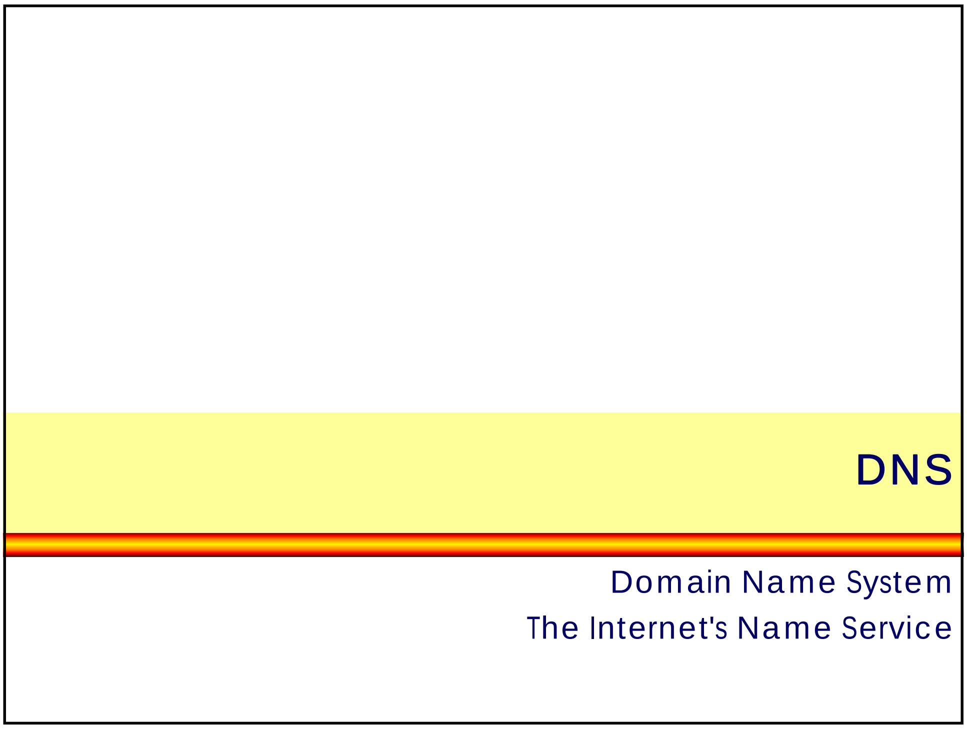
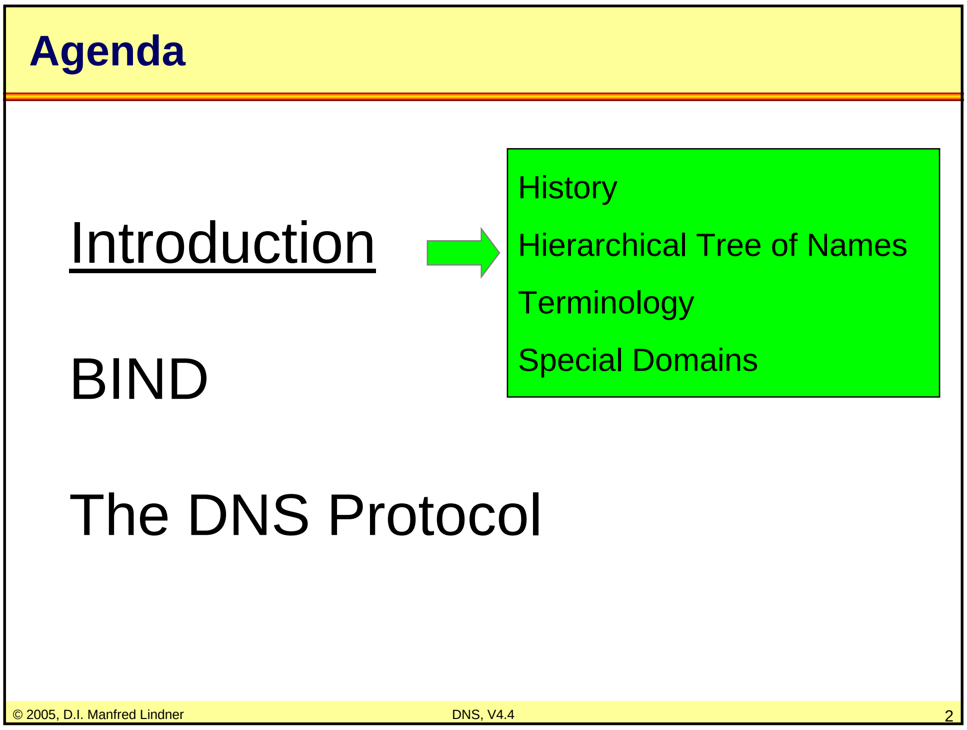


DNS

Domain Name System
The Internet's Name Service

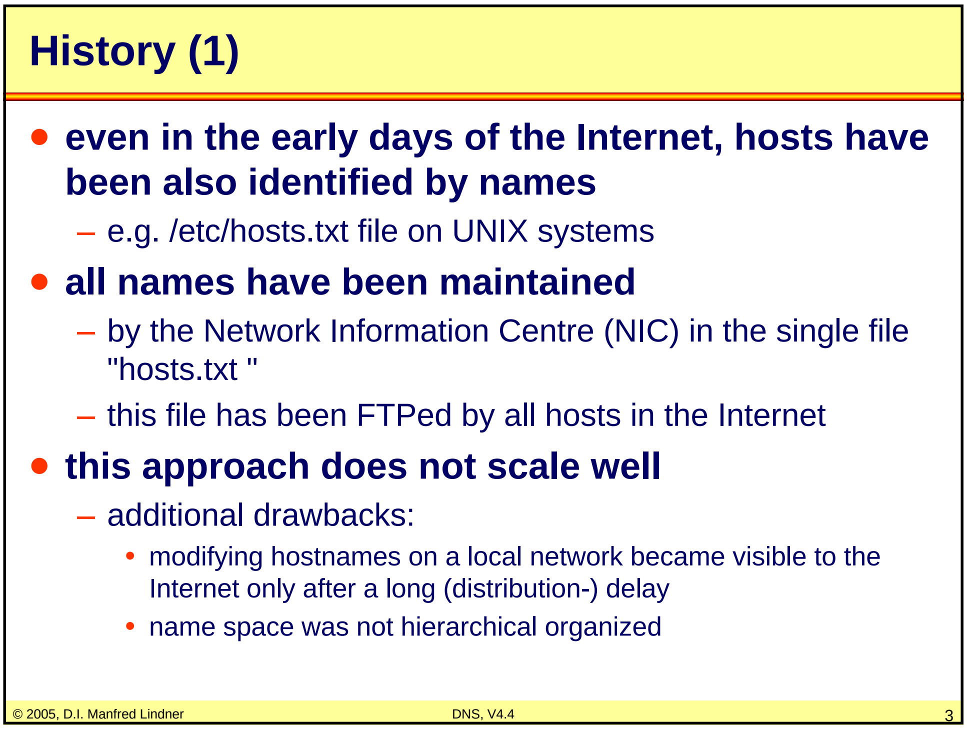


Agenda

Introduction → History, Hierarchical Tree of Names, Terminology, Special Domains

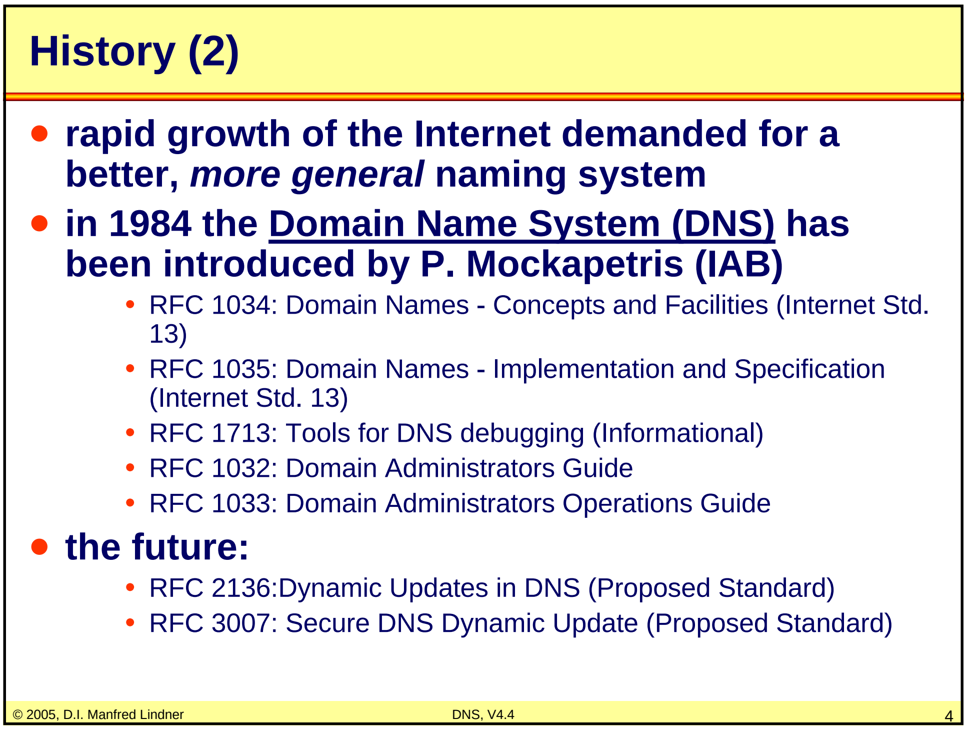
BIND

The DNS Protocol



History (1)

- **even in the early days of the Internet, hosts have been also identified by names**
  - e.g. /etc/hosts.txt file on UNIX systems
- **all names have been maintained**
  - by the Network Information Centre (NIC) in the single file "hosts.txt "
  - this file has been FTPed by all hosts in the Internet
- **this approach does not scale well**
  - additional drawbacks:
    - modifying hostnames on a local network became visible to the Internet only after a long (distribution-) delay
    - name space was not hierarchical organized



History (2)

- **rapid growth of the Internet demanded for a better, *more general* naming system**
- **in 1984 the Domain Name System (DNS) has been introduced by P. Mockapetris (IAB)**
    - RFC 1034: Domain Names - Concepts and Facilities (Internet Std. 13)
    - RFC 1035: Domain Names - Implementation and Specification (Internet Std. 13)
    - RFC 1713: Tools for DNS debugging (Informational)
    - RFC 1032: Domain Administrators Guide
    - RFC 1033: Domain Administrators Operations Guide
- **the future:**
    - RFC 2136:Dynamic Updates in DNS (Proposed Standard)
    - RFC 3007: Secure DNS Dynamic Update (Proposed Standard)

Mnemonic Approach

- **Problem: the 32-bit IP address-format encodes 2^32 single addresses (4 294 967 296)**
  - theoretically (!) – many of them have been wasted
  - how to build an effective directory for such a huge number of hosts?
- **Solution:**
  - hierarchy of simple, mnemonic names: *Domain Names*

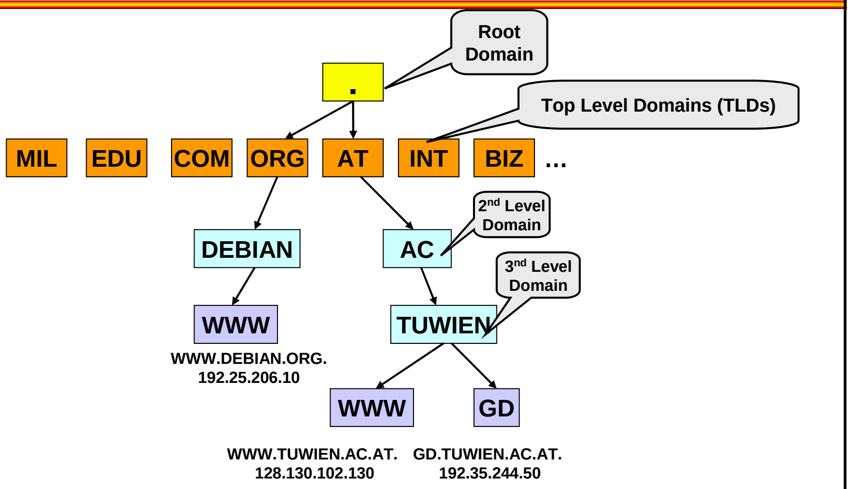
    e.g. instead of remembering all IP addresses from 216.32.74.50 to 216.32.74.55, it is sufficient to know "www.yahoo.com"
- **Why is the Internet so convenient to use?**
  - domain names can be *guessed* and *bookmarked*

    and of course search engines do the rest...

What Basically Does DNS ?

- **DNS "replaces" the IP address of hosts to a human readable format**
  - DNS enables a mapping between names and addresses
  - often called "hostname resolution"
  - due to its size DNS is a world-wide *distributed* database
- **DNS assigns hosts to a tree-like directory hierarchy**
  - each part of the hierarchy is called a "domain“, each hierarchy level is assigned a label, called "domain name"
  - the Domain Name Tree does NOT reflect the physical network structure !!!

Tree of Names



Compare this DNS tree with a file directory tree of a common Operating Systems where **C:\at\ac\tuwien\www\ip_address.txt** is used to specify the location of the file ip_address.txt on the hard disk

Name Servers - DNS Resolver

- **the DNS tree is realized by**
  - Name Servers
- **each Name Server take cares**
  - for a subset of the DNS tree
  - so called “zones”
- **the physical location of name server**
  - has nothing to do with the DNS tree
- **if an IP host wants to resolve a symbolic name**
  - resolver software acting as DNS client will ask a DNS name server using the DNS protocol
  - IP address of name server either manually configured or known through DHCP or explicitly specified by the user

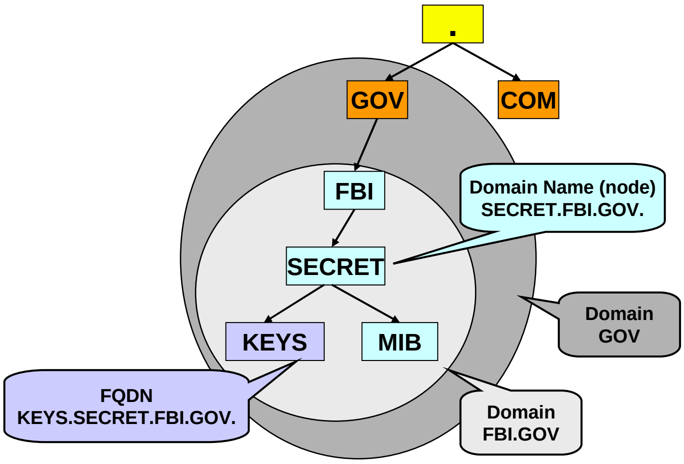
Conventions (1)

- Terminology**: a "Domain" ...**
  - is a complete sub-tree
    - everything under a particular point in the tree
  - relates to the naming structure itself, not the way things are distributed
- Terminology**: a "Domain Name" ...**
  - is the name of a node in the tree (domain, host, ...)
  - consists of all concatenated labels from the root to the current domain, listed from right to left, separated by dots
    - max 255 characters

Conventions (2)

- Terminology**: a "Label" ...**
  - is a component of the domain name
  - need only be unique at a particular point in the tree
    - that is, both "name.y.z" and "name.x.y.z" are allowed
    - max 63 characters
  - DNS is not case sensitive !
    - "www.nic.org" is the same as "WWW.NIC.ORG"
  - due to SMTP restrictions, domain names may contain only characters of {a-z, A-Z, 0-9, "-"}
    - there are some new conventions concerning national characters
- Terminology**: a "Fully Qualified Domain Name“ (FQDN)**
  - concatenation of all labels of including trailing dot **". "**

Example for Terminology



.

GOV

COM

FBI

SECRET

KEYS

MIB

Domain Name (node)
SECRET.FBI.GOV.

Domain
GOV

FQDN
KEYS.SECRET.FBI.GOV.

Domain
FBI.GOV

Conventions (3)

- **hosts with *multiple* network addresses can be assigned a *single* domain name**

  e.g. routers, servers with several network interfaces, ...

- **hosts with a *single* IP address can be assigned *multiple* domain names**

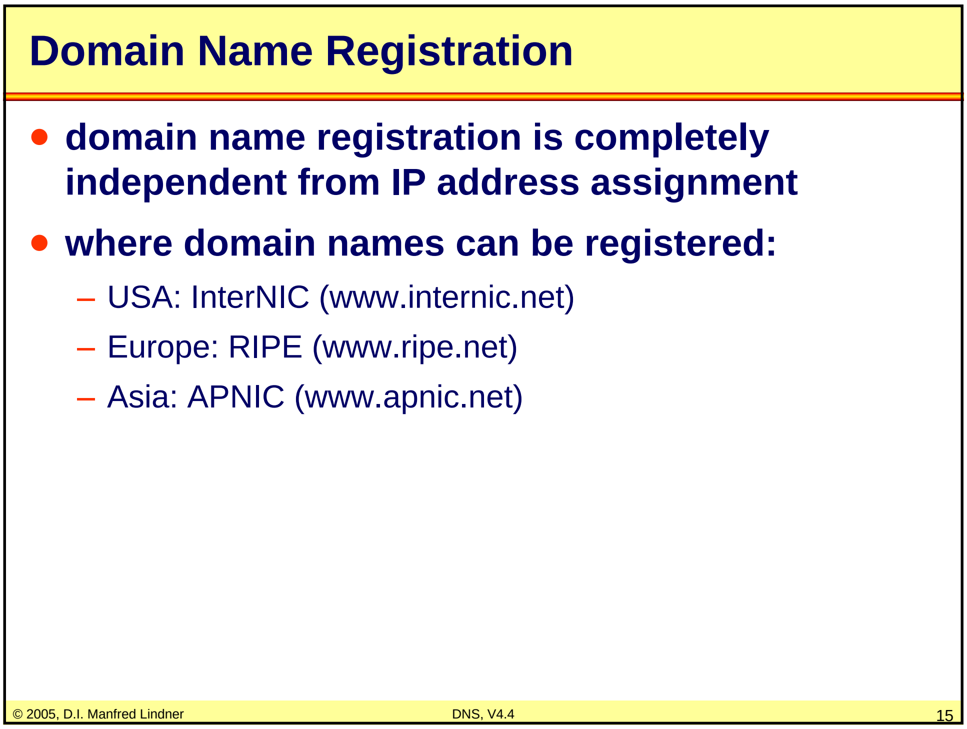
  e.g. to differentiate several services: **www**.x.y.z, **ftp**.x.y.z, **mail**.x.y.z, ...

The Root Domain

- **the root of the DNS tree is denoted as a single dot "."**
  - each domain name without this root-dot is only a relative domain name
    - although, most applications do not follow this rule
    - but essential in BIND configuration files (master files)
  - otherwise it is a Fully Qualified Domain Name (FQDN) which exactly identifies a single host from all hosts in the world
- **the root is implemented by *several* root-servers**
  - name server at the highest hierarchy level
- **below the root, a domain may be called top-level, second-level, third-level etc...**

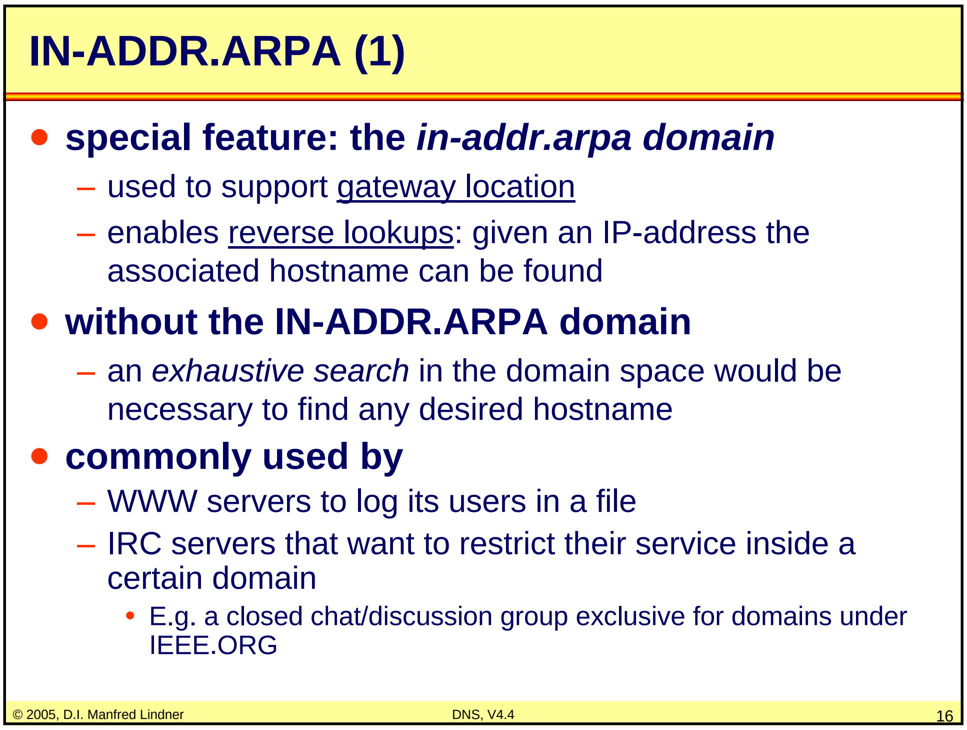
Top Level Domains (RFC1591)

- **inside US: "generic domains"**
  - **com** - Commercial
  - **edu** - Educational
  - **org** - Non Profit Organizations (NPOs)
  - **net** - Networking providers
  - **mil** - US military
  - **gov** - US goverment
  - **int** - International organizations
- **outside US: two letter country code**
  - defined in ISO-3166
  - examples: **uk** (United Kingdom), **fr** (France), **us** (United States), **de** (Germany), **at** (Austria), **ax** (Antarctica)
  - Note: country code does not reflect real location !



