 bytes, 63 bytes per label
- They can also **delegate** portions of the namespace to other nameservers
- A group of resource records served from one nameserver is called a **zone**

DNS as Database

- The DNS forms a loosely coupled distributed database, containing key/value pairs
  - Sometimes grossly abused for that purpose, as we will see later
- Lots of caching and redundancy, dating back to a slower, less reliable, less connected Internet
- As a general rule, everyone’s DNS infrastructure is broken, and the definition of “not broken” is the topic of much debate
  - I hope I’m treading a middle-of-the-road position

Resource Records (RR Sets)

- Map a name to some data, plus have some book-keeping data in them.
- Simple case, A records contain IPv4 addresses
- AAAA records contain IPv6 addresses

Name TTL Class Type Data

```
gromit.cs.bham.ac.uk.     86400   IN A      147.188.193.16

research−1.batten.eu.org. 86400  IN AAAA  2001:630:c2:3263:8:20ff:fe89:b5a0
```

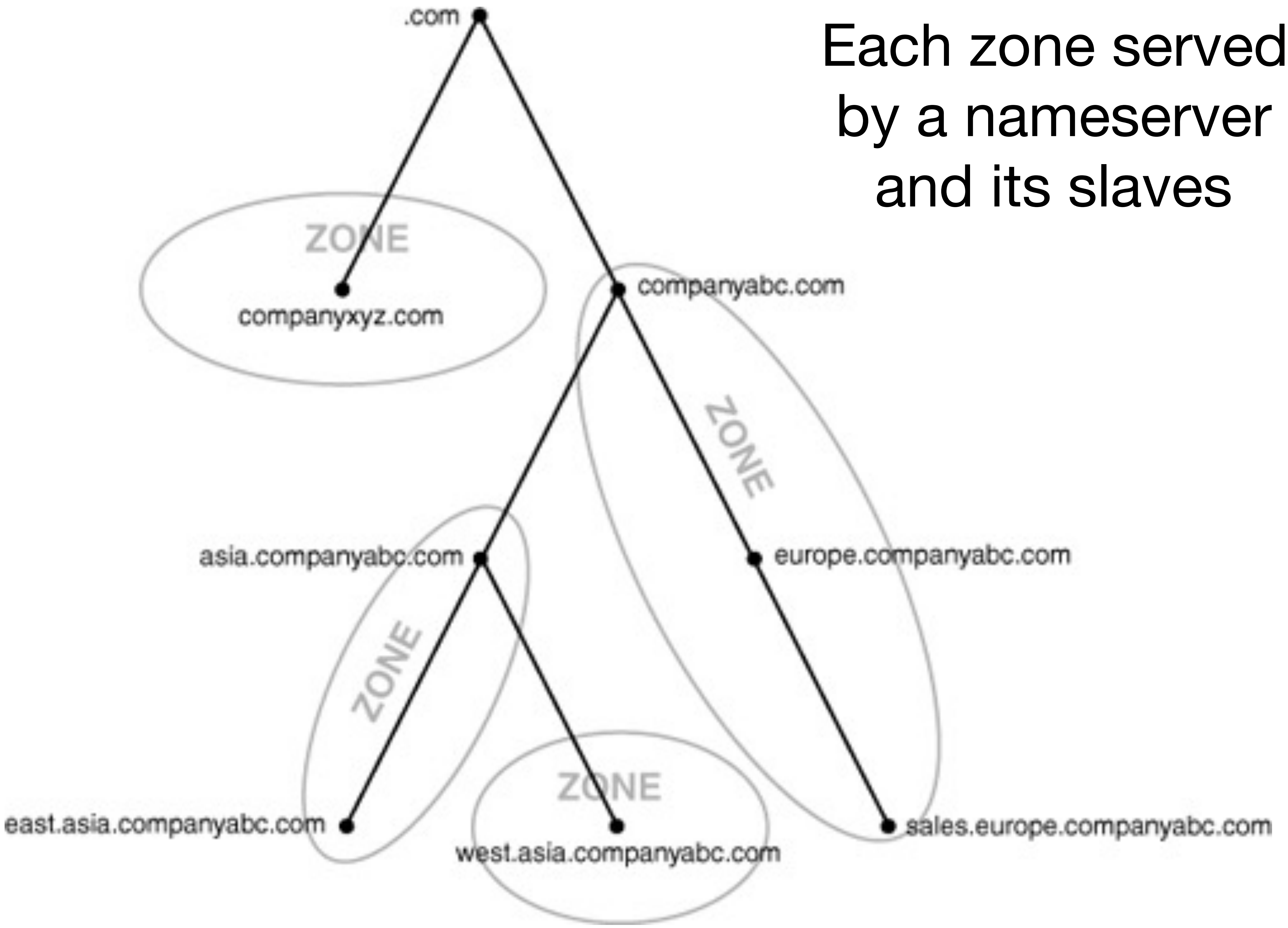
RR Sets

- Class is always IN for the INternet (older classes no longer relevant, wise implementors reject them out of hand)
- Types include A, AAAA, PTR (address->name), MX (mail exchangers), NS (nameservers), CNAME (aliases), SOA (authority records) and TXT (random dumping ground for textual information)
- Can be multiple records for a given name, hence RR **sets.**

TTL

- TTL: “Time to Live”, usually in seconds
  - You quickly learn that 3600 is an hour, 86400 is a day, 604800 is a week.
- You can cache an RR for that long