Domain Name Registration

- **domain name registration is completely independent from IP address assignment**
- **where domain names can be registered:**
  - USA: InterNIC (www.internic.net)
  - Europe: RIPE (www.ripe.net)
  - Asia: APNIC (www.apnic.net)



IN-ADDR.ARPA (1)

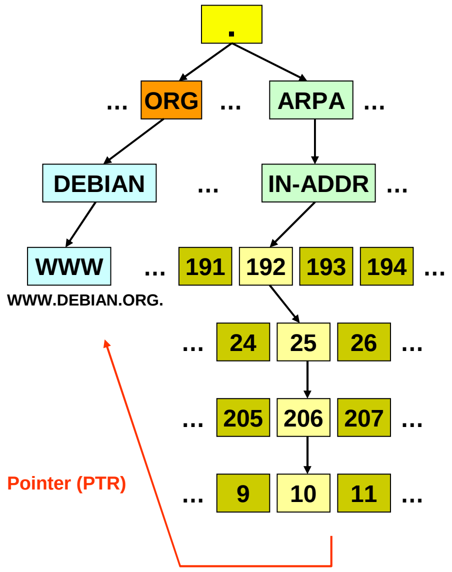
- **special feature: the *in-addr.arpa domain***
  - used to support gateway location
  - enables reverse lookups: given an IP-address the associated hostname can be found
- **without the IN-ADDR.ARPA domain**
  - an *exhaustive search* in the domain space would be necessary to find any desired hostname
- **commonly used by**
  - WWW servers to log its users in a file
  - IRC servers that want to restrict their service inside a certain domain
    - E.g. a closed chat/discussion group exclusive for domains under IEEE.ORG

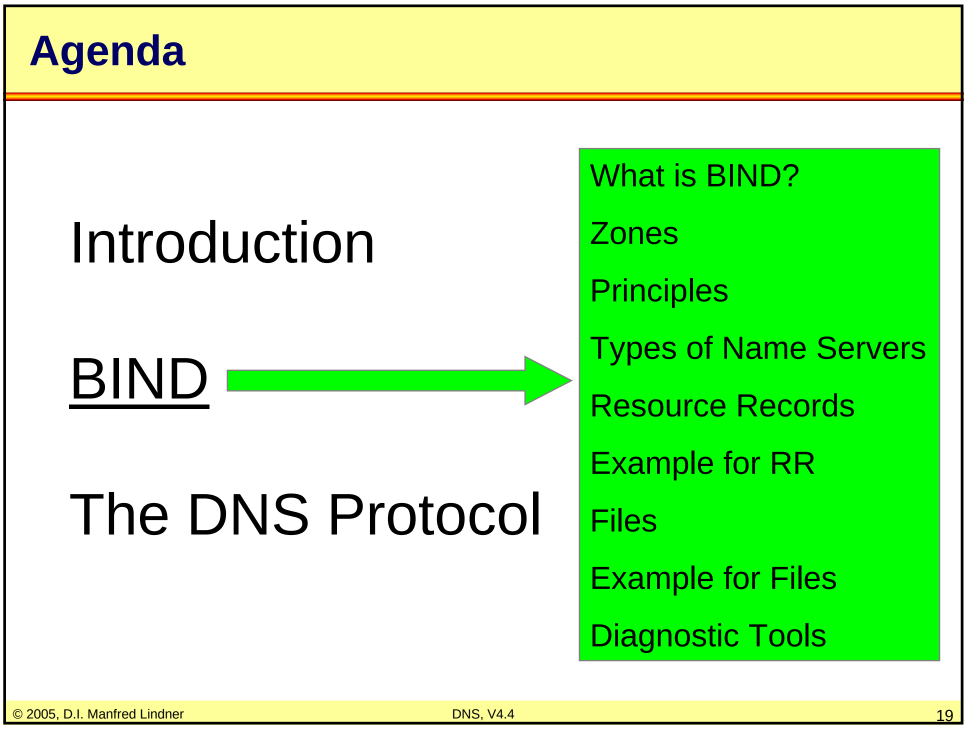
IN-ADDR.ARPA (2)

- **the domain in-addr.arpa is structured according to the IP address**
  - this special domain begins at "IN-ADDR.ARPA"
  - its substructure follows the Internet addressing structure
- **each domain name has up to 4 additional labels**
  - each label represents one octet of the IP address
    - expressed as character string for its decimal value ("0" - "255")
    - the reverse host/domain names are organized on byte boundaries
  - Note: labels are attached to the suffix in reverse order
    - e.g. data for internet address 216.32.74.50 is found at 50.74.32.216.IN-ADDR.ARPA
    - hosts have all four labels specified

IN-ADDR.ARPA (3)

What's the Domain Name of 192.25.206.10 ?





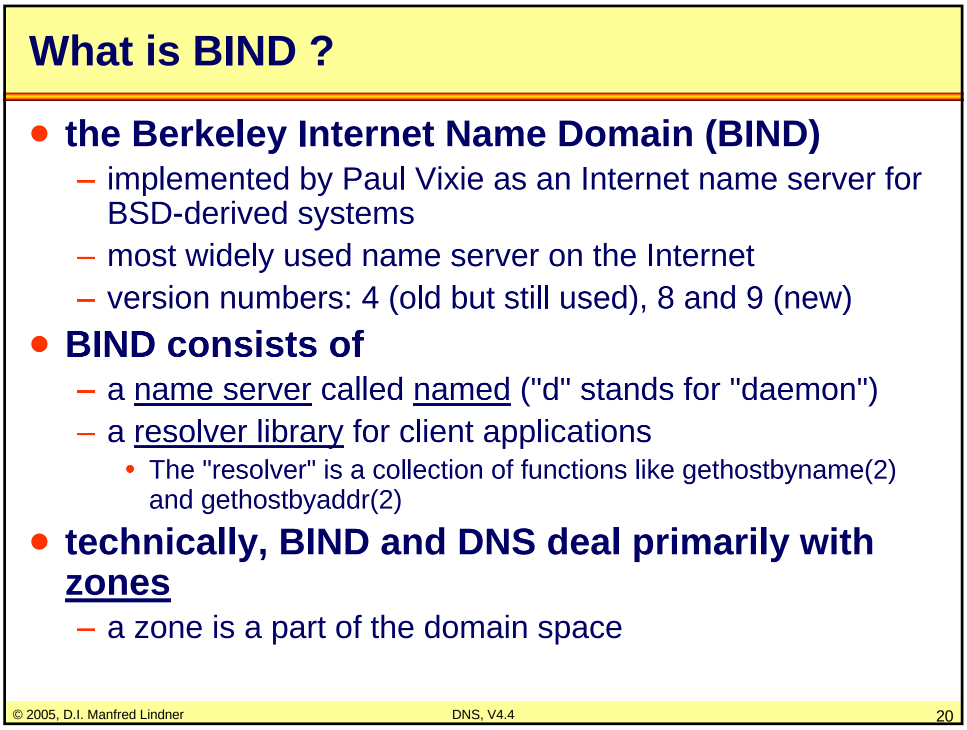
Agenda

Introduction

BIND

The DNS Protocol

- What is BIND?
- Zones
- Principles
- Types of Name Servers
- Resource Records
- Example for RR
- Files
- Example for Files
- Diagnostic Tools



What is BIND ?

- **the Berkeley Internet Name Domain (BIND)**
  - implemented by Paul Vixie as an Internet name server for BSD-derived systems
  - most widely used name server on the Internet
  - version numbers: 4 (old but still used), 8 and 9 (new)
- **BIND consists of**
  - a name server called named ("d" stands for "daemon")
  - a resolver library for client applications
    - The "resolver" is a collection of functions like gethostbyname(2) and gethostbyaddr(2)
- **technically, BIND and DNS deal primarily with zones**
  - a zone is a part of the domain space

What is a Zone ?

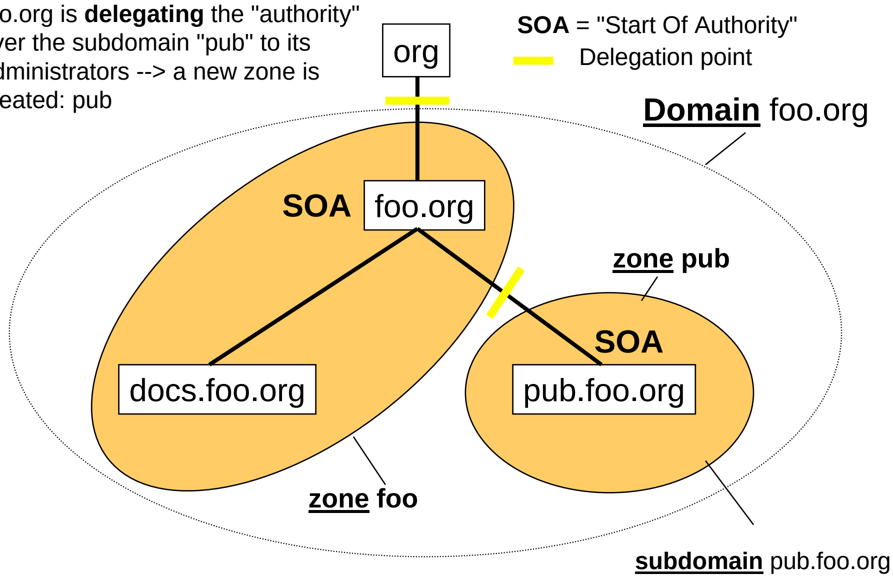
- a zone is a "point of delegation"
  - contains all names from this point downwards the domain-tree except those which are delegated to other zones (to other name servers)
  - a zone can span over a whole domain or just be part of it
- *in other words:* a zone is a pruned domain !
  - pruning occurs when zones are delegated
  - zones relate to the way the database is partitioned and distributed
- a name server is ***authoritative*** over a domain
  - if he keeps a *master file* (zone file) with information of that domain

Zones and SOA

foo.org is **delegating** the "authority" over the subdomain "pub" to its administrators --> a new zone is created: pub

SOA = "Start Of Authority"

▬ Delegation point

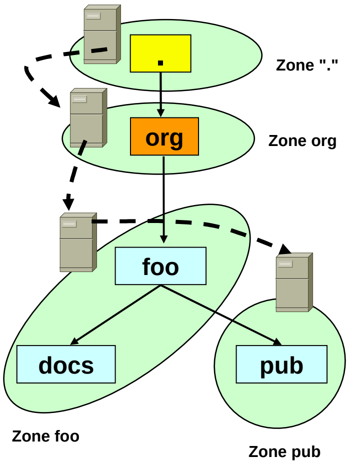


Delegation and Name Servers

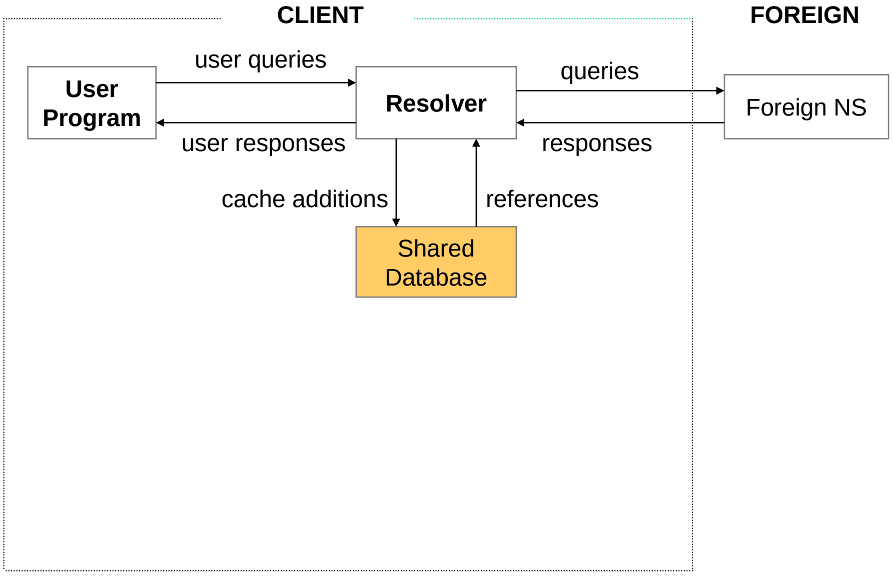
Root Server responsible for root domain delegates authority for building symbols org. to NS below which holds the master file for zone org

NS responsible for domain org delegates authority for building symbols foo.org. to NS below which holds the master file for zone foo

NS responsible for domain foo.org delegates authority for building symbols pub.foo.org. to NS below which holds the master file for zone pub



BIND Principles (Client)



CLIENT

FOREIGN

User Program

user queries

user responses

Resolver

queries

responses

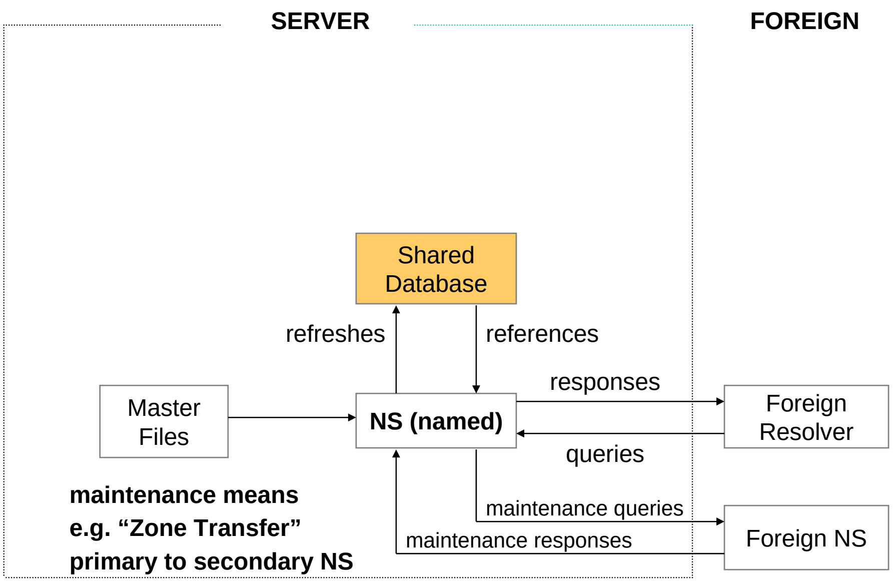
Foreign NS

cache additions

references

Shared Database

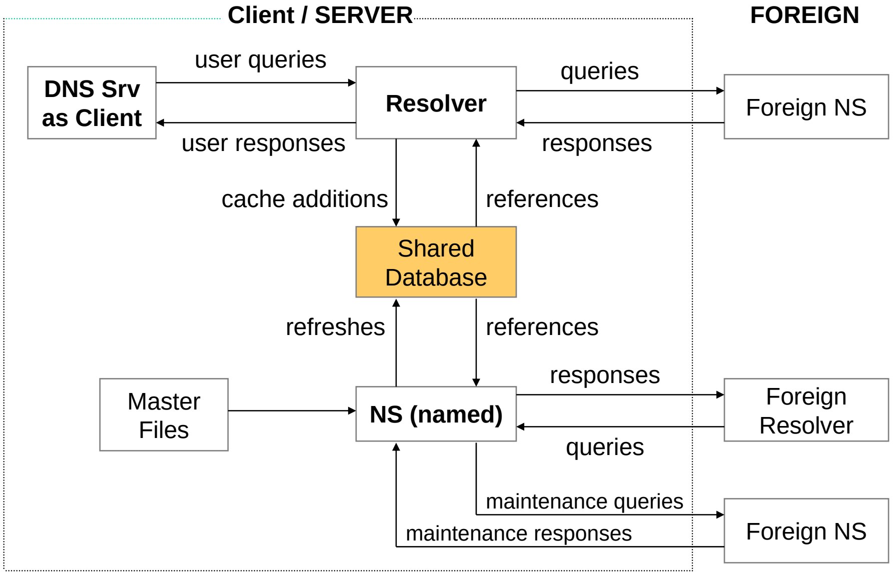
BIND Principles (Server)



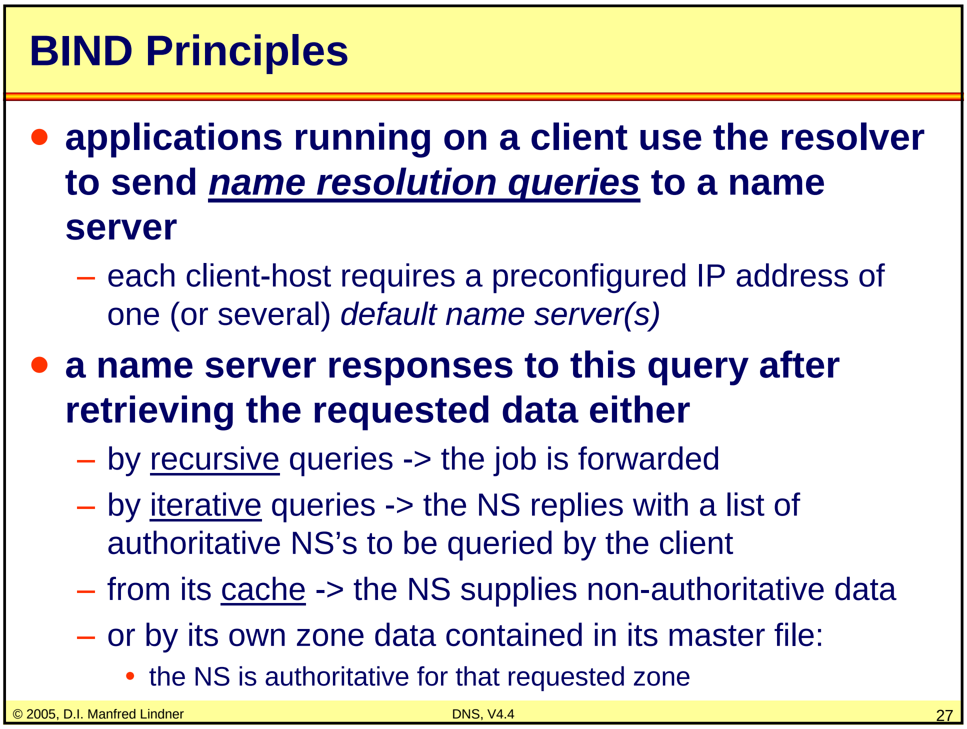
maintenance means e.g. “Zone Transfer” primary to secondary NS

© 2005, D.I. Manfred Lindner DNS, V4.4 25

BIND Principles (Server complete)

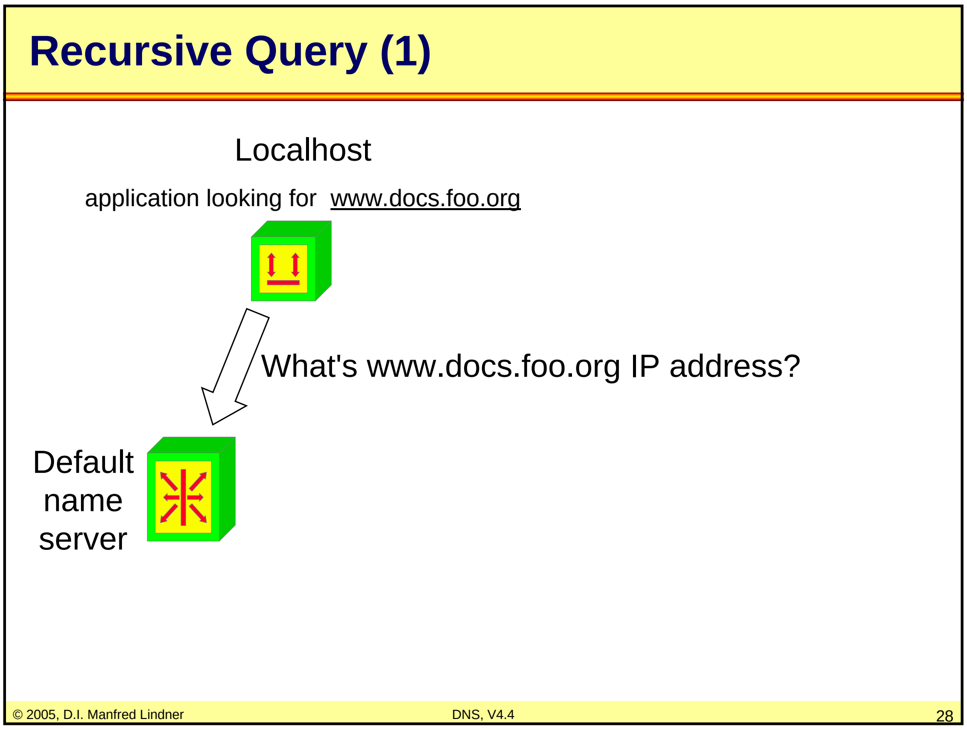


© 2005, D.I. Manfred Lindner DNS, V4.4 26



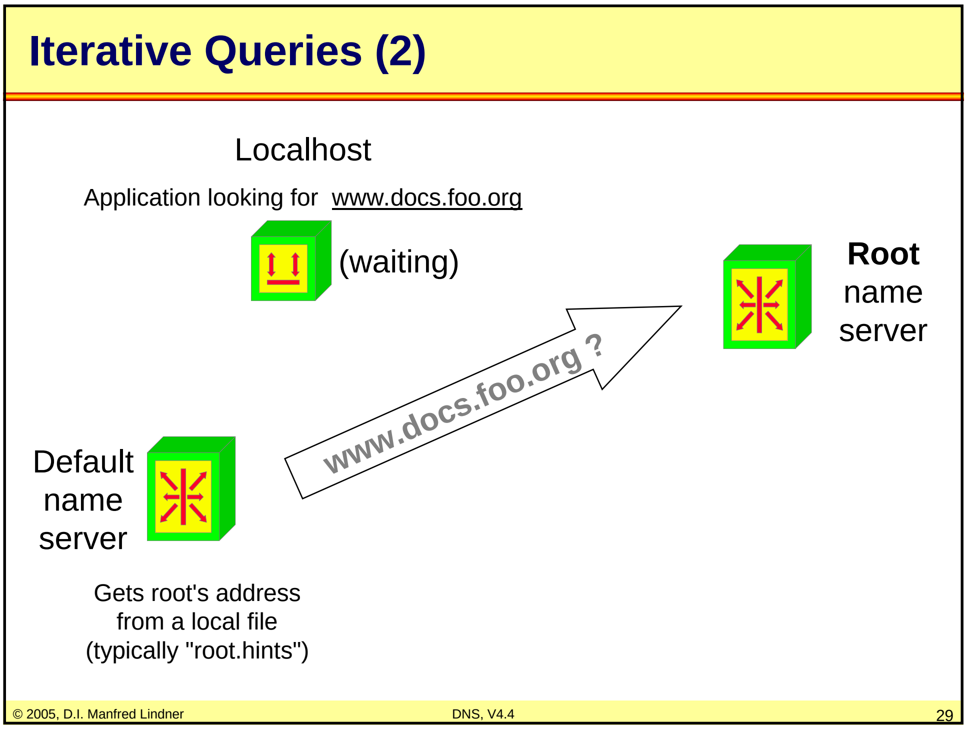
BIND Principles

- **applications running on a client use the resolver to send *name resolution queries* to a name server**
  - each client-host requires a preconfigured IP address of one (or several) *default name server(s)*
- **a name server responses to this query after retrieving the requested data either**
  - by recursive queries -> the job is forwarded
  - by iterative queries -> the NS replies with a list of authoritative NS's to be queried by the client
  - from its cache -> the NS supplies non-authoritative data
  - or by its own zone data contained in its master file:
    - the NS is authoritative for that requested zone



Recursive Query (1)

Localhost
application looking for www.docs.foo.org
What's www.docs.foo.org IP address?
Default name server

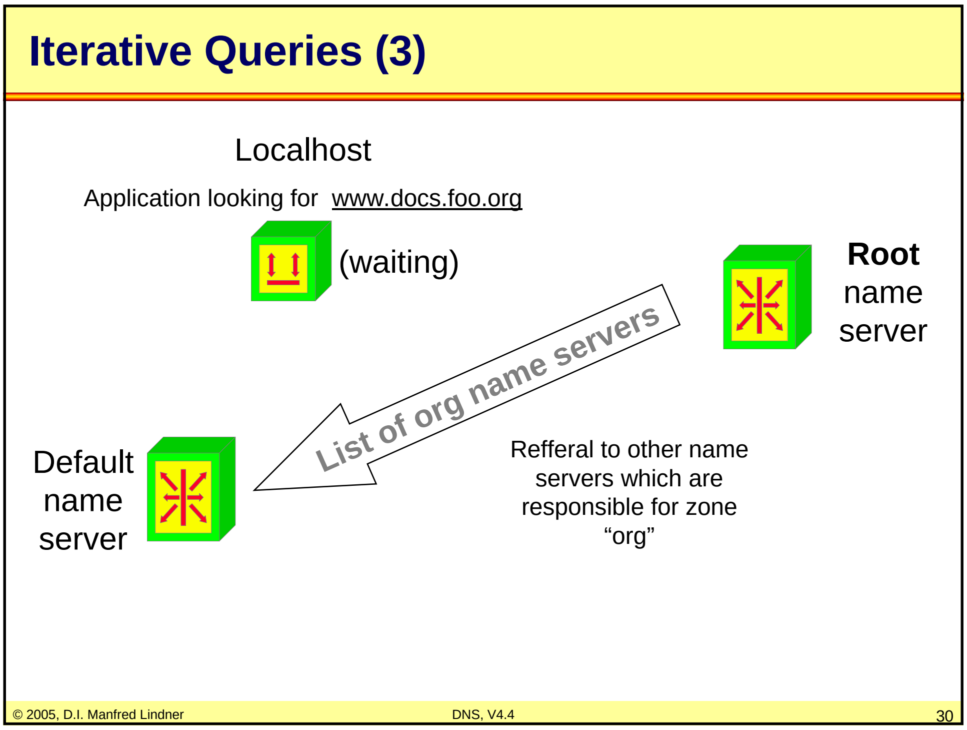


Iterative Queries (2)

Localhost

Application looking for www.docs.foo.org

Gets root's address from a local file (typically "root.hints")

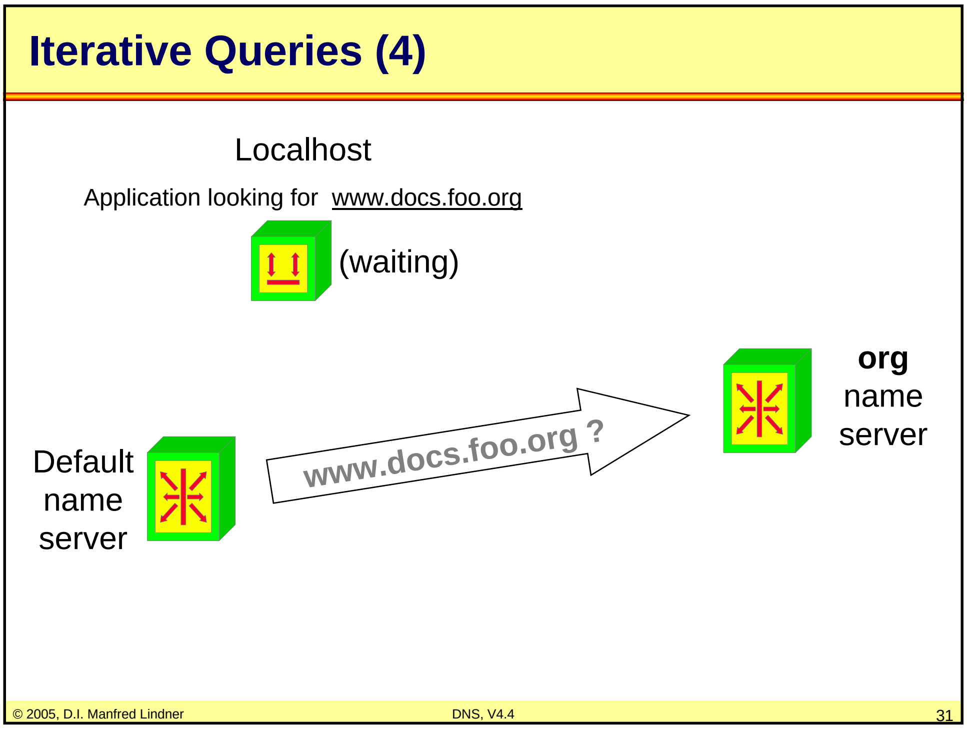


Iterative Queries (3)

Localhost

Application looking for www.docs.foo.org

Refferal to other name servers which are responsible for zone “org”



Iterative Queries (4)

Localhost

Application looking for www.docs.foo.org

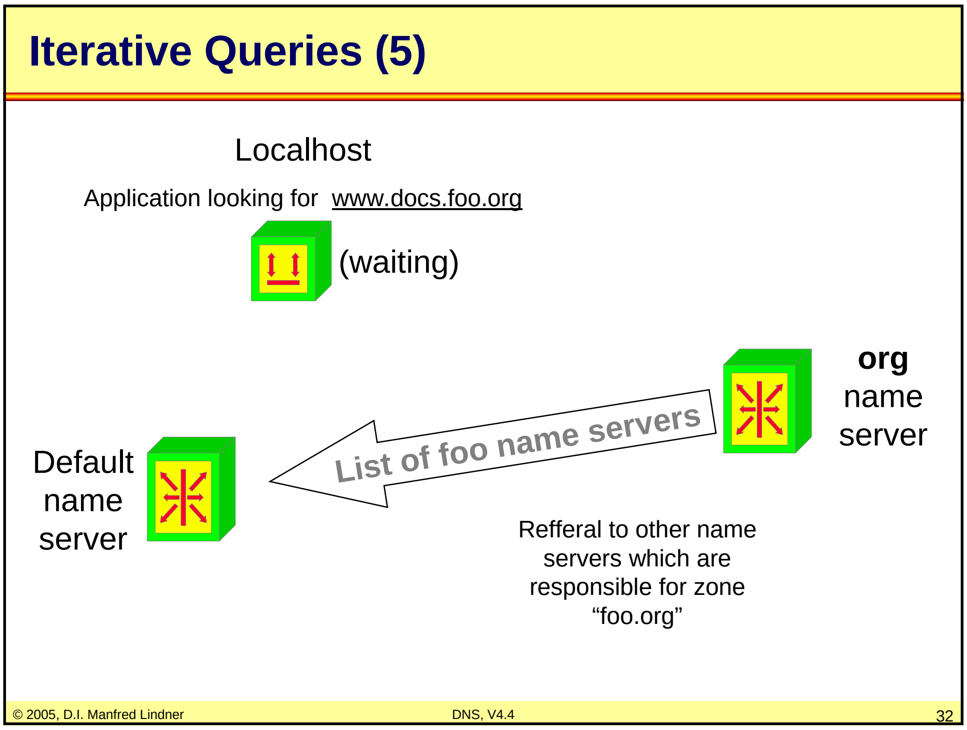
(waiting)

Default name server

www.docs.foo.org ?

org name server

© 2005, D.I. Manfred Lindner DNS, V4.4 31



Iterative Queries (5)

Localhost

Application looking for www.docs.foo.org

(waiting)

Default name server

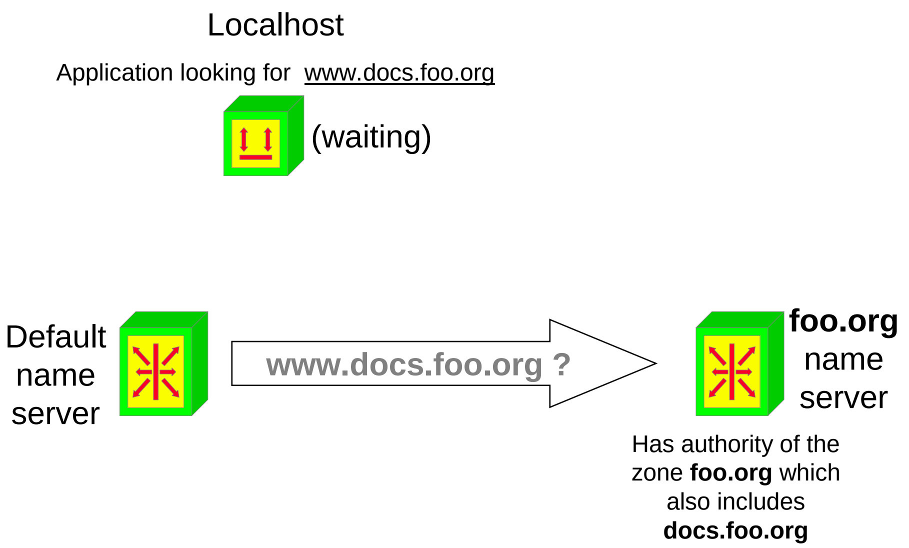
List of foo name servers

org name server

Refferal to other name servers which are responsible for zone “foo.org”

© 2005, D.I. Manfred Lindner DNS, V4.4 32

Iterative Queries (6)



Localhost

Application looking for www.docs.foo.org

(waiting)

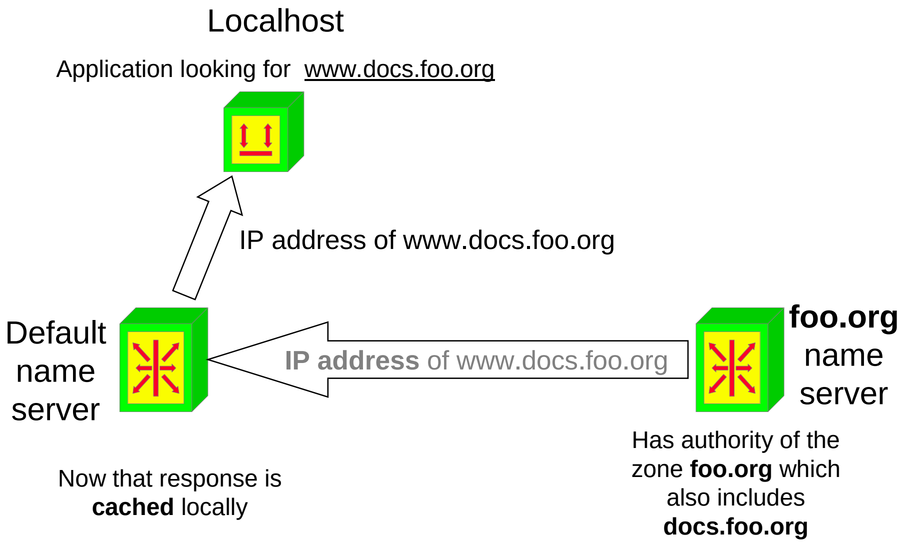
Default name server

www.docs.foo.org ?

foo.org name server

Has authority of the zone **foo.org** which also includes **docs.foo.org**

Iterative Queries (7)



Localhost

Application looking for www.docs.foo.org

IP address of www.docs.foo.org

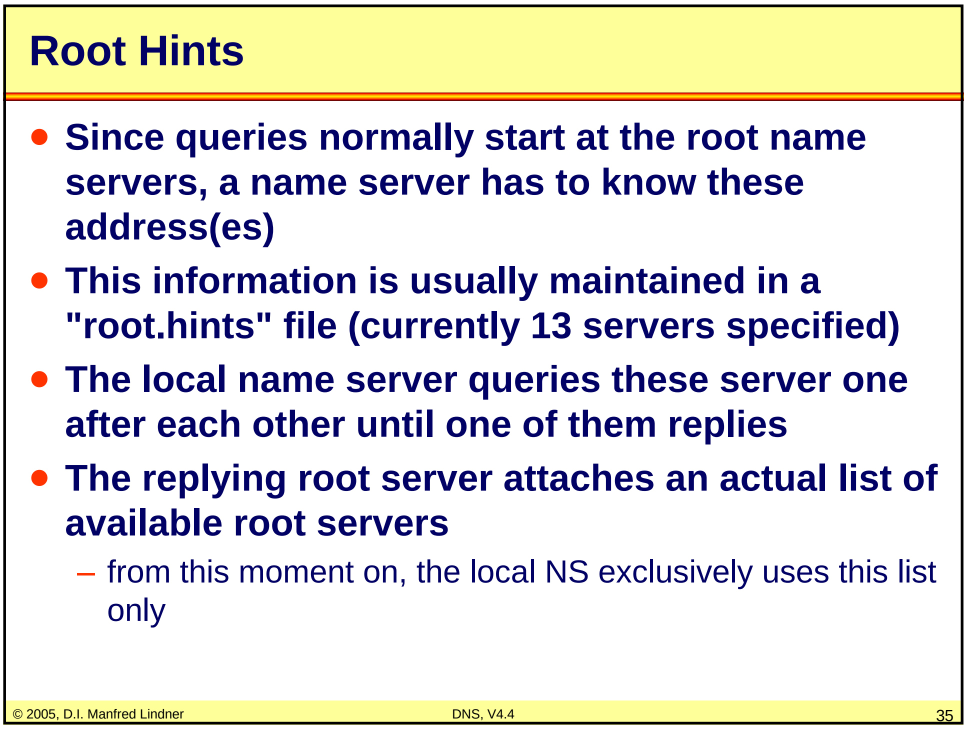
Default name server

IP address of www.docs.foo.org

foo.org name server

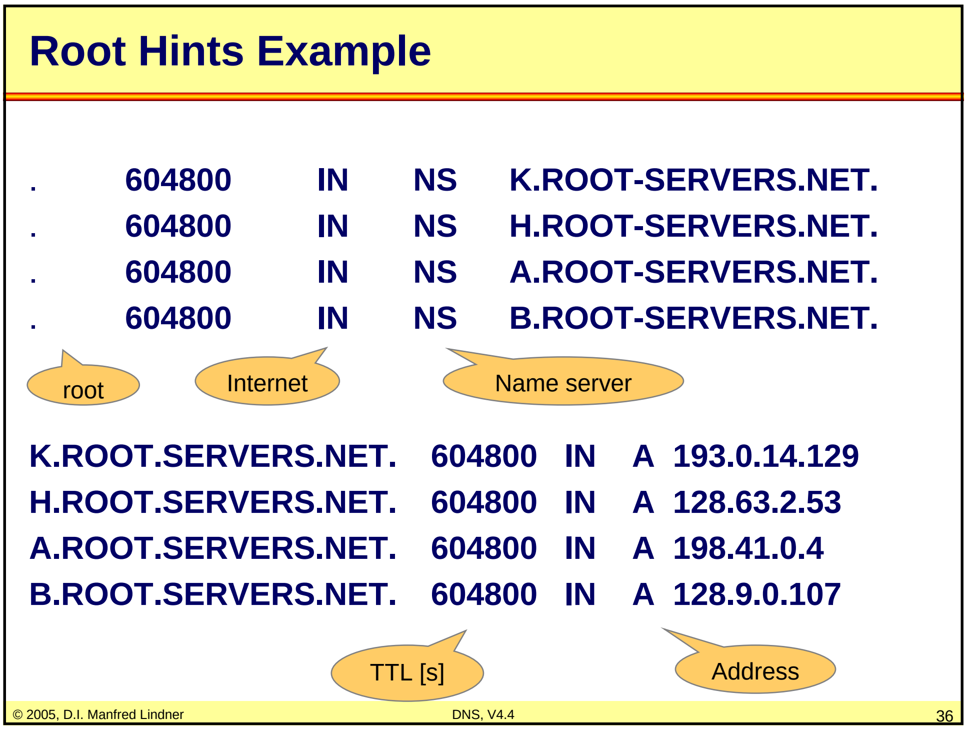
Now that response is **cached** locally

Has authority of the zone **foo.org** which also includes **docs.foo.org**



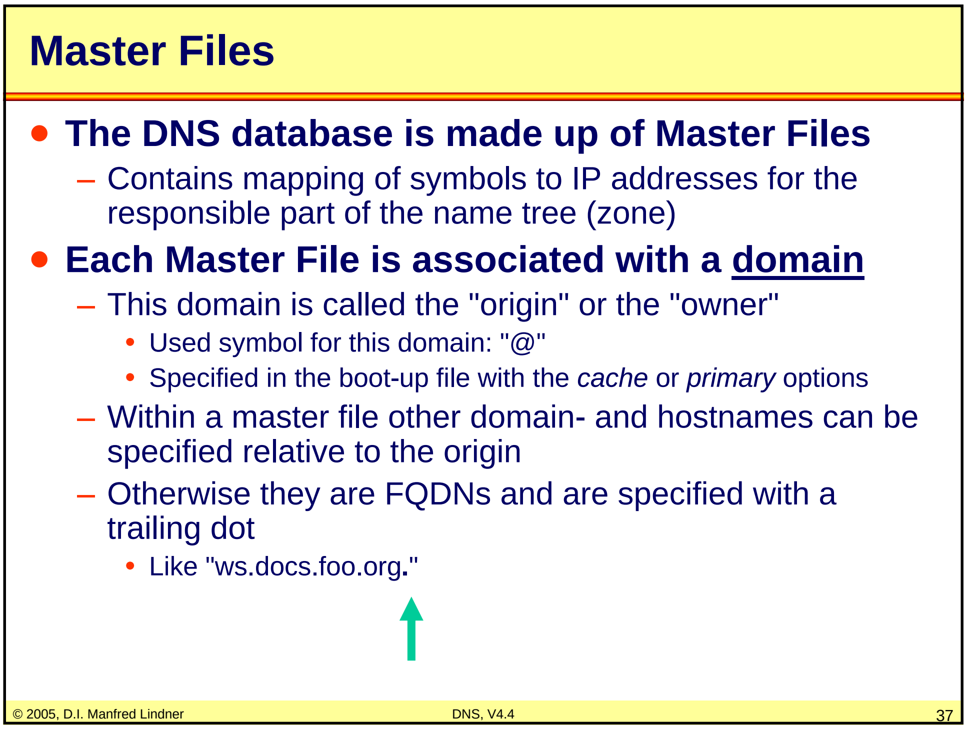
Root Hints

- Since queries normally start at the root name servers, a name server has to know these address(es)
- This information is usually maintained in a "root.hints" file (currently 13 servers specified)
- The local name server queries these server one after each other until one of them replies
- The replying root server attaches an actual list of available root servers
  - from this moment on, the local NS exclusively uses this list only



Root Hints Example

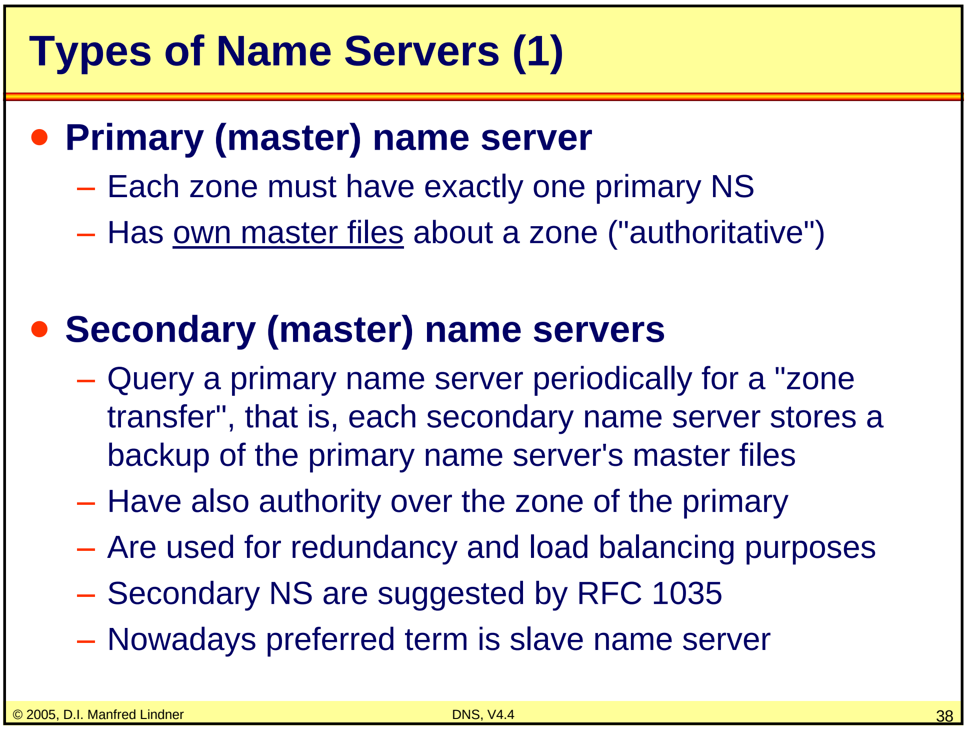
```
.       604800    IN    NS    K.ROOT-SERVERS.NET.
.       604800    IN    NS    H.ROOT-SERVERS.NET.
.       604800    IN    NS    A.ROOT-SERVERS.NET.
.       604800    IN    NS    B.ROOT-SERVERS.NET.

K.ROOT.SERVERS.NET.   604800   IN   A  193.0.14.129
H.ROOT.SERVERS.NET.   604800   IN   A  128.63.2.53
A.ROOT.SERVERS.NET.   604800   IN   A  198.41.0.4
B.ROOT.SERVERS.NET.   604800   IN   A  128.9.0.107
```



Master Files

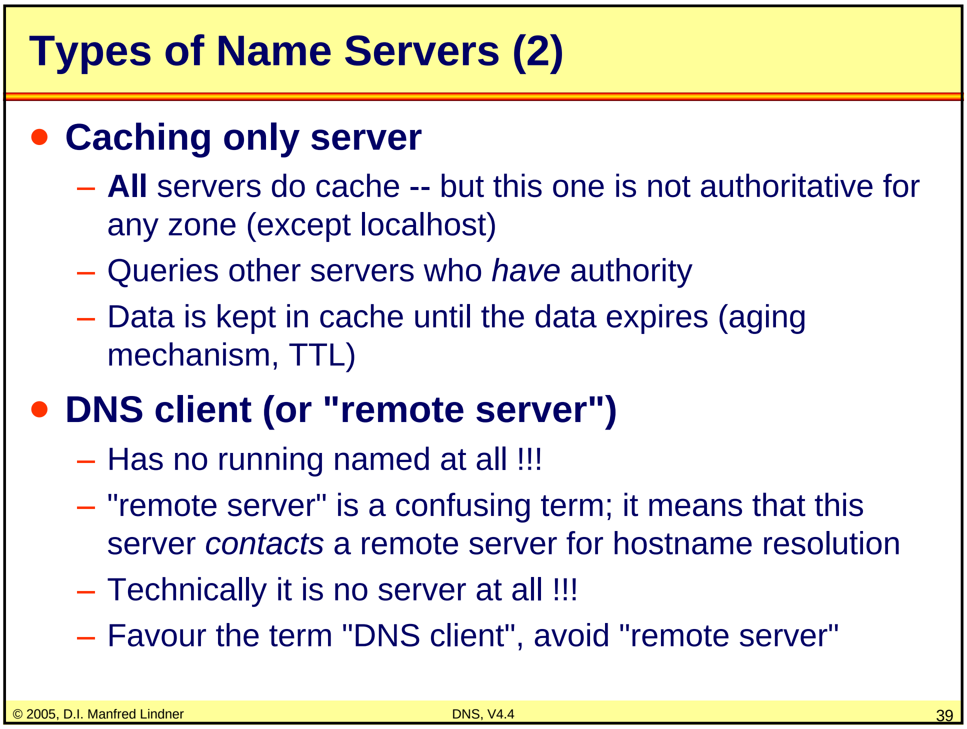
- **The DNS database is made up of Master Files**
  - Contains mapping of symbols to IP addresses for the responsible part of the name tree (zone)
- **Each Master File is associated with a domain**
  - This domain is called the "origin" or the "owner"
    - Used symbol for this domain: "@"
    - Specified in the boot-up file with the *cache* or *primary* options
  - Within a master file other domain- and hostnames can be specified relative to the origin
  - Otherwise they are FQDNs and are specified with a trailing dot
    - Like "ws.docs.foo.org**.**"