Zones

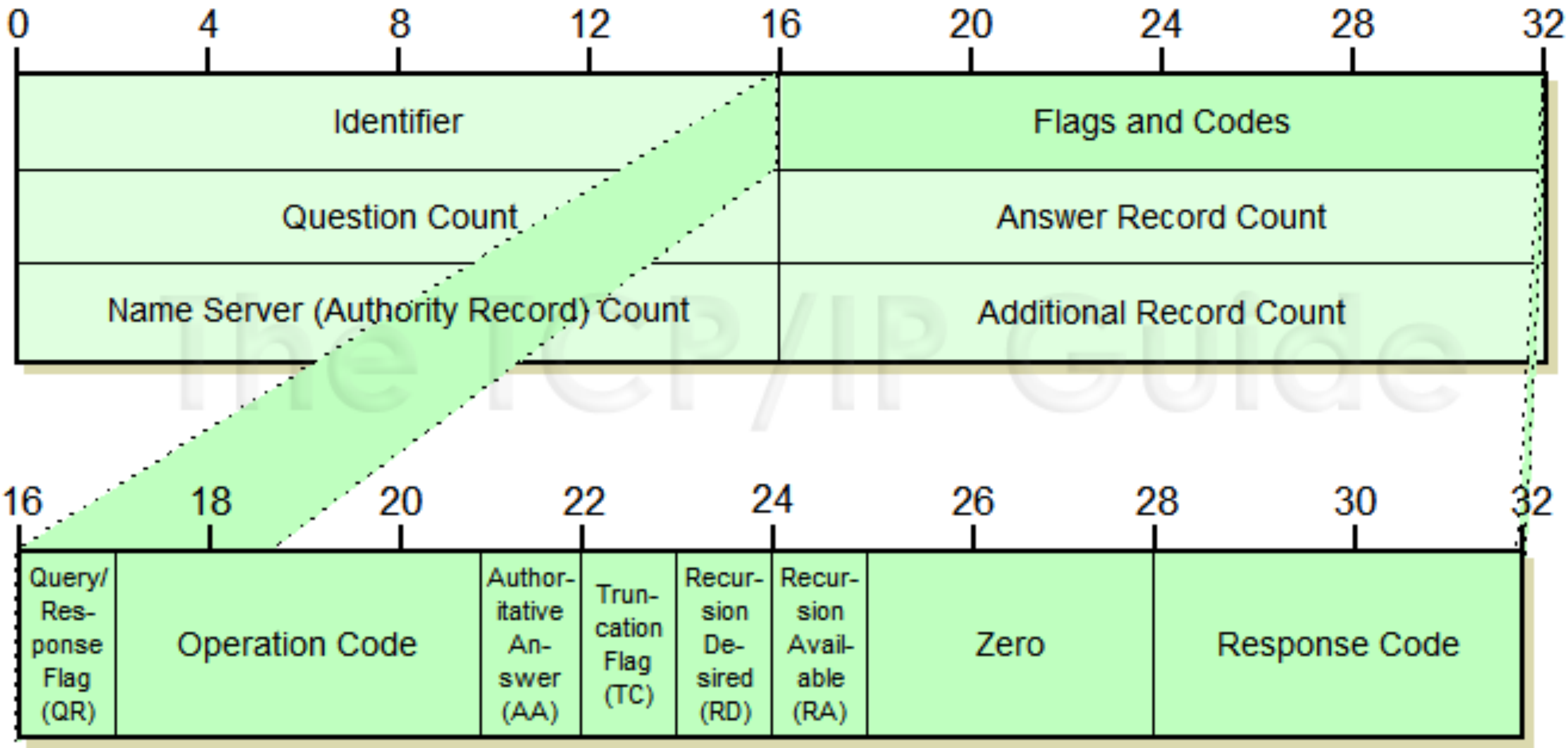


Each zone served by a nameserver and its slaves

DNS Components

- **Clients**: ask questions to **recursive servers** and receive answers
- **Recursive Servers:** (sometimes called **caching servers**) can be asked a complete question (“what is the address of some.machine.dom.ain?”) and will give a complete, but sometimes **inauthoritative** answer
- **Authoritative Servers**: (sometimes called **iterative servers**) only give **authoritative** answers about zones they are configured to know about.
    - **Masters (Primaries)** and **Slaves (Secondaries)** are both **Authoritative**

DNS Packets



Identifier	Flags and Codes
Question Count	Answer Record Count
Name Server (Authority Record) Count	Additional Record Count

Query/ Response Flag (QR)	Operation Code	Authoritative Answer (AA)	Truncation Flag (TC)	Recursion Desired (RD)	Recursion Available (RA)	Zero	Response Code

Header

- ID: sixteen bit identifier for this question and answer. 16 bits massively too small: details later
- QR: 0 is a query, 1 is a response to a query
- Op:
  - 0 is a query (or an answer to a query if QR=1)
  - 1 is an IQUERY (obsolete since 2002)
  - 2 is a STATUS (rarely implemented)
  - 4 is NOTIFY (master informing slave that zone has changed)
  - 5 is UPDATE (dynamic DNS, where used)

Flags

- AA: Authoritative Answer
  - 1 means either I am authoritative, or this answer has freshly come from an authoritative server. 0 means it’s come from a cache.
- TC: Truncation
  - 1 means that there is more than 512 bytes of response, so the client should establish a TCP connection and fetch data that way.
  - If it’s just “additional information” that has been truncated, may not bother. 512 bytes relates to old Arpanet MSS of 560 bytes.

Flags

- RD, RA
  - Recursion Desired means “please answer this question in its entirety”.
  - Recursion Available means “I would do recursion if you asked me for it”, which is worth recording
  - Error Codes:
    - See RFCs

Resource Records

- Name TTL Class Type Data
- foo.domain.mytld is represented as [3] foo [6] domain [5] mytld [0] where [3] is a byte with the value 3, 0x3, 00000011.
- Labels can be compressed

Label Compression

- Labels are maximum 63 bytes, so largest value for a length field is 00111111.

top 2 bits set to 1

- Length fields starting 11 (ie, >= 192) are special. If length field starts 11 (ie length & C0 == C0) then next six bits plus the next byte, total fourteen bits, are special.
- 14 bits are read as “read from this offset in message to next zero”.