↑



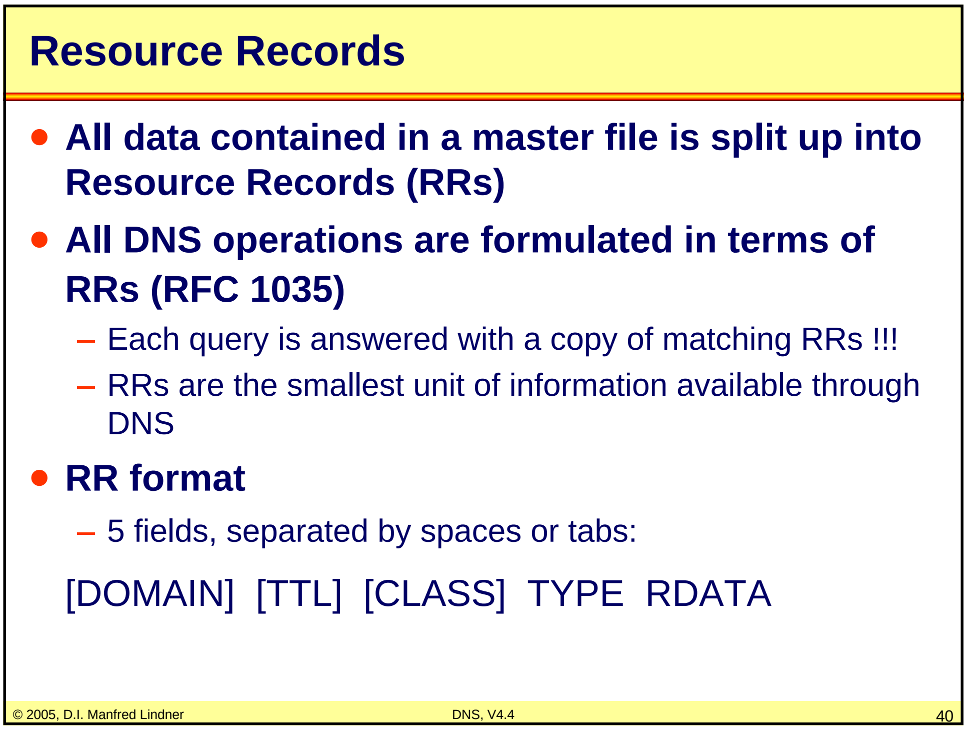
Types of Name Servers (1)

- **Primary (master) name server**
  - Each zone must have exactly one primary NS
  - Has own master files about a zone ("authoritative")
- **Secondary (master) name servers**
  - Query a primary name server periodically for a "zone transfer", that is, each secondary name server stores a backup of the primary name server's master files
  - Have also authority over the zone of the primary
  - Are used for redundancy and load balancing purposes
  - Secondary NS are suggested by RFC 1035
  - Nowadays preferred term is slave name server



Types of Name Servers (2)

- **Caching only server**
  - **All** servers do cache -- but this one is not authoritative for any zone (except localhost)
  - Queries other servers who *have* authority
  - Data is kept in cache until the data expires (aging mechanism, TTL)
- **DNS client (or "remote server")**
  - Has no running named at all !!!
  - "remote server" is a confusing term; it means that this server *contacts* a remote server for hostname resolution
  - Technically it is no server at all !!!
  - Favour the term "DNS client", avoid "remote server"



Resource Records

- **All data contained in a master file is split up into Resource Records (RRs)**
- **All DNS operations are formulated in terms of RRs (RFC 1035)**
  - Each query is answered with a copy of matching RRs !!!
  - RRs are the smallest unit of information available through DNS
- **RR format**
  - 5 fields, separated by spaces or tabs:

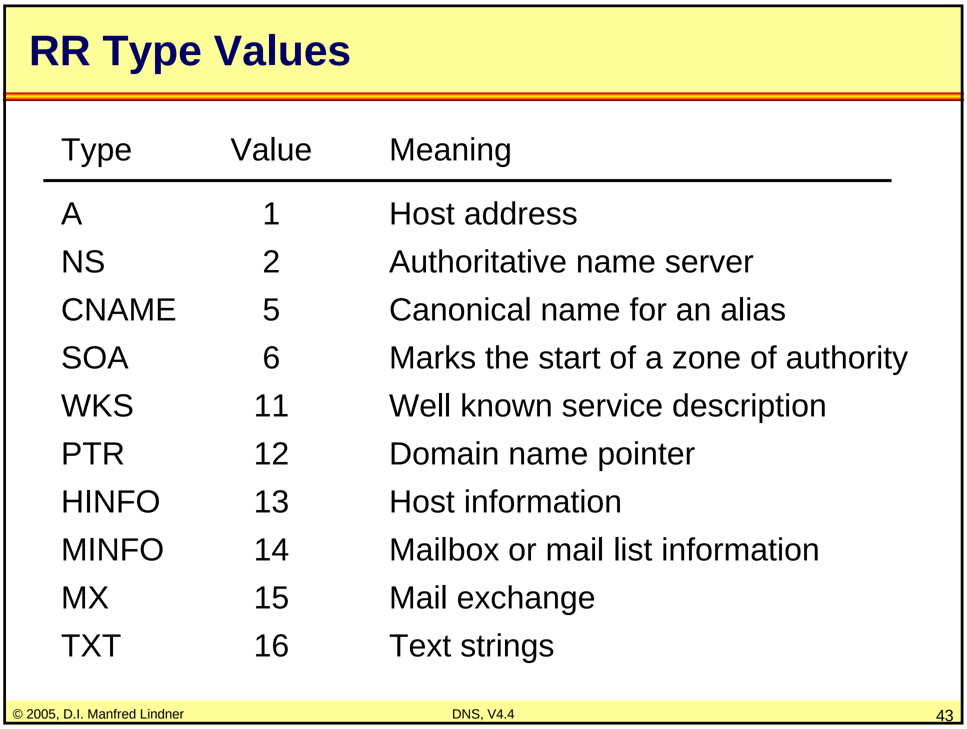
[DOMAIN] [TTL] [CLASS] TYPE RDATA

Resource Record Components (1)

- **DOMAIN**
  - Domain name to which the entry applies
  - If no domain name is given the RR applies to the domain of the previous RR
- **TTL**
  - Time To Live = time in seconds this RR is valid after it has been retrieved from the server
  - 8 digit decimal number
- **CLASS**
  - Address class: IN for Internet, CH for CHAOS, HS for Hesiod (MIT)
  - 2 bytes

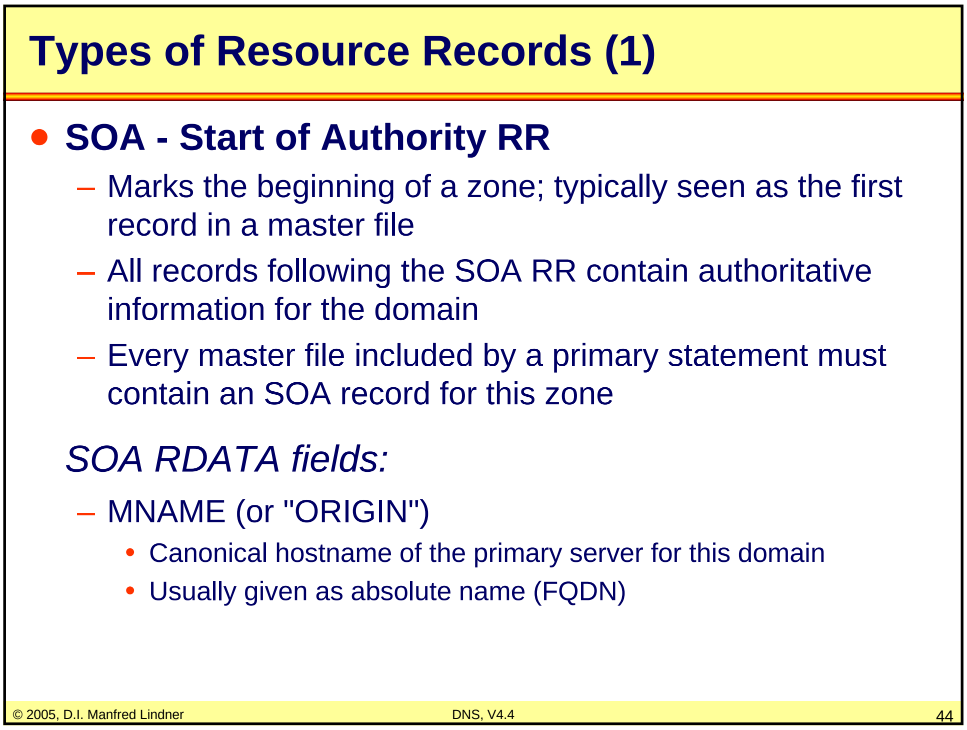
Resource Record Components (2)

- **TYPE**
  - Describes the type of the RR
  - e.g. SOA, A, NS, PTR (see below)
  - 2 bytes
- **RDATA**
  - Contains the actual data of the RR
  - Its format depends on the type of the RR (see below)
  - Variable length



RR Type Values

Type	Value	Meaning
A	1	Host address
NS	2	Authoritative name server
CNAME	5	Canonical name for an alias
SOA	6	Marks the start of a zone of authority
WKS	11	Well known service description
PTR	12	Domain name pointer
HINFO	13	Host information
MINFO	14	Mailbox or mail list information
MX	15	Mail exchange
TXT	16	Text strings

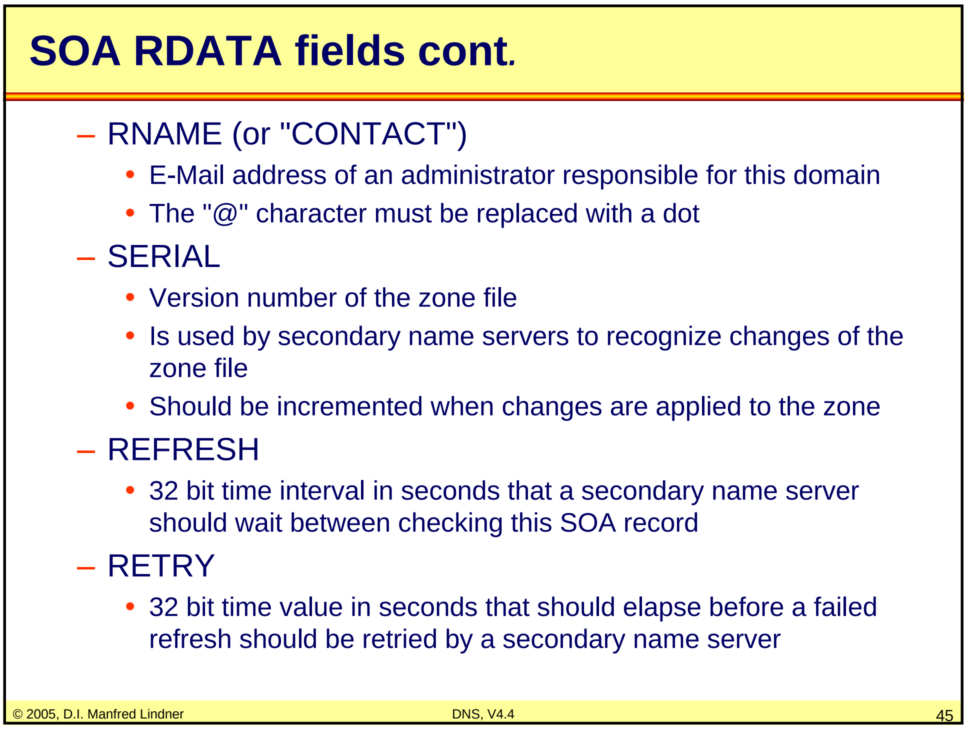


Types of Resource Records (1)

- **SOA - Start of Authority RR**
  - Marks the beginning of a zone; typically seen as the first record in a master file
  - All records following the SOA RR contain authoritative information for the domain
  - Every master file included by a primary statement must contain an SOA record for this zone

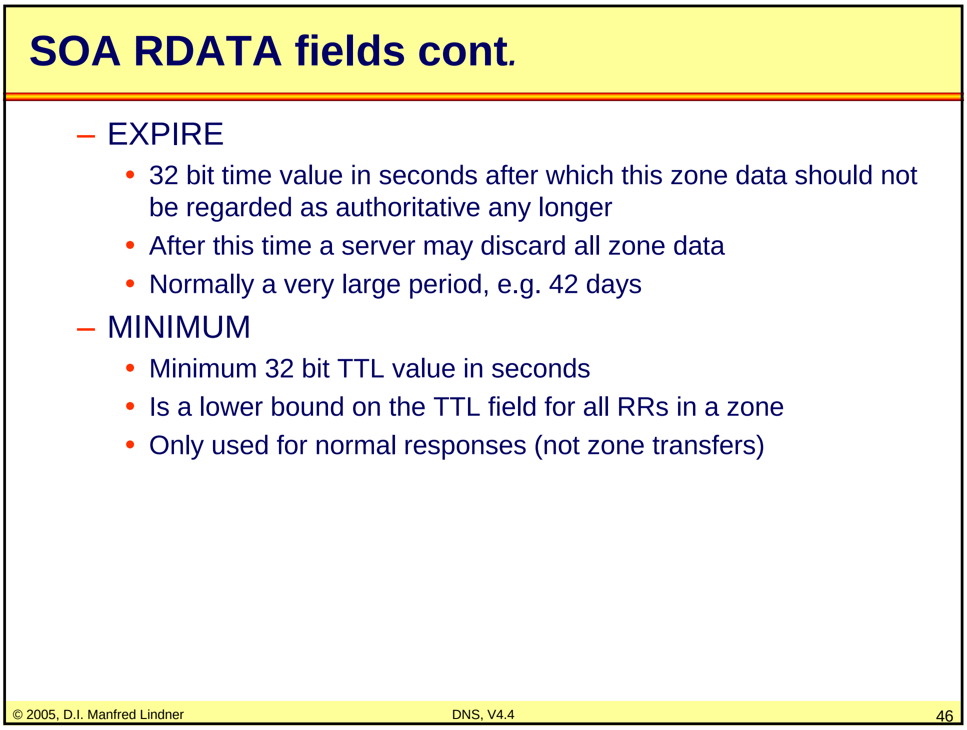
SOA RDATA fields:

- MNAME (or "ORIGIN")
  - Canonical hostname of the primary server for this domain
  - Usually given as absolute name (FQDN)



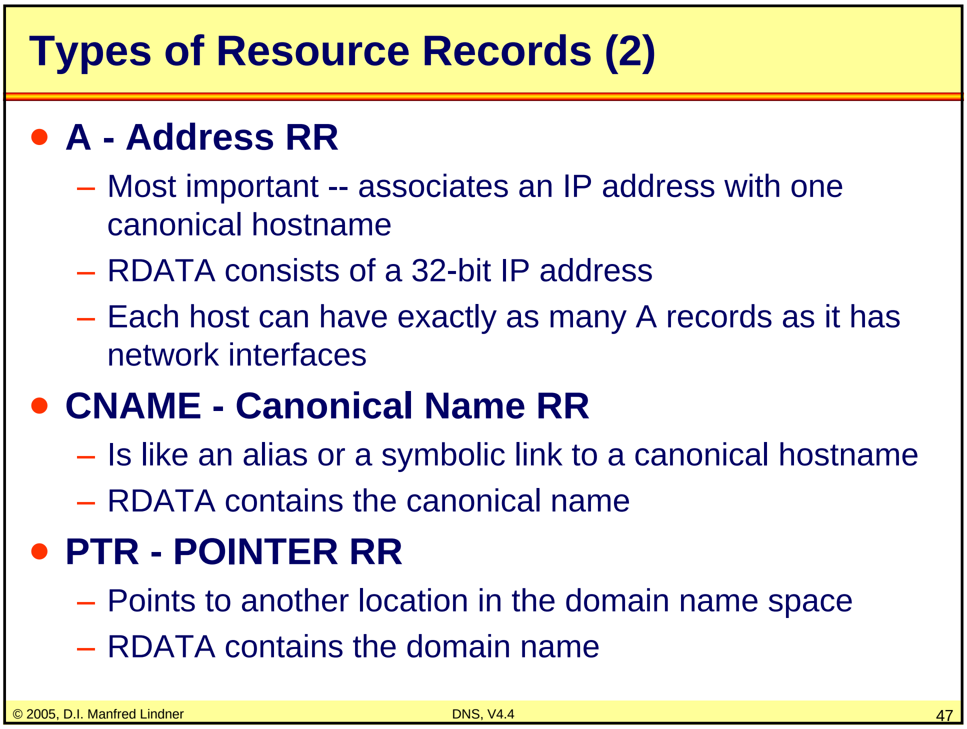
SOA RDATA fields cont.

- RNAME (or "CONTACT")
  - E-Mail address of an administrator responsible for this domain
  - The "@" character must be replaced with a dot
- SERIAL
  - Version number of the zone file
  - Is used by secondary name servers to recognize changes of the zone file
  - Should be incremented when changes are applied to the zone
- REFRESH
  - 32 bit time interval in seconds that a secondary name server should wait between checking this SOA record
- RETRY
  - 32 bit time value in seconds that should elapse before a failed refresh should be retried by a secondary name server



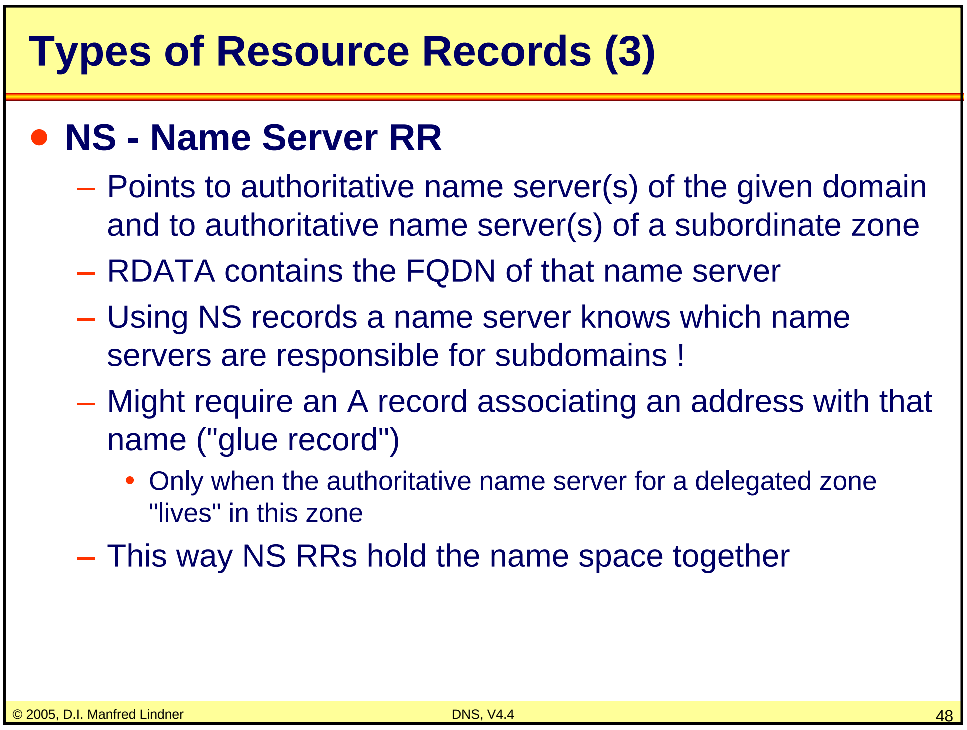
SOA RDATA fields cont.

- EXPIRE
  - 32 bit time value in seconds after which this zone data should not be regarded as authoritative any longer
  - After this time a server may discard all zone data
  - Normally a very large period, e.g. 42 days
- MINIMUM
  - Minimum 32 bit TTL value in seconds
  - Is a lower bound on the TTL field for all RRs in a zone
  - Only used for normal responses (not zone transfers)



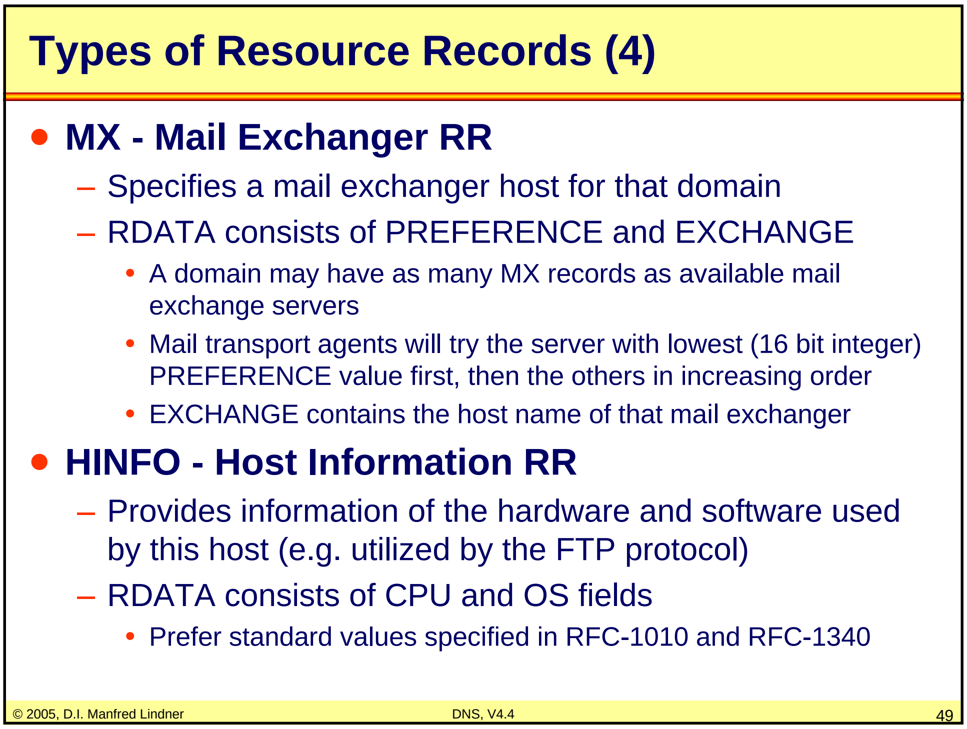
Types of Resource Records (2)

- **A - Address RR**
  - Most important -- associates an IP address with one canonical hostname
  - RDATA consists of a 32-bit IP address
  - Each host can have exactly as many A records as it has network interfaces
- **CNAME - Canonical Name RR**
  - Is like an alias or a symbolic link to a canonical hostname
  - RDATA contains the canonical name
- **PTR - POINTER RR**
  - Points to another location in the domain name space
  - RDATA contains the domain name



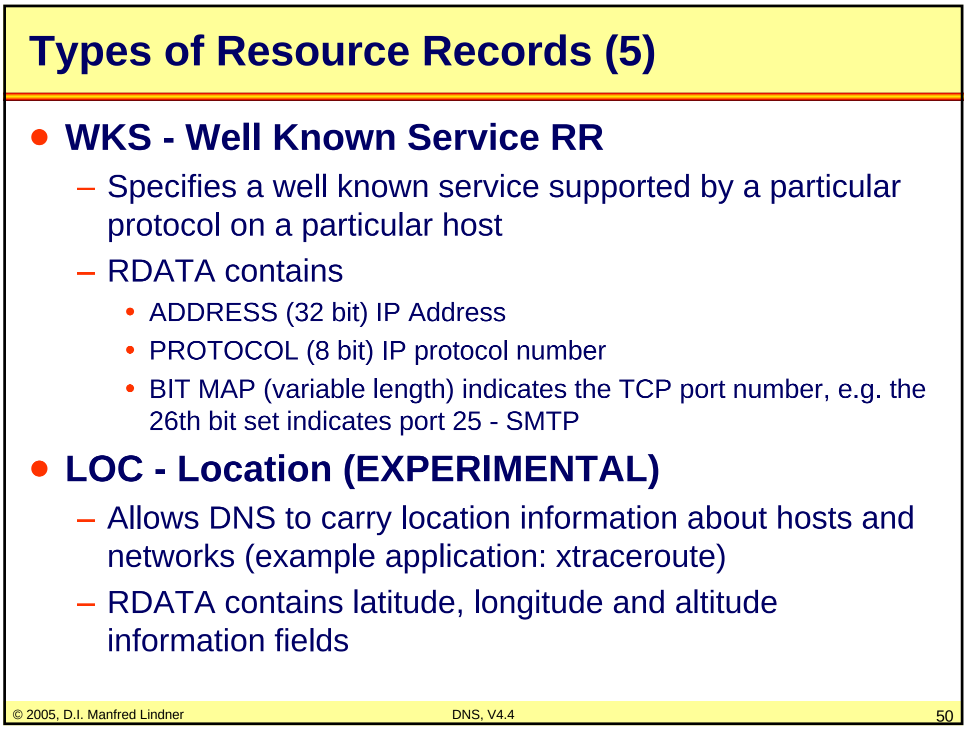
Types of Resource Records (3)

- **NS - Name Server RR**
  - Points to authoritative name server(s) of the given domain and to authoritative name server(s) of a subordinate zone
  - RDATA contains the FQDN of that name server
  - Using NS records a name server knows which name servers are responsible for subdomains !
  - Might require an A record associating an address with that name ("glue record")
    - Only when the authoritative name server for a delegated zone "lives" in this zone
  - This way NS RRs hold the name space together



Types of Resource Records (4)

- **MX - Mail Exchanger RR**
  - Specifies a mail exchanger host for that domain
  - RDATA consists of PREFERENCE and EXCHANGE
    - A domain may have as many MX records as available mail exchange servers
    - Mail transport agents will try the server with lowest (16 bit integer) PREFERENCE value first, then the others in increasing order
    - EXCHANGE contains the host name of that mail exchanger
- **HINFO - Host Information RR**
  - Provides information of the hardware and software used by this host (e.g. utilized by the FTP protocol)
  - RDATA consists of CPU and OS fields
    - Prefer standard values specified in RFC-1010 and RFC-1340



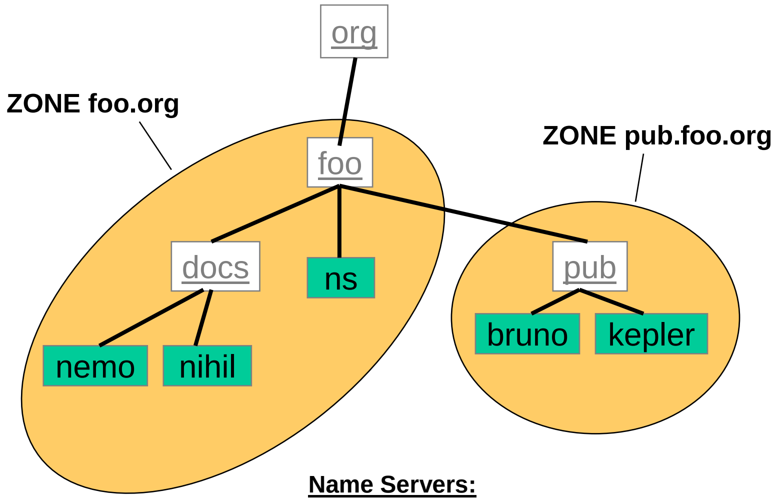
Types of Resource Records (5)

- **WKS - Well Known Service RR**
  - Specifies a well known service supported by a particular protocol on a particular host
  - RDATA contains
    - ADDRESS (32 bit) IP Address
    - PROTOCOL (8 bit) IP protocol number
    - BIT MAP (variable length) indicates the TCP port number, e.g. the 26th bit set indicates port 25 - SMTP
- **LOC - Location (EXPERIMENTAL)**
  - Allows DNS to carry location information about hosts and networks (example application: xtraceroute)
  - RDATA contains latitude, longitude and altitude information fields

Multihome/Virtual Host Support

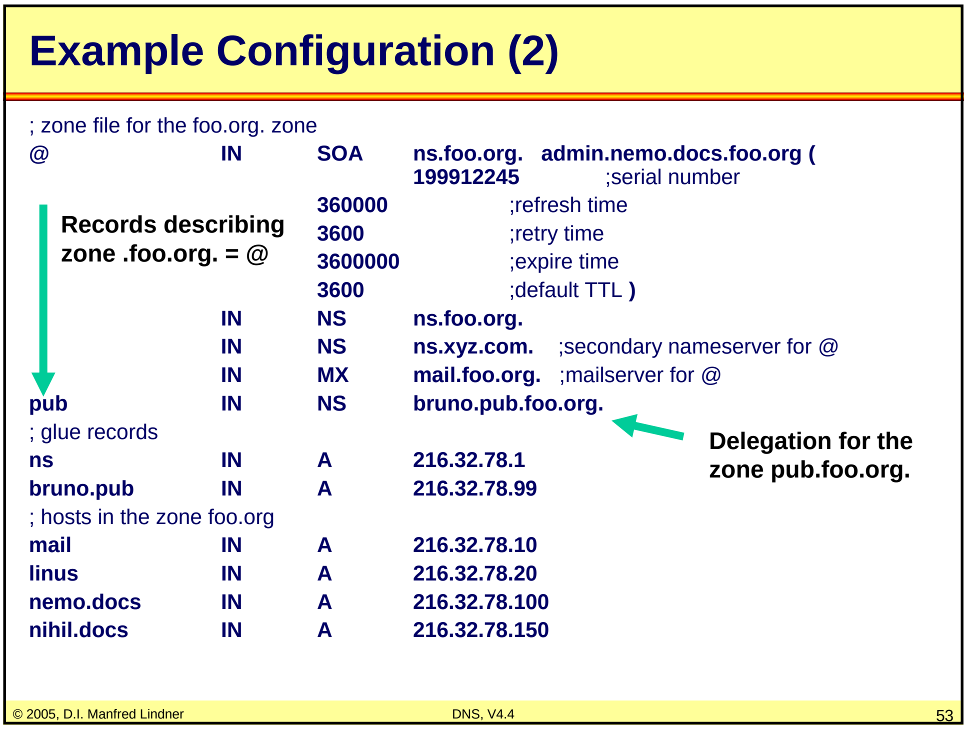
- **Using CNAME resource records of DNS**
  - Servers can appear with more than one name in the Internet
  - So, several CNAMES such as www, ftp, news, mail, etc... correspond to a single machine
- **If a server has several IP addresses with different CNAMES**
  - Then different websites should be delivered according to the CNAME
  - E.g. www.foo.org and www.bar.com could be the same webserver on the same machine
  - Each CNAME reflects another IP address
- **Solution: Webservers can be configured "multihomed"**
  - Webserver recognizes which interface (which IP address) the server was called from
  - Also called "Virtual Host Support"

Example Configuration (1)



Name Servers:

ns.foo.org
bruno.pub.foo.org



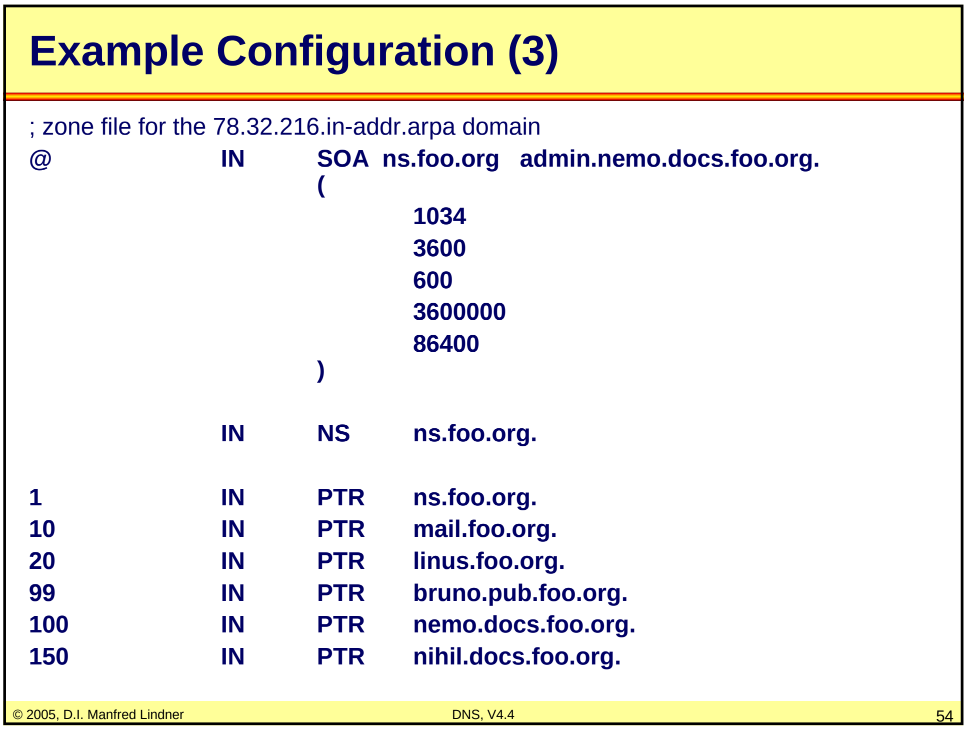
Example Configuration (2)

```
; zone file for the foo.org. zone
@              IN     SOA    ns.foo.org.   admin.nemo.docs.foo.org (
                             199912245          ;serial number
                      360000           ;refresh time
                      3600             ;retry time
                      3600000          ;expire time
                      3600             ;default TTL )
               IN     NS     ns.foo.org.
               IN     NS     ns.xyz.com.    ;secondary nameserver for @
               IN     MX     mail.foo.org.  ;mailserver for @
pub            IN     NS     bruno.pub.foo.org.
; glue records
ns             IN     A      216.32.78.1
bruno.pub      IN     A      216.32.78.99
; hosts in the zone foo.org
mail           IN     A      216.32.78.10
linus          IN     A      216.32.78.20
nemo.docs      IN     A      216.32.78.100
nihil.docs     IN     A      216.32.78.150
```

Records describing zone .foo.org. = @

Delegation for the zone pub.foo.org.

© 2005, D.I. Manfred Lindner DNS, V4.4 53



Example Configuration (3)

```
; zone file for the 78.32.216.in-addr.arpa domain
@              IN     SOA  ns.foo.org   admin.nemo.docs.foo.org.
                      (
                             1034
                             3600
                             600
                             3600000
                             86400
                      )

               IN     NS     ns.foo.org.

1              IN     PTR    ns.foo.org.
10             IN     PTR    mail.foo.org.
20             IN     PTR    linus.foo.org.
99             IN     PTR    bruno.pub.foo.org.
100            IN     PTR    nemo.docs.foo.org.
150            IN     PTR    nihil.docs.foo.org.
```

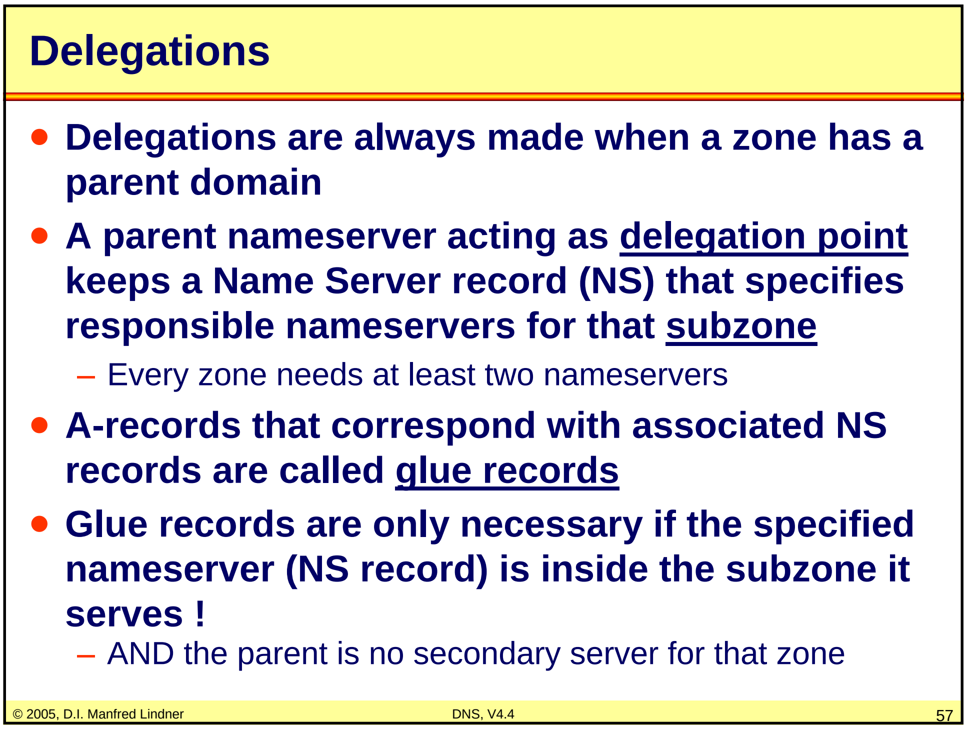
© 2005, D.I. Manfred Lindner DNS, V4.4 54

Example Configuration (4)

```
; zone file for pub.foo.org
@             IN      SOA  bruno.pub.foo.org
                                     hostmaster.bruno.pub.foo.org.
                      ( 1034
                      3600
                      600
                      3600000
                      86400 )
; Name Servers
              IN      NS             bruno
              IN      NS             ns.foo.org.        ;secondary NS
; glue records
bruno         IN      A              216.32.78.99

nameserver    IN      CNAME          bruno
```

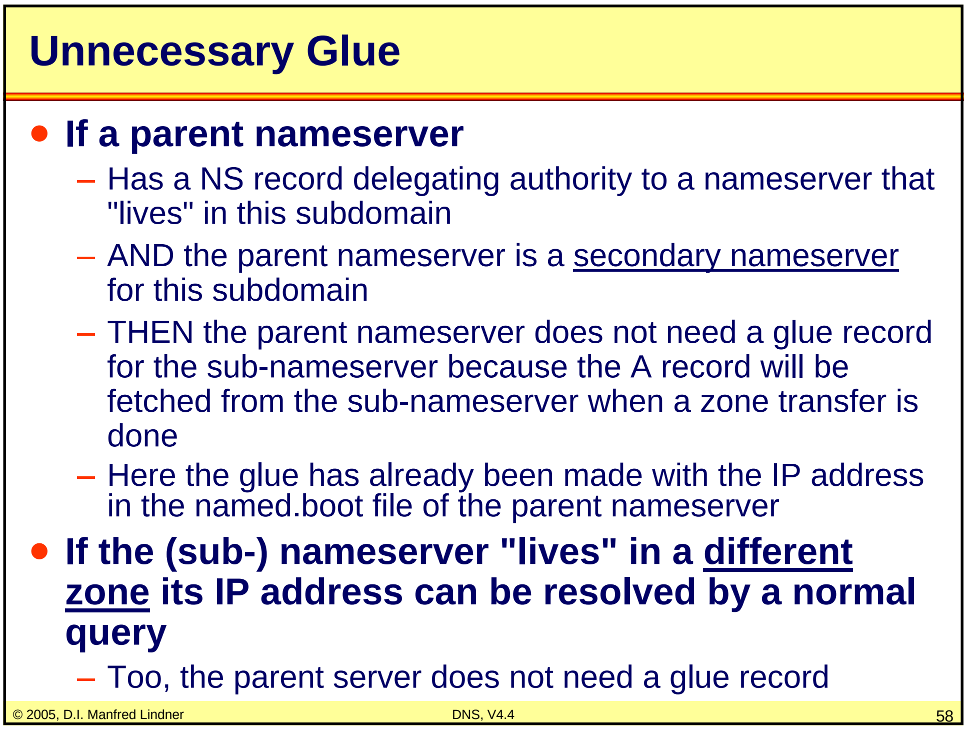
Example Configuration (5)

```
; other hosts:
kepler        IN      A              216.32.22.50
              IN      MX             1  mail.foo.com
              IN      MX             2  picasso.art.net
              IN      MX             5  mail.ct.oberon.tuwien.ac.at
aristarch     IN      A              216.32.22.51
galilei       IN      A              216.32.22.52
              IN      HINFO          VAX-11/780  UNIX
              IN      WKS            216.32.22.52  TCP (telnet ftp
                                     netstat finger pop)
laplace       IN      A              216.32.34.2
              IN      HINFO          SUN  UNIX
; etc.....
```



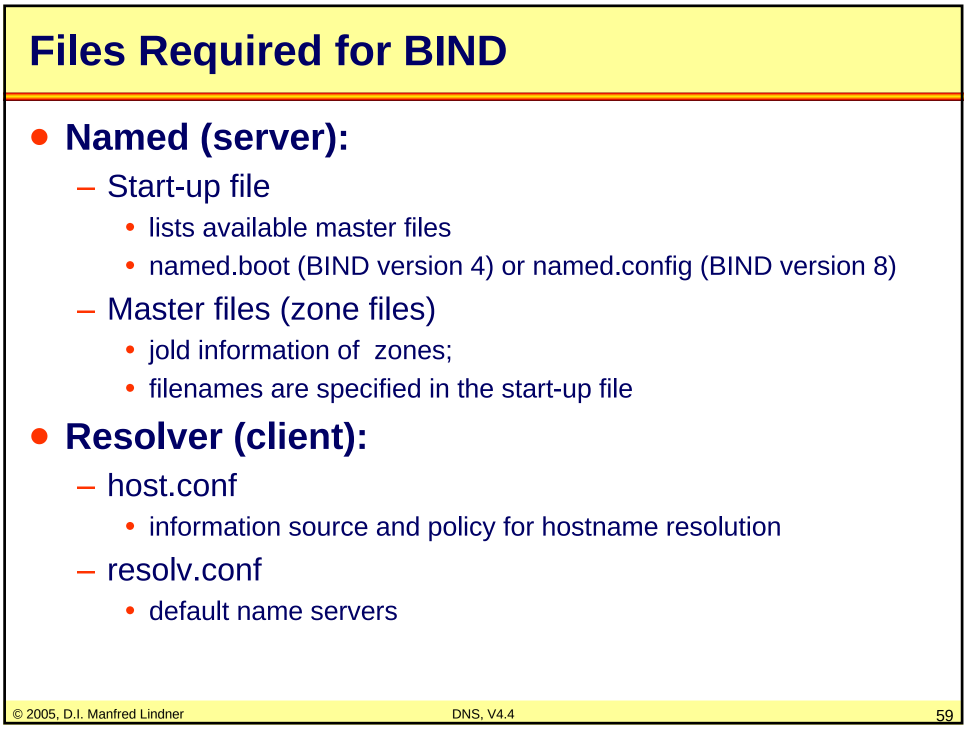
Delegations

- **Delegations are always made when a zone has a parent domain**
- **A parent nameserver acting as <u>delegation point</u> keeps a Name Server record (NS) that specifies responsible nameservers for that <u>subzone</u>**
  - Every zone needs at least two nameservers
- **A-records that correspond with associated NS records are called <u>glue records</u>**
- **Glue records are only necessary if the specified nameserver (NS record) is inside the subzone it serves !**
  - AND the parent is no secondary server for that zone



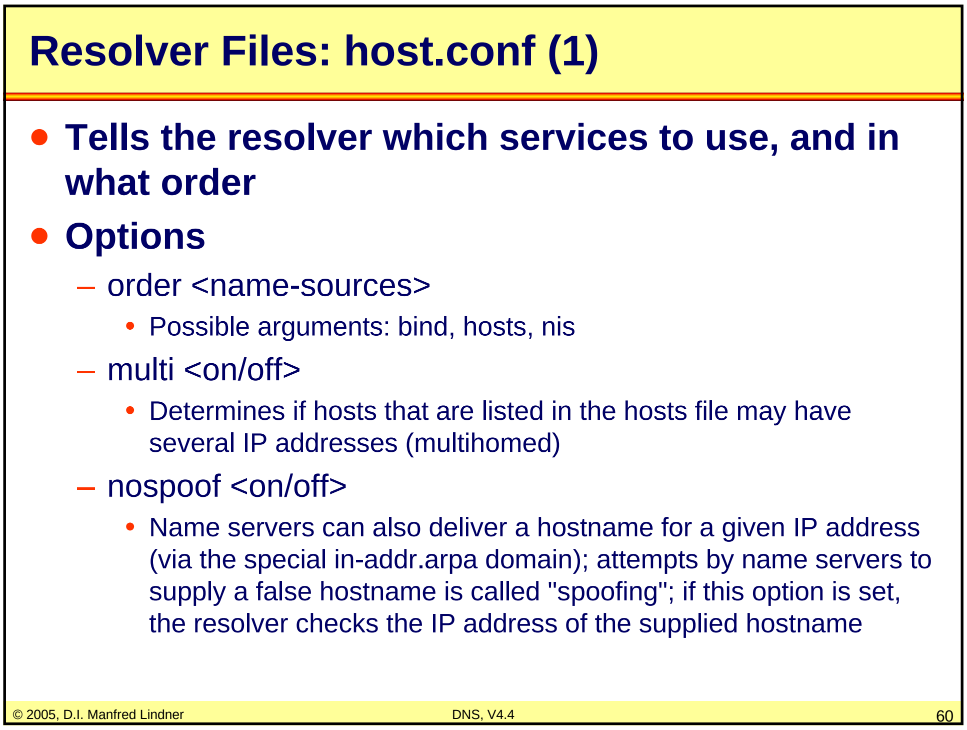
Unnecessary Glue

- **If a parent nameserver**
  - Has a NS record delegating authority to a nameserver that "lives" in this subdomain
  - AND the parent nameserver is a <u>secondary nameserver</u> for this subdomain
  - THEN the parent nameserver does not need a glue record for the sub-nameserver because the A record will be fetched from the sub-nameserver when a zone transfer is done
  - Here the glue has already been made with the IP address in the named.boot file of the parent nameserver
- **If the (sub-) nameserver "lives" in a <u>different zone</u> its IP address can be resolved by a normal query**
  - Too, the parent server does not need a glue record