Compression

- Suppose [3] foo [3] dom [3] ain [0] starts at byte 48 in the message.
- Suppose we want to represent the name mail.foo.dom.ain.
- Choice 1: [4] mail [3] foo [3] dom [3] ain [0], 18 bytes.
- Choice 2: [4] mail [192] [240], 7 bytes
- Particularly effective as queries about things in domain x.y.z tend to have a lot of x.y.z in them.
- Compression goes to end of name, you can't "reuse" labels like www and mail from other domains.

Clients

- An utter shambles on most operating systems
- DNS clients are also known as “resolvers”
- You can make direct queries to the DNS using the facilities of libresolv (res_mkquery)
- Normally you call getaddrinfo() or gethostbyname() and that chooses mechanism from DNS, local files, LDAP, NIS/YP (is Ronald Reagan still president?) and so on
    - /etc/nsswitch.conf is common, originally Ultrix (I think), then Solaris, now Linux as well — maps queries (hosts, passwd, printers) to sources (DNS, files, NetInfo, whatever)
- Modern systems maintain a cache over all this (nscd on most Unixes).
    - nscd caches tend to be sloppy with TTL handling and cache everything for up to an hour.

Recursive Servers

- Know how to answer a complete question (mechanism to follow)
- Should be access-control and firewall restricted to local network
- Once they have resolved a name they can cache the result, and can answer repeated questions from the cache so long as they appropriately decrement the TTL

Caching In Action

```
ians-macbook-air:clocks igb$ dig aaaa rsync.batten.eu.org

; <<>> DiG 9.8.3-P1 <<>> aaaa rsync.batten.eu.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15509
;; flags: qr aa ra; QUERY: 1, ANSWER: 1, AUTHORITY: 7, ADDITIONAL: 6

;; QUESTION SECTION:
;rsync.batten.eu.org.     IN AAAA

;; ANSWER SECTION:
rsync.batten.eu.org. 86400 IN AAAA 2001:630:c2:3263:8:20ff:fee9:4d41
```

AA = Authoritative Answer

```
ians-macbook-air:clocks igb$ dig aaaa rsync.batten.eu.org

; <<>> DiG 9.8.3-P1 <<>> aaaa rsync.batten.eu.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 232
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;rsync.batten.eu.org.     IN AAAA

;; ANSWER SECTION:
rsync.batten.eu.org. 86245 IN AAAA 2001:630:c2:3263:8:20ff:fee9:4d41
```

!AA

Full Initial Response

```
ians-macbook-air:clocks igb$ dig aaaa rsync.batten.eu.org

; <<>> DiG 9.8.3-P1 <<>> aaaa rsync.batten.eu.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15509
;; flags: qr aa ra; QUERY: 1, ANSWER: 1, AUTHORITY: 7, ADDITIONAL: 6

;; QUESTION SECTION:
;rsync.batten.eu.org.    IN AAAA

;; ANSWER SECTION:
rsync.batten.eu.org.  86400 IN AAAA  2001:630:c2:3263:8:20ff:fee9:4d41

;; AUTHORITY SECTION:
batten.eu.org.    86400 IN NS offsite6.batten.eu.org.
batten.eu.org.    86400 IN NS ns5.he.net.
batten.eu.org.    86400 IN NS ns3.he.net.
batten.eu.org.    86400 IN NS ns2.he.net.
batten.eu.org.    86400 IN NS mail.batten.eu.org.
batten.eu.org.    86400 IN NS ns4.he.net.
batten.eu.org.    86400 IN NS offsite7.batten.eu.org.

;; ADDITIONAL SECTION:
offsite6.batten.eu.org.  86400 IN A  128.204.195.144
offsite6.batten.eu.org.  86400 IN AAAA  2a00:7b80:3019:12::579c:4928
mail.batten.eu.org.   300IN A  147.188.192.250
mail.batten.eu.org.   300IN AAAA  2001:630:c2:3263:8:20ff:fed7:92ef
offsite7.batten.eu.org.  86400 IN A  64.188.45.237
offsite7.batten.eu.org.  86400 IN AAAA  2607:f2e0:10f:14:4321:4321:5e6:9ee6
```