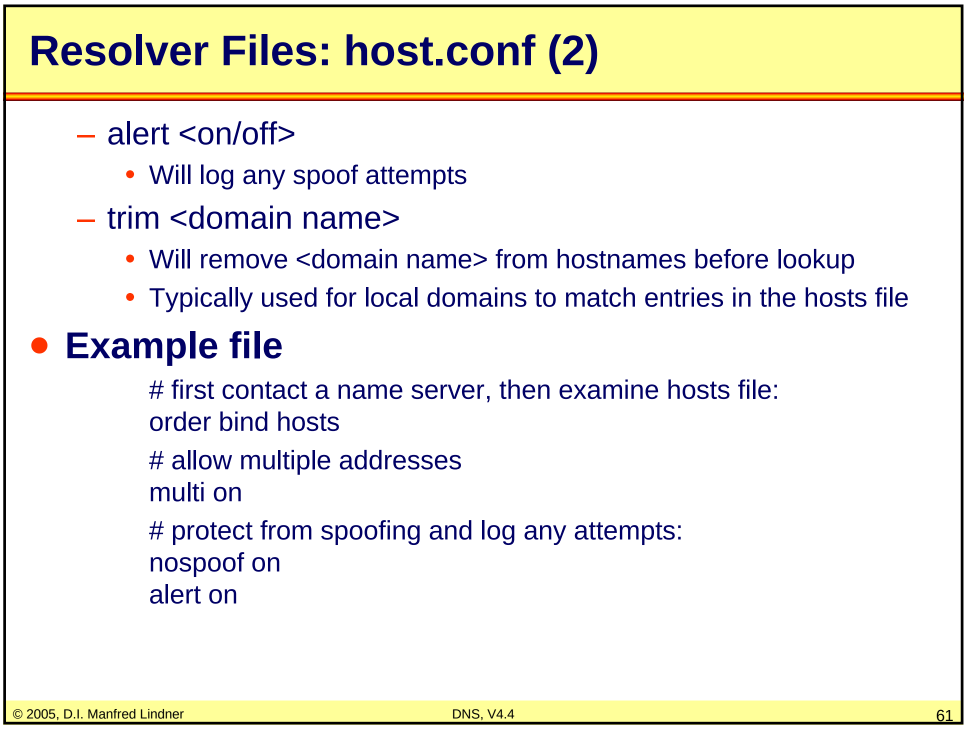
Files Required for BIND

- **Named (server):**
  - Start-up file
    - lists available master files
    - named.boot (BIND version 4) or named.config (BIND version 8)
  - Master files (zone files)
    - jold information of zones;
    - filenames are specified in the start-up file
- **Resolver (client):**
  - host.conf
    - information source and policy for hostname resolution
  - resolv.conf
    - default name servers



Resolver Files: host.conf (1)

- **Tells the resolver which services to use, and in what order**
- **Options**
  - order <name-sources>
    - Possible arguments: bind, hosts, nis
  - multi <on/off>
    - Determines if hosts that are listed in the hosts file may have several IP addresses (multihomed)
  - nospoof <on/off>
    - Name servers can also deliver a hostname for a given IP address (via the special in-addr.arpa domain); attempts by name servers to supply a false hostname is called "spoofing"; if this option is set, the resolver checks the IP address of the supplied hostname

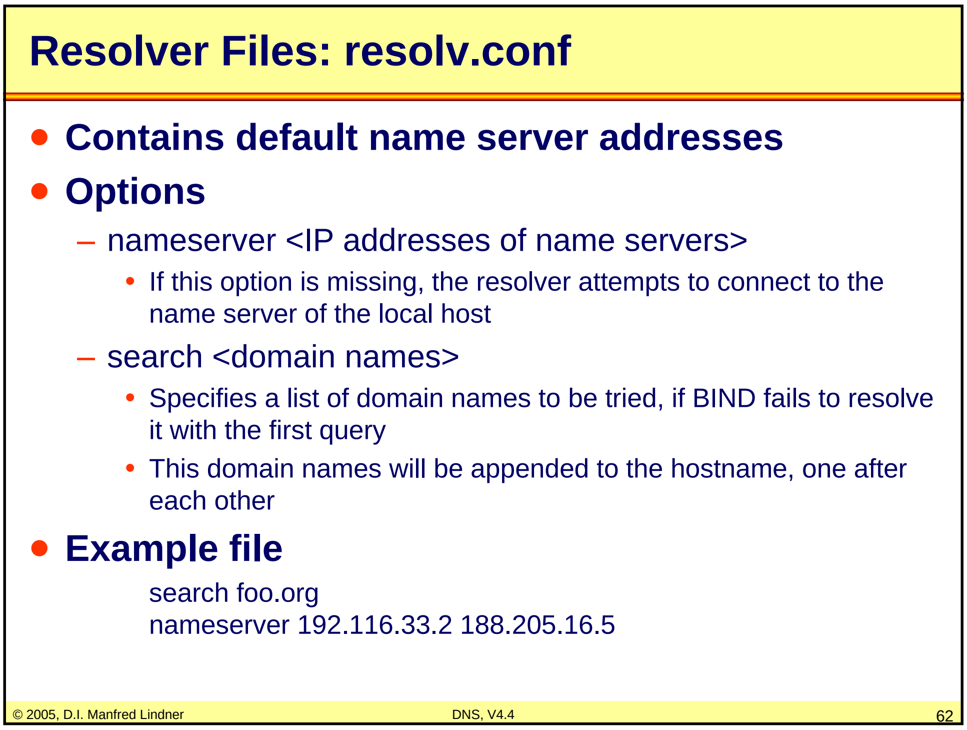


Resolver Files: host.conf (2)

- alert <on/off>
  - Will log any spoof attempts
- trim <domain name>
  - Will remove <domain name> from hostnames before lookup
  - Typically used for local domains to match entries in the hosts file

- **Example file**

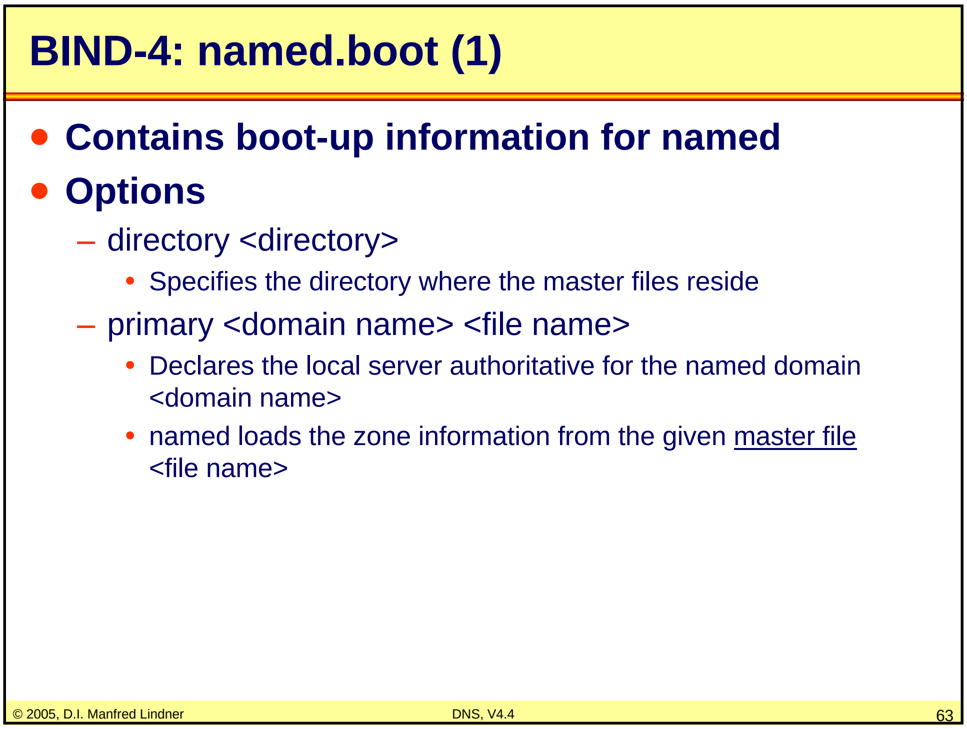
```
# first contact a name server, then examine hosts file:
order bind hosts
# allow multiple addresses
multi on
# protect from spoofing and log any attempts:
nospoof on
alert on
```



Resolver Files: resolv.conf

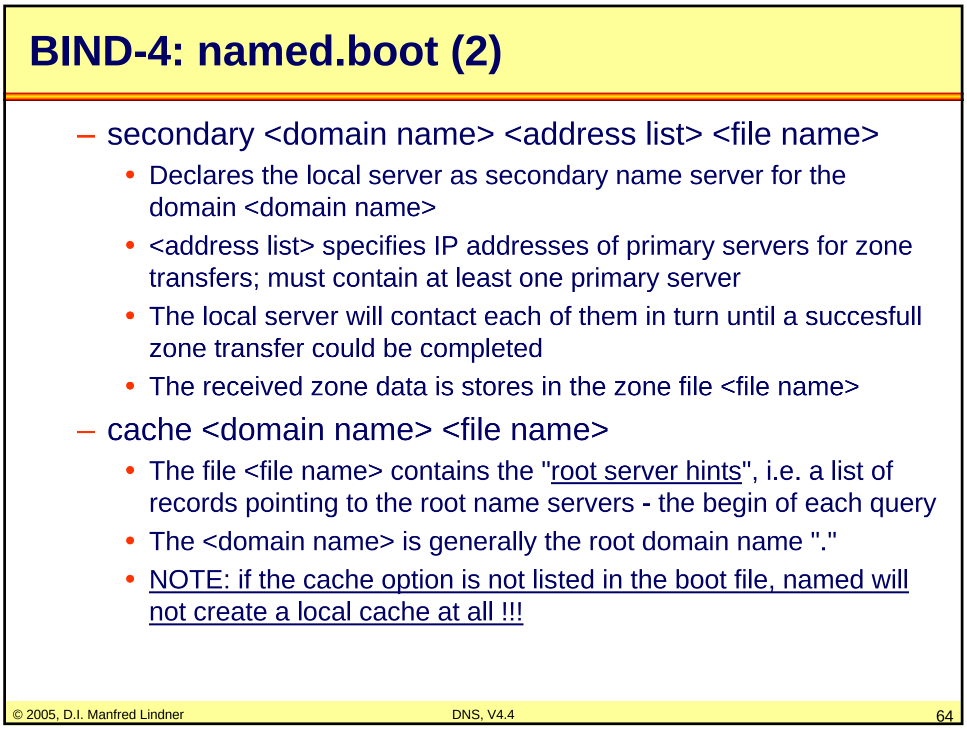
- **Contains default name server addresses**
- **Options**
  - nameserver <IP addresses of name servers>
    - If this option is missing, the resolver attempts to connect to the name server of the local host
  - search <domain names>
    - Specifies a list of domain names to be tried, if BIND fails to resolve it with the first query
    - This domain names will be appended to the hostname, one after each other
- **Example file**

```
search foo.org
nameserver 192.116.33.2 188.205.16.5
```



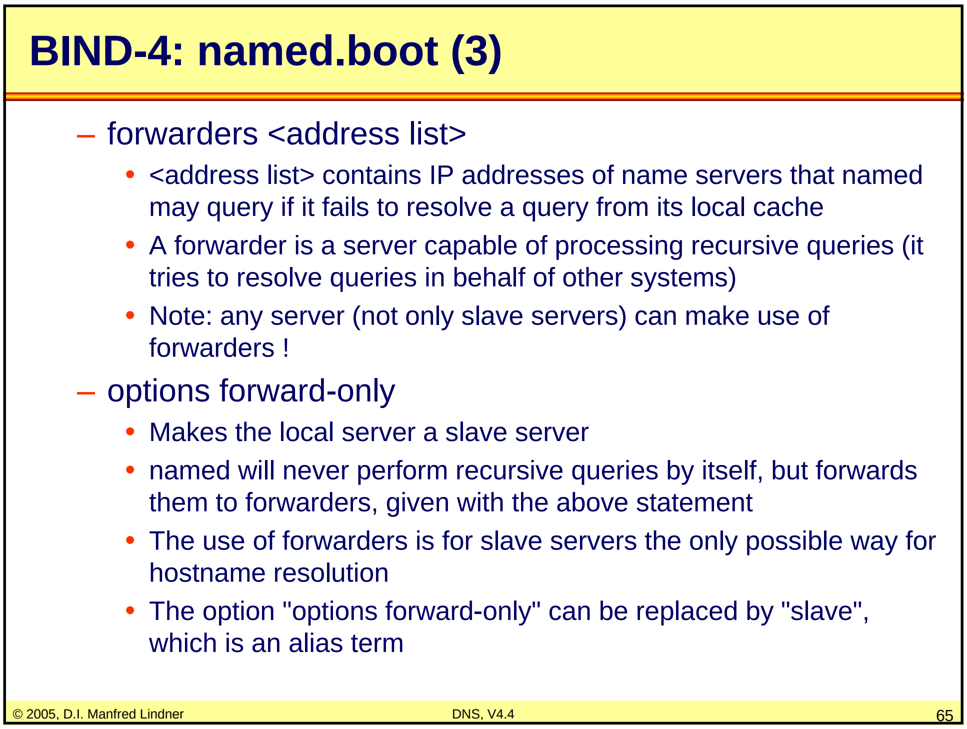
BIND-4: named.boot (1)

- **Contains boot-up information for named**
- **Options**
  - directory <directory>
    - Specifies the directory where the master files reside
  - primary <domain name> <file name>
    - Declares the local server authoritative for the named domain <domain name>
    - named loads the zone information from the given master file <file name>



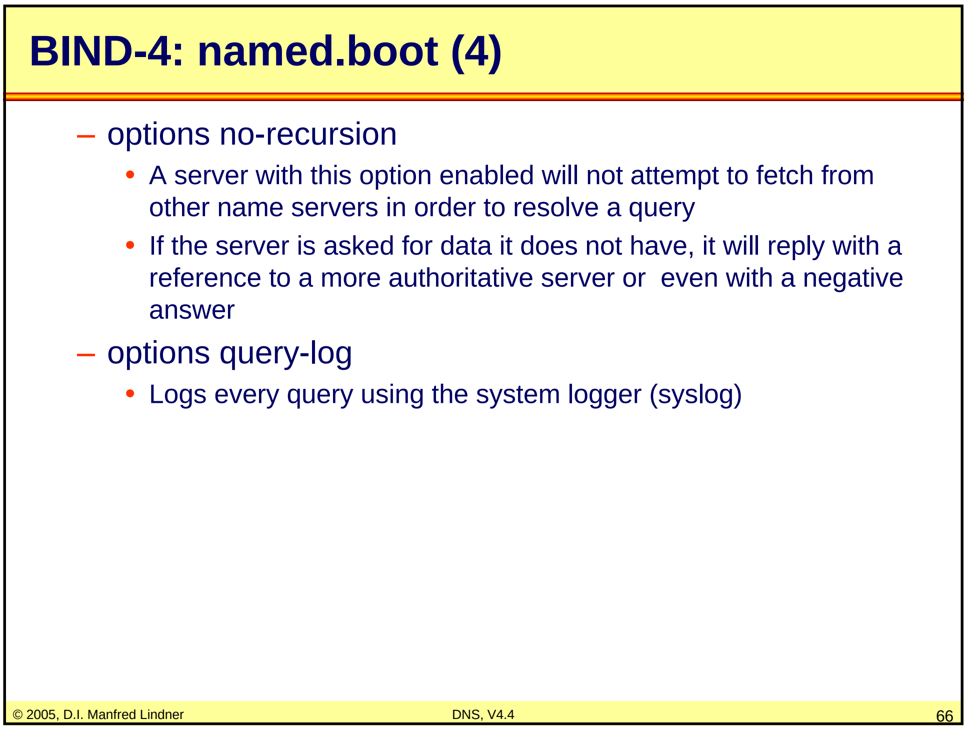
BIND-4: named.boot (2)

- secondary <domain name> <address list> <file name>
  - Declares the local server as secondary name server for the domain <domain name>
  - <address list> specifies IP addresses of primary servers for zone transfers; must contain at least one primary server
  - The local server will contact each of them in turn until a succesfull zone transfer could be completed
  - The received zone data is stores in the zone file <file name>
- cache <domain name> <file name>
  - The file <file name> contains the "root server hints", i.e. a list of records pointing to the root name servers - the begin of each query
  - The <domain name> is generally the root domain name "."
  - NOTE: if the cache option is not listed in the boot file, named will not create a local cache at all !!!



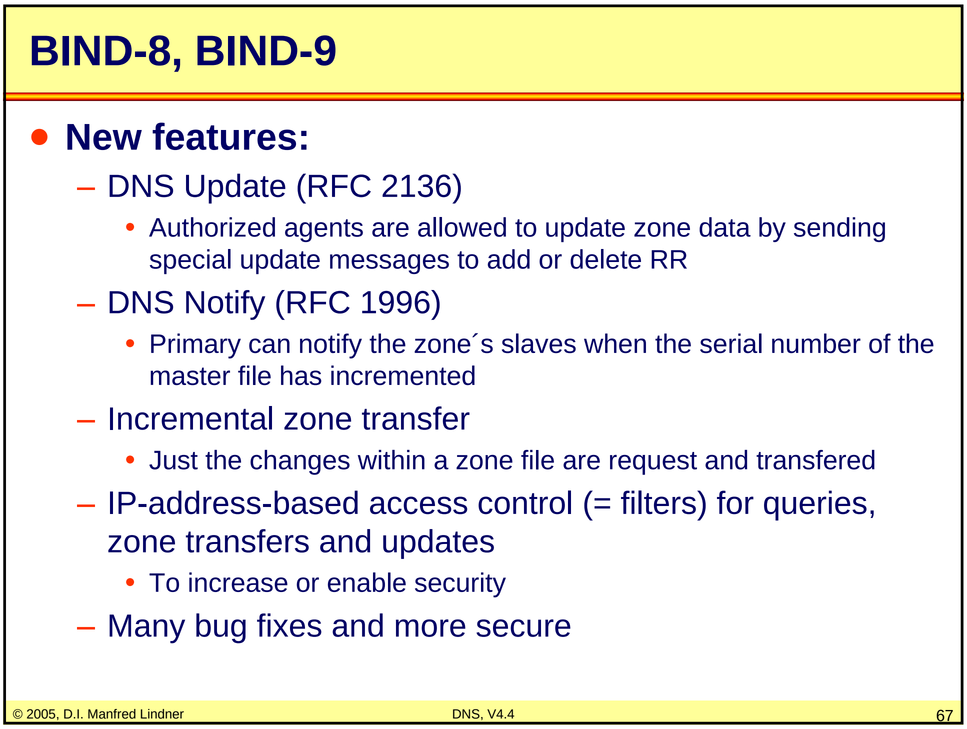
BIND-4: named.boot (3)

- forwarders <address list>
  - <address list> contains IP addresses of name servers that named may query if it fails to resolve a query from its local cache
  - A forwarder is a server capable of processing recursive queries (it tries to resolve queries in behalf of other systems)
  - Note: any server (not only slave servers) can make use of forwarders !
- options forward-only
  - Makes the local server a slave server
  - named will never perform recursive queries by itself, but forwards them to forwarders, given with the above statement
  - The use of forwarders is for slave servers the only possible way for hostname resolution
  - The option "options forward-only" can be replaced by "slave", which is an alias term



BIND-4: named.boot (4)

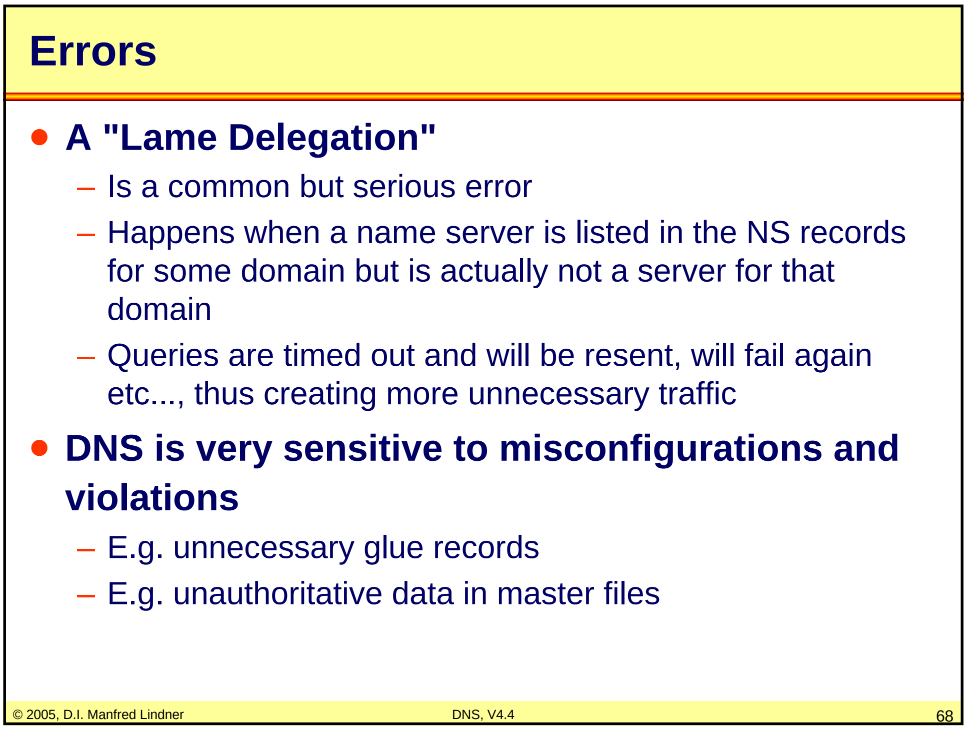
- options no-recursion
  - A server with this option enabled will not attempt to fetch from other name servers in order to resolve a query
  - If the server is asked for data it does not have, it will reply with a reference to a more authoritative server or even with a negative answer
- options query-log
  - Logs every query using the system logger (syslog)



BIND-8, BIND-9

- **New features:**
  - – DNS Update (RFC 2136)
    - Authorized agents are allowed to update zone data by sending special update messages to add or delete RR
  - – DNS Notify (RFC 1996)
    - Primary can notify the zone´s slaves when the serial number of the master file has incremented
  - – Incremental zone transfer
    - Just the changes within a zone file are request and transfered
  - – IP-address-based access control (= filters) for queries, zone transfers and updates
    - To increase or enable security
  - – Many bug fixes and more secure

© 2005, D.I. Manfred Lindner DNS, V4.4 67



Errors

- **A "Lame Delegation"**
  - – Is a common but serious error
  - – Happens when a name server is listed in the NS records for some domain but is actually not a server for that domain
  - – Queries are timed out and will be resent, will fail again etc..., thus creating more unnecessary traffic
- **DNS is very sensitive to misconfigurations and violations**
  - – E.g. unnecessary glue records
  - – E.g. unauthoritative data in master files

© 2005, D.I. Manfred Lindner DNS, V4.4 68

Diagnostic Tools

- **DIG - Domain Information Groper**
  - Send domain name query packets to name servers
  - Comand-line driven
  - Results are printed in a human-readable format
  - dig [@server] domain [<query-type>] [<query-class>] [+<query-option>] [-<dig-option>] [%comment]
- **NSLOOKUP**
  - Query Internet name servers interactively
  - More powerful utility as DIG

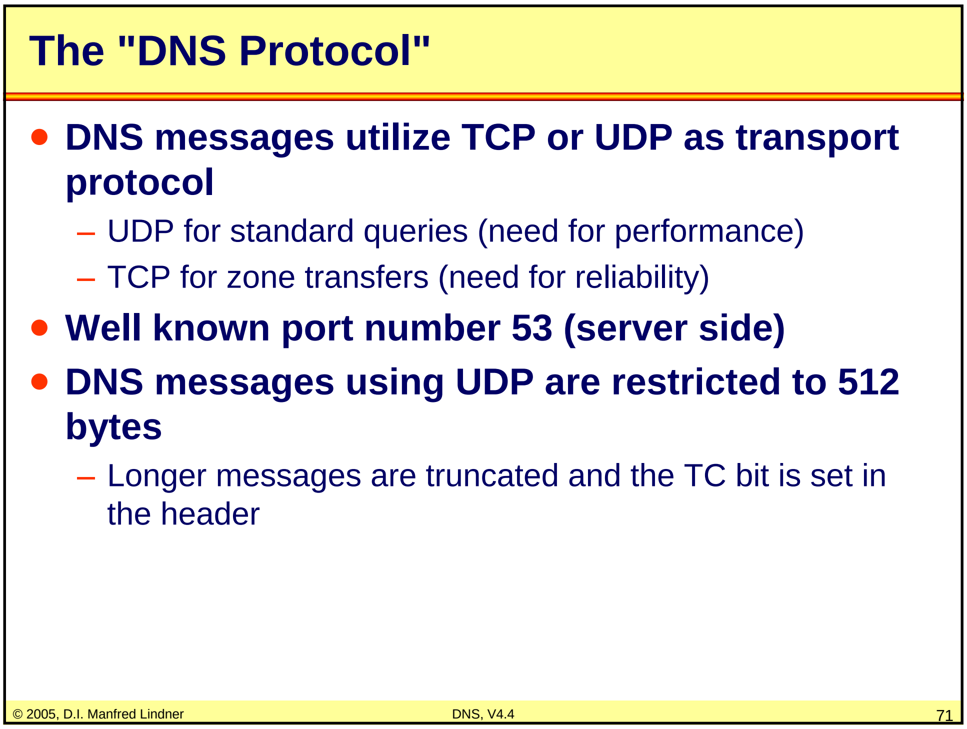
Agenda

Introduction

BIND

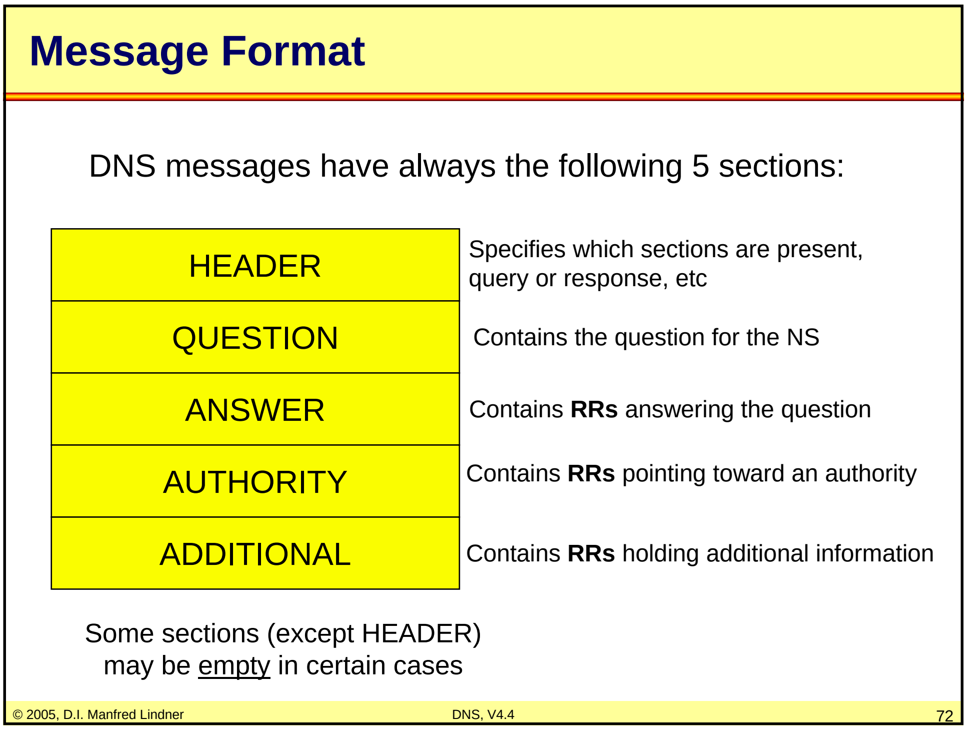
The DNS Protocol

- Message Format
- Header Format
- Question Format
- Domain Names
- Resource Records
- Message Compression
- DNS Query Example



The "DNS Protocol"

- **DNS messages utilize TCP or UDP as transport protocol**
  - UDP for standard queries (need for performance)
  - TCP for zone transfers (need for reliability)
- **Well known port number 53 (server side)**
- **DNS messages using UDP are restricted to 512 bytes**
  - Longer messages are truncated and the TC bit is set in the header

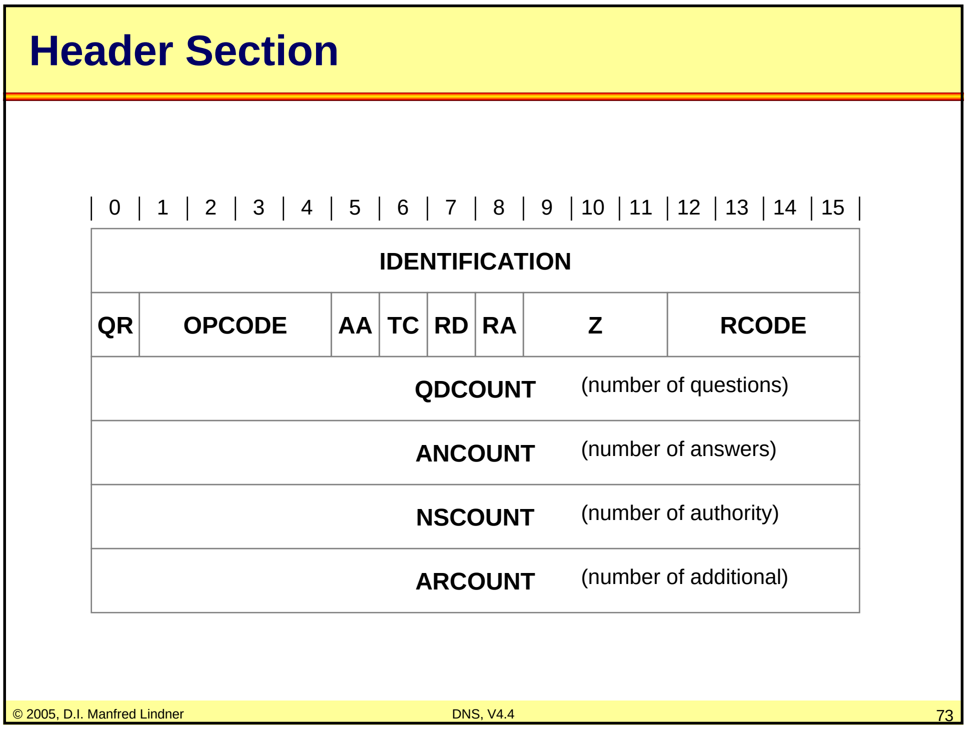


Message Format

DNS messages have always the following 5 sections:

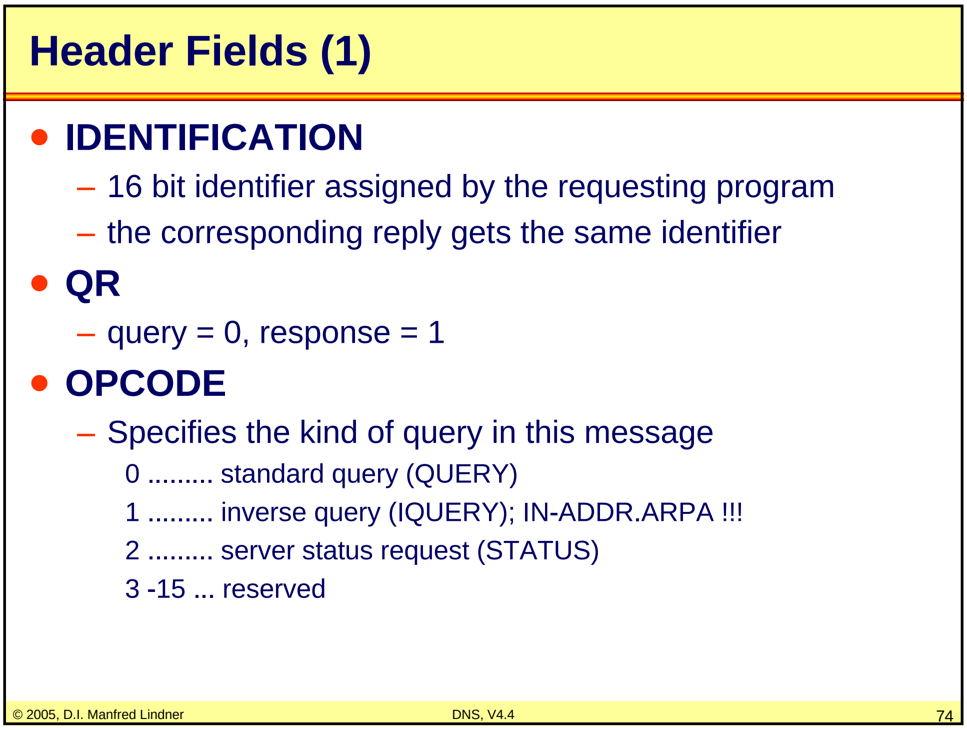
HEADER	Specifies which sections are present, query or response, etc
QUESTION	Contains the question for the NS
ANSWER	Contains **RRs** answering the question
AUTHORITY	Contains **RRs** pointing toward an authority
ADDITIONAL	Contains **RRs** holding additional information

Some sections (except HEADER) may be empty in certain cases



Header Section

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
IDENTIFICATION															
QR	**OPCODE**				**AA**	**TC**	**RD**	**RA**	**Z**			**RCODE**			
QDCOUNT (number of questions)															
ANCOUNT (number of answers)															
NSCOUNT (number of authority)															
ARCOUNT (number of additional)															



Header Fields (1)

- **IDENTIFICATION**
  - 16 bit identifier assigned by the requesting program
  - the corresponding reply gets the same identifier
- **QR**
  - query = 0, response = 1
- **OPCODE**
  - Specifies the kind of query in this message

    0 .... standard query (QUERY)

    1 .... inverse query (IQUERY); IN-ADDR.ARPA !!!

    2 .... server status request (STATUS)

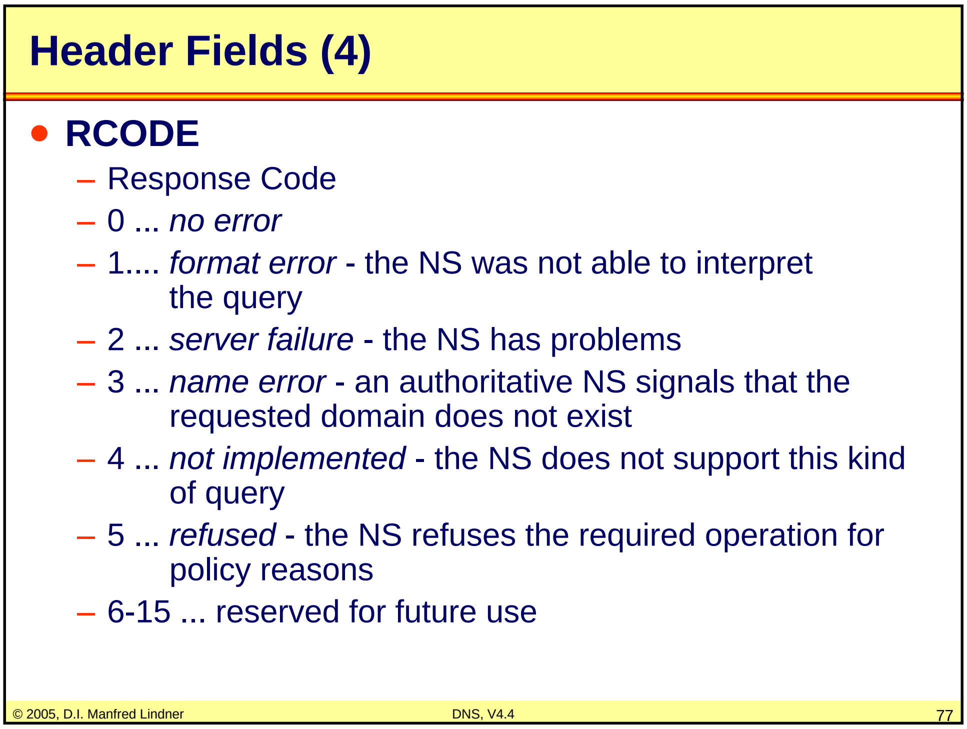
    3 -15 ... reserved

Header Fields (2)

- **AA**
  - Authoritative Answer
  - The responding NS is an authority for the domain name in the question section
  - If set, the data comes directly from a primary or secondary name server and not from a cache
- **TC**
  - TrunCation
  - Indicates that this message has been truncated (due to transmission channel's max message size)
- **RD**
  - Recursion Desired
  - The NS should solve the query recursively

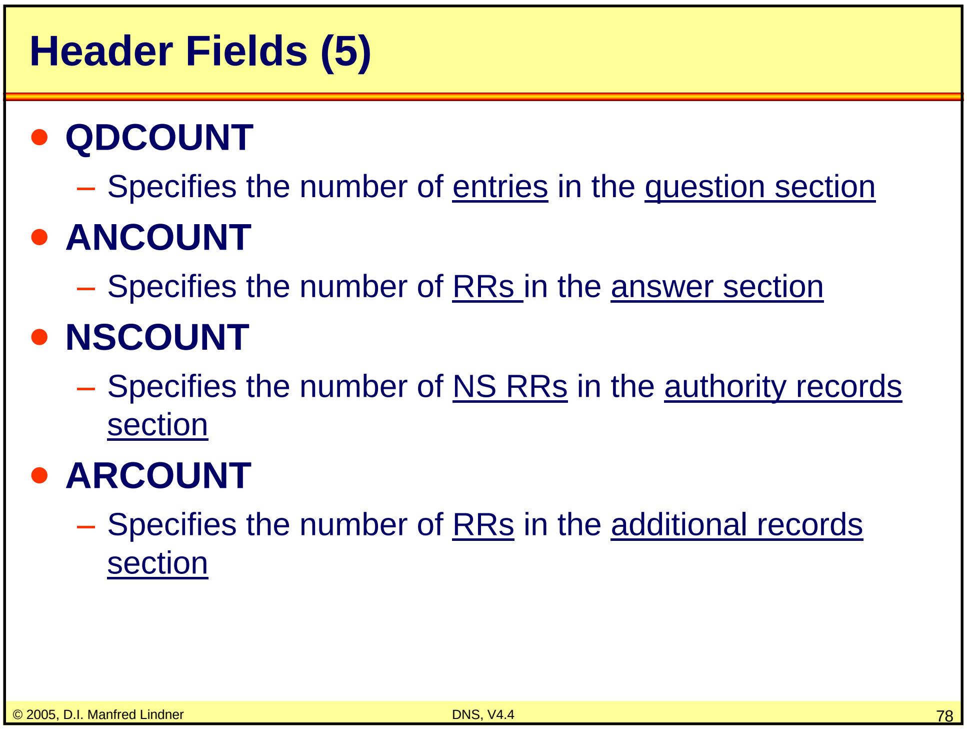
Header Fields (3)

- **RA**
  - Recursion Available
  - May be set or cleared in a response
  - Indicates whether recursive queries are supported by the NS
- **Z**
  - Reserved
  - Must be zero



Header Fields (4)

- **RCODE**
  - Response Code
  - 0 ... *no error*
  - 1.... *format error* - the NS was not able to interpret the query
  - 2 ... *server failure* - the NS has problems
  - 3 ... *name error* - an authoritative NS signals that the requested domain does not exist
  - 4 ... *not implemented* - the NS does not support this kind of query
  - 5 ... *refused* - the NS refuses the required operation for policy reasons
  - 6-15 ... reserved for future use

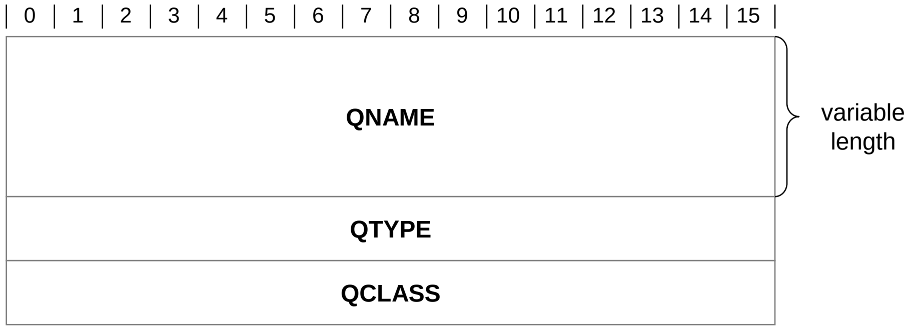


Header Fields (5)

- **QDCOUNT**
  - Specifies the number of entries in the question section
- **ANCOUNT**
  - Specifies the number of RRs in the answer section
- **NSCOUNT**
  - Specifies the number of NS RRs in the authority records section
- **ARCOUNT**
  - Specifies the number of RRs in the additional records section

Question Section

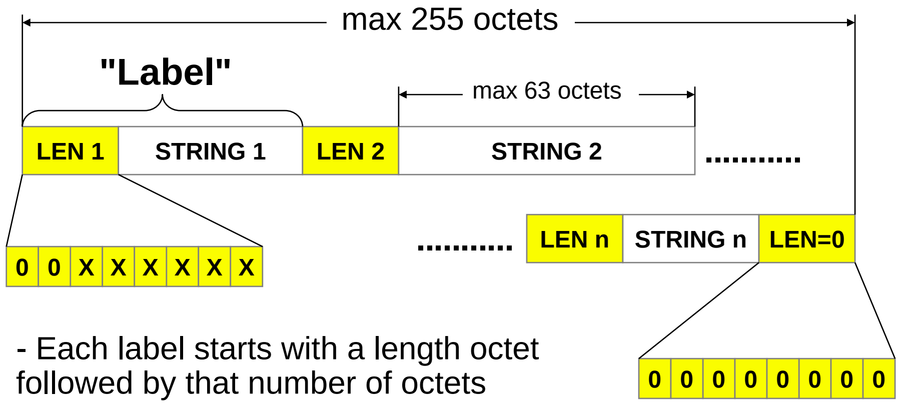
The question section contains QDCOUNT entries, each of the following format:



Question Fields

- **QNAME**
  - A domain name represented as a set of labels
    See the domain name message format below
  - Can have an odd number of octets, no padding is used as reminder
- **QTYPE**
  - Type of query; values are a superset of the TYPE values in RRs
    - AXFR (252) request for a transfer of the entire zone
    - " * " (255) request for all records
- **QCLASS**
  - Class of the query; values are a superset of the CLASS values in RRs (usually "IN" for Internet, " * " for any class)