“Here’s where I got it from, these are the sources of truth”

These might be useful

Cached Response

```
ians-macbook-air:clocks igb$ dig aaaa rsync.batten.eu.org

; <<>> DiG 9.8.3-P1 <<>> aaaa rsync.batten.eu.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 232
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;rsync.batten.eu.org.		IN	AAAA

;; ANSWER SECTION:
rsync.batten.eu.org.	86245	IN	AAAA	2001:630:c2:3263:8:20ff:fee9:4d41

;; Query time: 80 msec
;; SERVER: 147.188.129.250#53(147.188.129.250)
;; WHEN: Tue Feb 24 09:33:38 2015
;; MSG SIZE  rcvd: 65
```

How are names resolved?

- Clients ask recursive servers (“rd” = “recursion desired”)
- Recursive servers start at the root, and work downwards asking for the nameserver of the next label down, until they can finally ask the last nameserver for the required RR

Flow of packets

- Recursive nameserver is pre-configured with addresses of nameservers for “.”, the root of the domain space.
- To answer a query for “A foo.dom.ain”, asks one of the root nameservers “NS? foo.dom.ain”.
- Will get back either the location of the nameservers for “ain”, or “dom.ain”, or “foo.dom.ain”, depending on what knowledge the server has.
- Can then ask next server down the same question, until someone answers with an A record.
  - Called “iteration”, although actually feels somewhat recursive.

Command Line Example

```
ians-macbook-air:~ igb$ dig +norecurse @a.root-servers.net. ns uk | awk '$4=="NS"' | head -1
uk.        172800 IN NS nsd.nic.uk.
ians-macbook-air:~ igb$ dig +norecurse @nsd.nic.uk. ns ac.uk | awk '$4=="NS"' | head -1
ac.uk.       172800 IN NS ns1.surfnet.nl.
ians-macbook-air:~ igb$ dig +norecurse @ns1.surfnet.nl. ns bham.ac.uk | awk '$4=="NS"' |
head -1
bham.ac.uk.   86400  IN NS dns0.bham.ac.uk.
ians-macbook-air:~ igb$ dig +norecurse @dns0.bham.ac.uk. www.bham.ac.uk
```

You can experiment with this: sometimes the answer is returned as an answer, sometimes as authority records, depending the precise configuration of the server for the zone you are querying.

Command Line Example

```
ians-macbook-air:~ igb$ dig +norecurse @dns0.bham.ac.uk. www.bham.ac.uk

; <<>> DiG 9.8.3-P1 <<>> +norecurse @dns0.bham.ac.uk. www.bham.ac.uk
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24883
;; flags: qr aa ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;www.bham.ac.uk.         IN A

;; ANSWER SECTION:
www.bham.ac.uk.    172800 IN CNAME  corp501.bham.ac.uk.
corp501.bham.ac.uk.   172800 IN A  147.188.125.39

;; AUTHORITY SECTION:
bham.ac.uk.    172800 IN NS dns1.bham.ac.uk.
bham.ac.uk.    172800 IN NS dns0.bham.ac.uk.

;; ADDITIONAL SECTION:
dns0.bham.ac.uk.   172800 IN A  147.188.128.2
dns1.bham.ac.uk.   172800 IN A  147.188.128.102
```

Caching

- At each stage, everyone caches the data they get
- Particularly, the recursive server will rapidly build a cache of the NS records for popular domains, only refreshed once per TTL seconds
- In original protocol, anyone can supply cached information about anything and be believed
- Modern systems are more sceptical

Authoritative Servers

- For a given domain, there will be one or more nameservers specified in some zone logically closer to the root (not necessarily one node up).
- Each of these specified nameservers is **authoritative** for the specified domain.
- The NS record that points to them is called a **delegation** in the **parent.**