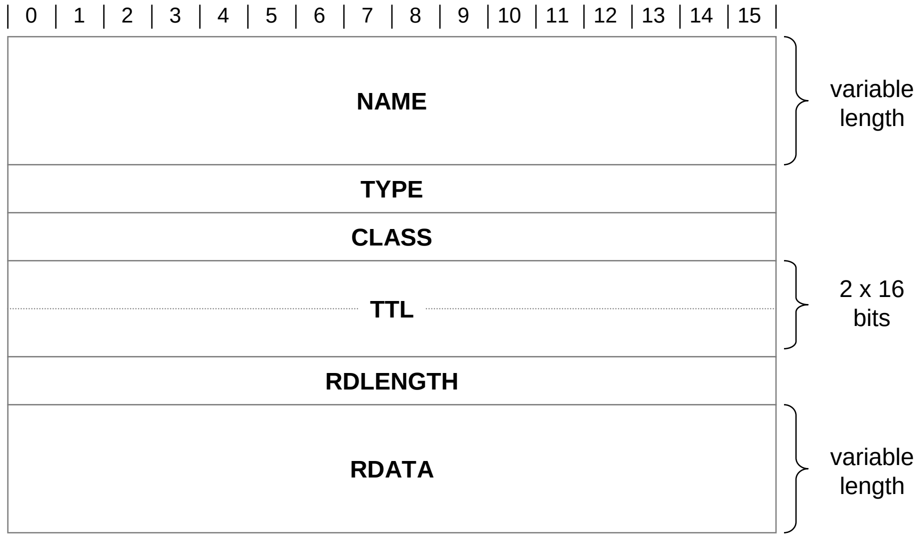
Domain Names in Messages

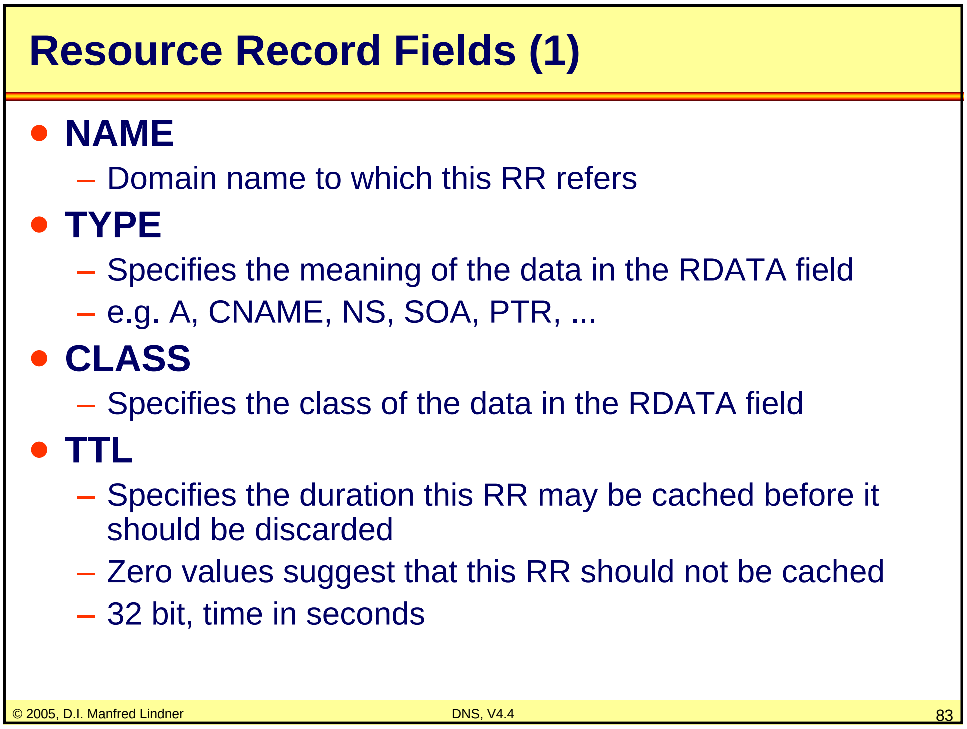


- Each label starts with a length octet followed by that number of octets

- The domain name is terminated with a zero length octet (= "null label" for the root)

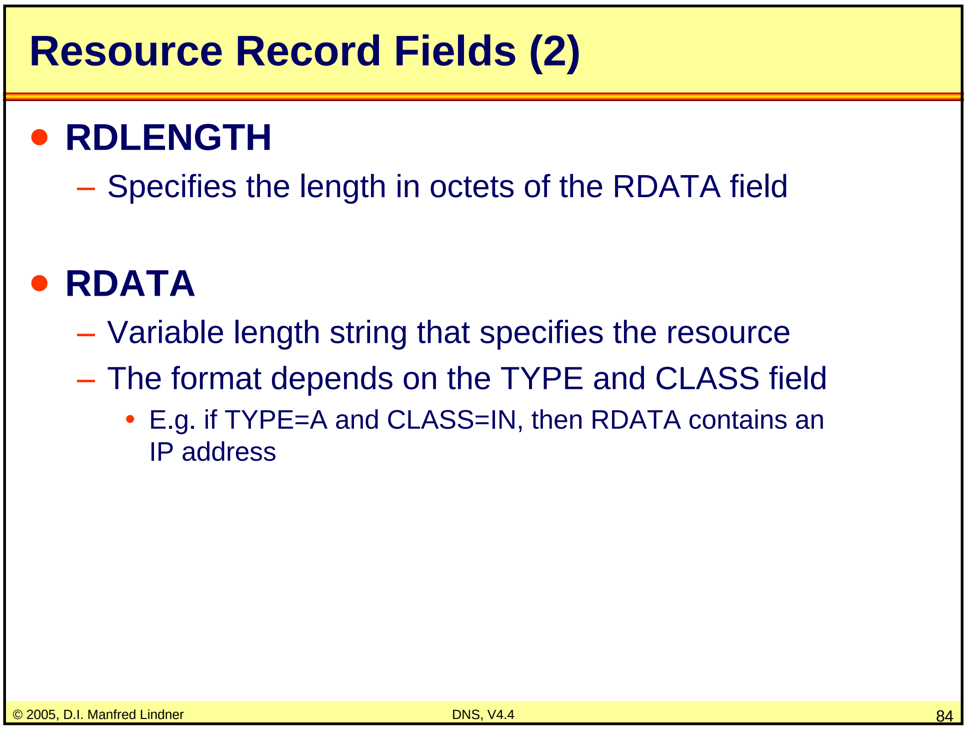
Resource Record Format in Answers, Authorative and Additional Fields





Resource Record Fields (1)

- **NAME**
  - Domain name to which this RR refers
- **TYPE**
  - Specifies the meaning of the data in the RDATA field
  - e.g. A, CNAME, NS, SOA, PTR, ...
- **CLASS**
  - Specifies the class of the data in the RDATA field
- **TTL**
  - Specifies the duration this RR may be cached before it should be discarded
  - Zero values suggest that this RR should not be cached
  - 32 bit, time in seconds



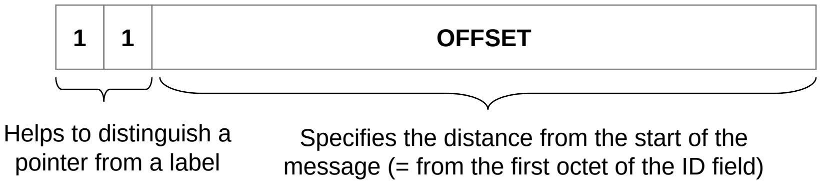
Resource Record Fields (2)

- **RDLENGTH**
  - Specifies the length in octets of the RDATA field
- **RDATA**
  - Variable length string that specifies the resource
  - The format depends on the TYPE and CLASS field
    - E.g. if TYPE=A and CLASS=IN, then RDATA contains an IP address

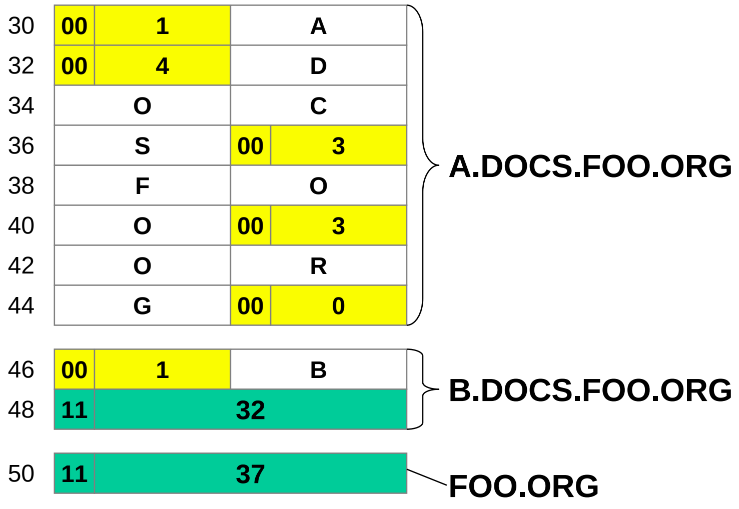
Message Compression

- To reduce the size of messages DNS provides a simple compression method
- Repetitions of domain names can be replaced with a pointer to the previous occurance
  - Works even for part of domain names (list of labels)

Pointer format:


1
1
OFFSET
Helps to distinguish a pointer from a label
Specifies the distance from the start of the message (= from the first octet of the ID field)

Message Compression Example

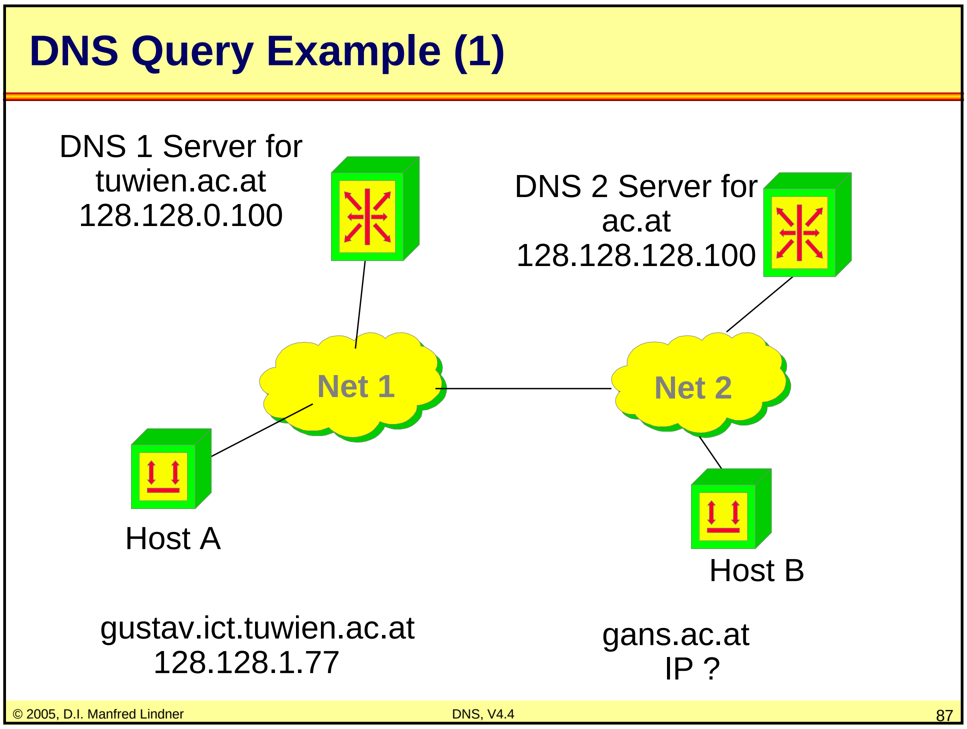


30	00	1	A	
32	00	4	D	
34	O		C	
36	S		00	3
38	F		O	
40	O		00	3
42	O		R	
44	G		00	0
46	00	1	B	
48	11	32		
50	11	37		

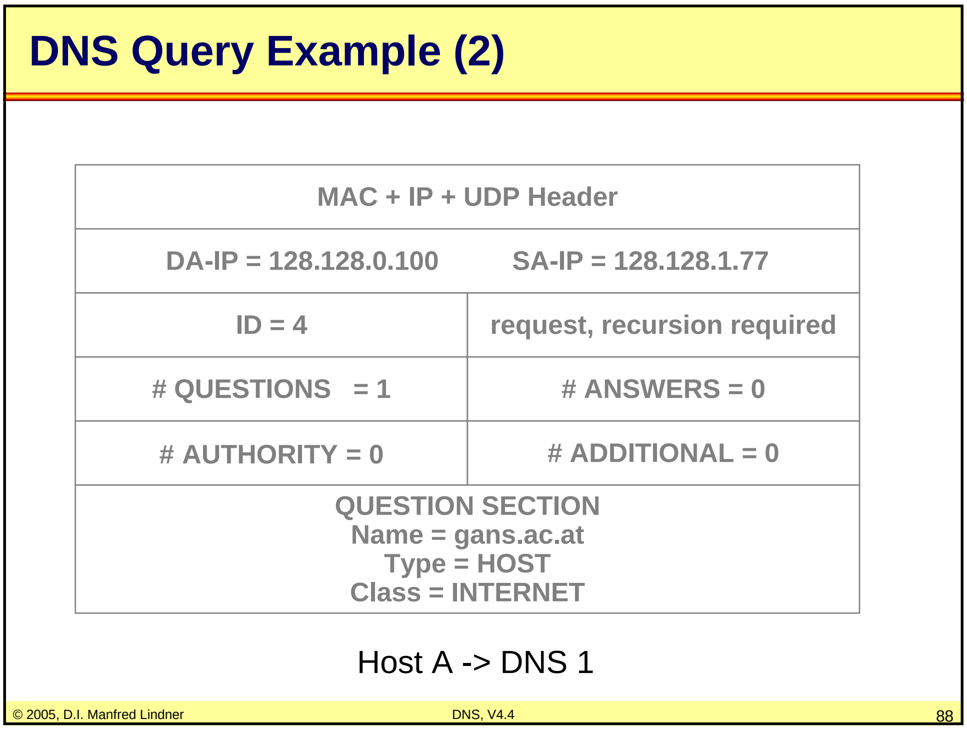
A.DOCS.FOO.ORG

B.DOCS.FOO.ORG

FOO.ORG



DNS Query Example (1)

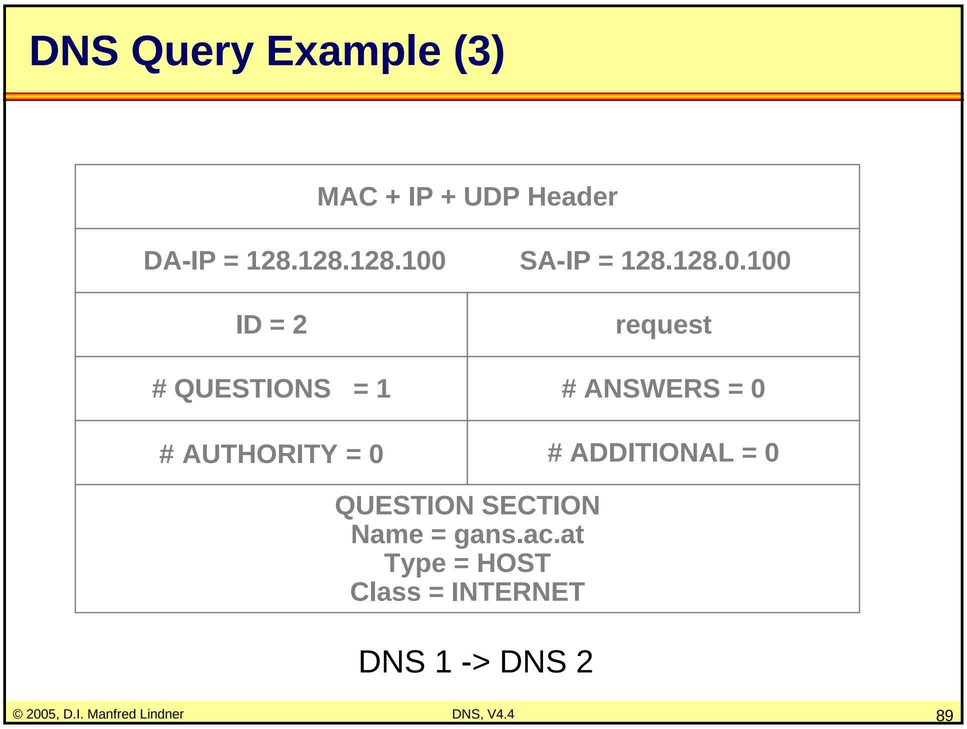


DNS Query Example (2)

MAC + IP + UDP Header	
DA-IP = 128.128.0.100	SA-IP = 128.128.1.77
ID = 4	request, recursion required
# QUESTIONS = 1	# ANSWERS = 0
# AUTHORITY = 0	# ADDITIONAL = 0

QUESTION SECTION
Name = gans.ac.at
Type = HOST
Class = INTERNET	

Host A -> DNS 1

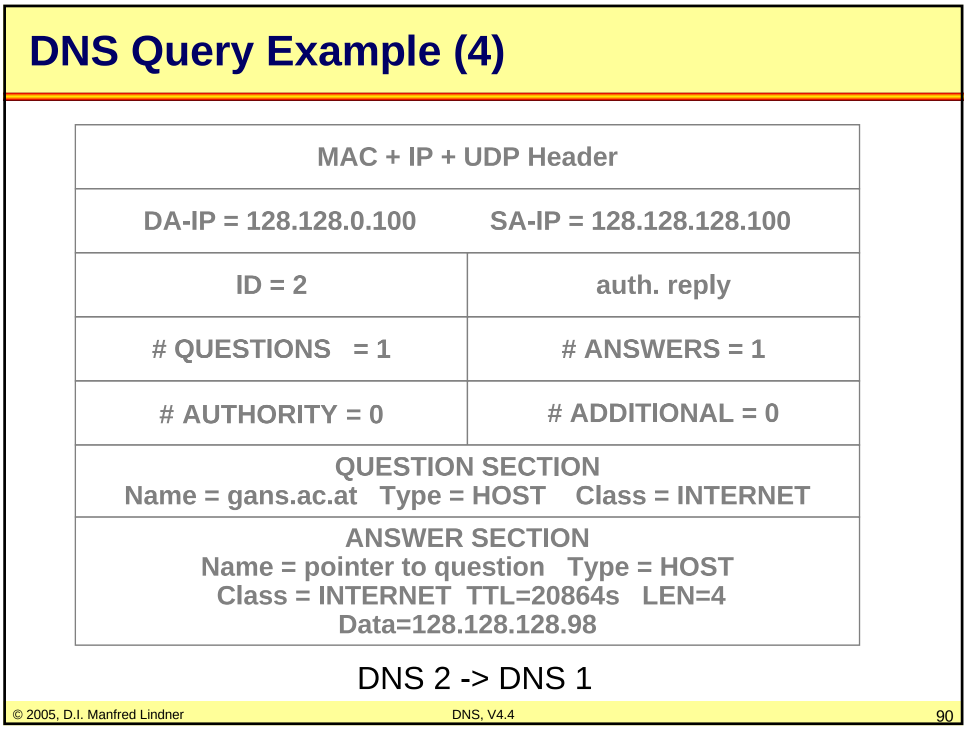


DNS Query Example (3)

MAC + IP + UDP Header	
DA-IP = 128.128.128.100	SA-IP = 128.128.0.100
ID = 2	request
# QUESTIONS = 1	# ANSWERS = 0
# AUTHORITY = 0	# ADDITIONAL = 0

QUESTION SECTION
Name = gans.ac.at
Type = HOST
Class = INTERNET	

DNS 1 -> DNS 2



DNS Query Example (4)

MAC + IP + UDP Header	
DA-IP = 128.128.0.100	SA-IP = 128.128.128.100
ID = 2	auth. reply
# QUESTIONS = 1	# ANSWERS = 1
# AUTHORITY = 0	# ADDITIONAL = 0
QUESTION SECTION Name = gans.ac.at Type = HOST Class = INTERNET	

ANSWER SECTION
Name = pointer to question Type = HOST
Class = INTERNET TTL=20864s LEN=4
Data=128.128.128.98	

DNS 2 -> DNS 1

DNS Query Example (5)

MAC + IP + UDP Header	
DA-IP = 128.128.1.77	SA-IP = 128.128.0.100
ID = 4	auth. reply, recursion avail.
# QUESTIONS = 1	# ANSWERS = 1
# AUTHORITY = 0	# ADDITIONAL = 0
QUESTION SECTION Name = gans.ac.at Type = HOST Class = INTERNET	

ANSWER SECTION
Name = pointer to question Type = HOST
Class = INTERNET TTL=20864s LEN=4
Data=128.128.128.98	

DNS 1 -> Host A

Selected RFCs (1)

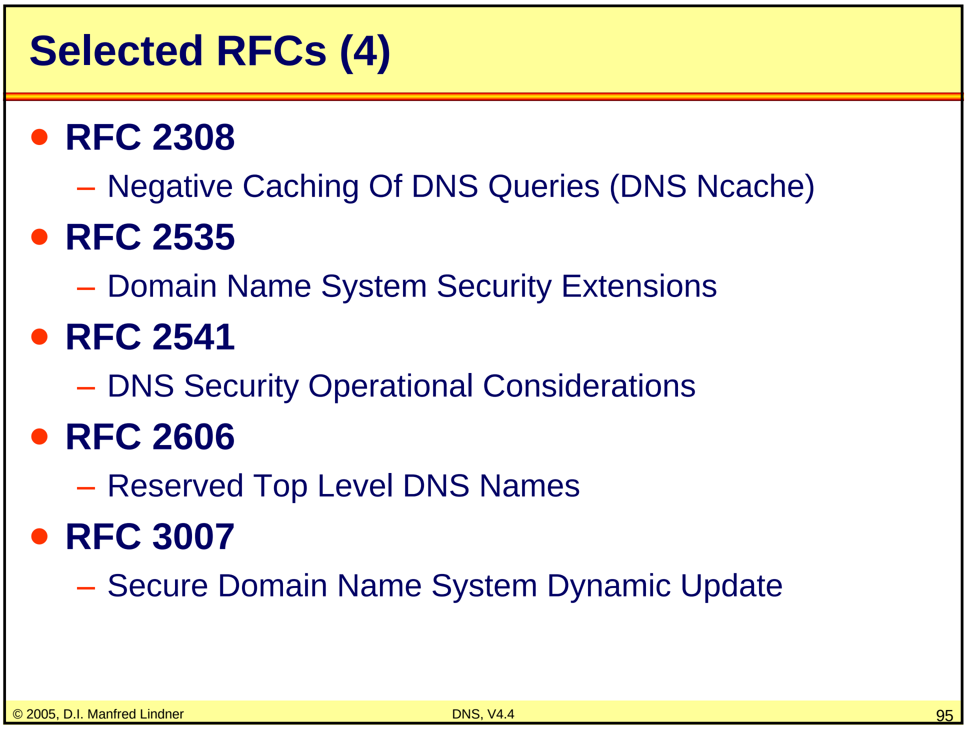
- RFC 1034
  - Domain Name Concept And Facilities
- RFC 1035
  - Domain Name Implementation and Specification
- RFC 1101
  - DNS Encoding Network Names And Other Types
- RFC 1183
  - New DNS RR Definitions
- RFC 1591
  - Domain Name System Structure And Delegation

Selected RFCs (2)

- **RFC 1664**
  - Using The Internet DNS To Distribute RFC1327 Mail Address Mapping Tables
- **RFC 1712**
  - DNS Encoding Of Geographical Location
- **RFC 1788**
  - ICMP Domain Name Messages
- **RFC 1794**
  - DNS Support For Load Balancing
- **RFC 1995**
  - Incremental Zone Transfers In DNS

Selected RFCs (3)

- **RFC 1996**
  - A Mechanism For Prompt Notification Of Zone Changes (DNS Notify)
- **RFC 2052**
  - A DNS RR For Specifying The Location Of Services (DNS SRV)
- **RFC 2065**
  - Domain Name System Security Extensions
- **RFC 2136**
  - Dynamic Updates In The Domain Name System (DNS Update)



Selected RFCs (4)

- **RFC 2308**
  - Negative Caching Of DNS Queries (DNS Ncache)
- **RFC 2535**
  - Domain Name System Security Extensions
- **RFC 2541**
  - DNS Security Operational Considerations
- **RFC 2606**
  - Reserved Top Level DNS Names
- **RFC 3007**
  - Secure Domain Name System Dynamic Update