Authoritative Servers

- Why multiple servers?
  - Load balancing and redundancy
  - Ideally, spread over the world, spread over multiple ASes
  - Anycasting is useful: a domain with just one NS record may be using it
- They are all equally authoritative: no concept of “master” or “slave” is exposed
- They hold “zone files” for the zone files they are authoritative for
  - Either as real files, or as “zones” within the server software

Authority Record

hostmaster is determine the master

```
cs.bham.ac.uk.    86400 IN SOA dns0.cs.bham.ac.uk. hostmaster.cs.bham.ac.uk. (
          2015022000 ; serial
          10800      ; refresh (3 hours)
          3600       ; retry (1 hour) if unsucc retry again
          604800     ; expire (1 week)
          86400      ; minimum (1 day)
          )
```

Serial, Refresh, Retry and Expire for benefit of slaves

Minimum now redefined for negative caching

NS record

match parent zone to child zone

Should match in bham.ac.uk and cs.bham.ac.uk zones

```
cs.bham.ac.uk.    86400 IN  NS dns1.cs.bham.ac.uk.
cs.bham.ac.uk.    86400 IN  NS dns0.cs.bham.ac.uk.
cs.bham.ac.uk.    86400 IN  NS dns0.bham.ac.uk.
cs.bham.ac.uk.    86400 IN  NS ns0.susx.ac.uk.
cs.bham.ac.uk.    86400 IN  NS ext-proxy.ftel.co.uk.
```

Additional Information

```
dns0.cs.bham.ac.uk. 86400   IN A 147.188.192.4
dns1.cs.bham.ac.uk. 86400   IN A 147.188.192.8
ext-proxy.ftel.co.uk. 86400   INA  192.65.220.99
ext-proxy.ftel.co.uk. 86400   INA  192.65.220.98
```

Note: incomplete

Old Attack

- Old nameserver software blindly accepted additional information and cached it
- Allows you to supply an IP number you control as "google.com 604800 IN A 1.2.3.4"; anyone who visits your nameserver has a chance of caching a bad nameserver for Google (or a bank, or whatever).
- Now stopped with "out of balliwick" controls: you only accept additional information that the server can reasonably be assumed to be authoritative for

Master / Slave

- DNS protocol has support for replicating zones between master and slaves
    - Master does not have to be visible, “hidden master” is a popular pattern
    - Multiple nameservers can be updated by other means (SQL replication, rsync, people carrying USB sticks)

Pro Tip

- A common pattern on small networks is for the authoritative server to also be the recursive / caching server for local clients
- **DO NOT DO THIS.**

Delegation

- Suppose we have a nameserver for batten.eu.org. How do we create the domain home.batten.eu.org on another nameserver (or at least another zone file, possibly with different access rules)?

Delegation: Just NS records

In batten.eu.org

```
home.batten.eu.org. 86400  IN  NS   pi-two.home.batten.eu.org.
home.batten.eu.org. 86400  IN  NS   dns-2.batten.eu.org.
home.batten.eu.org. 86400  IN  NS   pi-one.home.batten.eu.org.

;; ADDITIONAL SECTION:
dns-2.batten.eu.org.  86400  IN A 64.188.45.237
dns-2.batten.eu.org.  86400  IN AAAA 2607:f2e0:10f:14:4321:4321:5e6:9ee6
pi-one.home.batten.eu.org. 86400 IN   A  10.92.213.231
pi-one.home.batten.eu.org. 86400 IN   AAAA  2001:8b0:129f:a90f:3141:5926:5359:1
pi-two.home.batten.eu.org. 86400 IN   A  10.92.213.238
pi-two.home.batten.eu.org. 86400 IN   AAAA  2001:8b0:129f:a90f:3141:5926:5359:2
```

Glue Records

- How do you locate the A record for “dom.ain 86400 IN NS ns1.dom.ain”?
- The zonefile dom.ain in that case has an A record for ns1.dom.ain as well as an NS record for dom.ain.
- This is called a **Glue Record**

Mail Exchangers

in order to send email to this domain, check the server?

```
;; ANSWER SECTION:
batten.eu.org.    86400   IN MX 4 bham-mx2.bham.ac.uk.
batten.eu.org.    86400   IN MX 4 bham-mx3.bham.ac.uk.
batten.eu.org.    86400 IN   MX 2 mail.batten.eu.org.
batten.eu.org.    86400   IN MX 4 bham-mx1.bham.ac.uk.

;; ADDITIONAL SECTION:
bham-mx2.bham.ac.uk. 86350 IN A  147.188.128.219
bham-mx3.bham.ac.uk. 86350 IN A  147.188.128.221
mail.batten.eu.org.  300   IN A  147.188.192.250
mail.batten.eu.org.  300   IN AAAA 2001:630:c2:3263:8:20ff:fed7:92ef
bham-mx1.bham.ac.uk. 86350 IN A  147.188.128.129
```

Zone File Maintenance

- You can edit zone-files manually, but it is very prone to error
- Most sites generate the zone files from some other source of information, usually a database or some XML (classic “greybeard” shell scripts, which scare everyone once the author leaves)
- Also dynamic DNS

Dynamic Update

```
ians-macbook-air:~ igb$ nsupdate -k update-key
> server offsite7.batten.eu.org
> update add some-spurious-rrset.batten.eu.org 86400 in a 1.2.3.4
>
> ians-macbook-air:~ igb$ dig @offsite7.batten.eu.org some-spurious-rrset.batten.eu.org

; <<>> DiG 9.8.3-P1 <<>> @offsite7.batten.eu.org some-spurious-rrset.batten.eu.org
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64938
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 7, ADDITIONAL: 13

;; QUESTION SECTION:
;some-spurious-rrset.batten.eu.org. IN   A

;; ANSWER SECTION:
some-spurious-rrset.batten.eu.org. 86400 IN A   1.2.3.4
```

Works worldwide!

```
ians-macbook-air:~ igb$ dig @8.8.8.8 some-spurious-rrset.batten.eu.org.

; <<>> DiG 9.8.3-P1 <<>> @8.8.8.8 some-spurious-rrset.batten.eu.org.
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15394
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;some-spurious-rrset.batten.eu.org. IN A

;; ANSWER SECTION:
some-spurious-rrset.batten.eu.org. 21599 IN A 1.2.3.4
```

Dynamic Update

- Useful for DHCP servers in trusting environments
    - You can update the DNS to reflect equipment entering and leaving the network
- Allows anyone on the network to pretend to be anyone else by careful use of DHCP client ids

Reverse Mapping

- IP number is transformed into reverse order and looked up in special in-addr.arpa or .ip6.arpa domain:

```
250.192.188.147.in-addr.arpa. 86400 IN PTR   offsite.batten.eu.org.
```

```
1.0.0.0.9.5.3.5.6.2.9.5.1.4.1.3.e.0.9.a.f.9.2.1.0.b.8.0.1.0.0.2.ip6.arpa. 86400  IN PTR pi-one.batten.eu.org.
```

Why the strange formats?

- in-addr.arpa format allows delegation on 8-bit boundaries.
- Makes delegation of /28s (say) difficult
- Various messy solutions: look them up.
- Lesson learnt, so ipv6 reverse mapping allows delegation on 4-bit boundaries in hierarchy.

Reverse Attacks

- Suppose I control 2.3.4.in-addr.arpa
- I can create RR “1.2.3.4.in-addr.arpa PTR something.bham.ac.uk” and try to pose as part of Birmingham network for purposes of libraries, Apple discounts, etc.
- Check is to look up something.bham.ac.uk and see if it matches: only bham.ac.uk admin can install required “something.bham.ac.uk A 1.2.3.4” record

DNS Security

- DNS Sec exists to sign zones, providing evidence that packets haven’t been tampered with
- Topic for network security lectures to come
  - Complex
  - Didn’t scale without major modifications
  - Very low adoption after ~20 years
  - Doesn’t solve common use-cases