Distributed Systems
COMP 212

Lecture 17

Othon Michail

UNIVERSITY OF
LIVERPOOL

Naming Technologies within Distributed Systems

Names, Identifiers and Addresses

Computers vs. Humans

- Entities (Internet hosts, routers, file systems, services) are accessed using identifiers (numbers)
  - IP address
  - File descriptor
  - Port number …
- Humans like to use meaningful names
  - www.csc.liv.ac.uk
  - My Documents
  - “Dater” (from our Java RMI example)
- We need services that bind names and identifiers

Identifiers

Special type of (usually, computer readable) names with the following properties:

- An id refers to at most one entity
- Each entity is referred by at most one id
- An id always refers to the same entity (never reused)

An identifier includes or can be transformed to an address for an object

- e.g. NFS file handle, Java RMI remote object reference, etc

Names

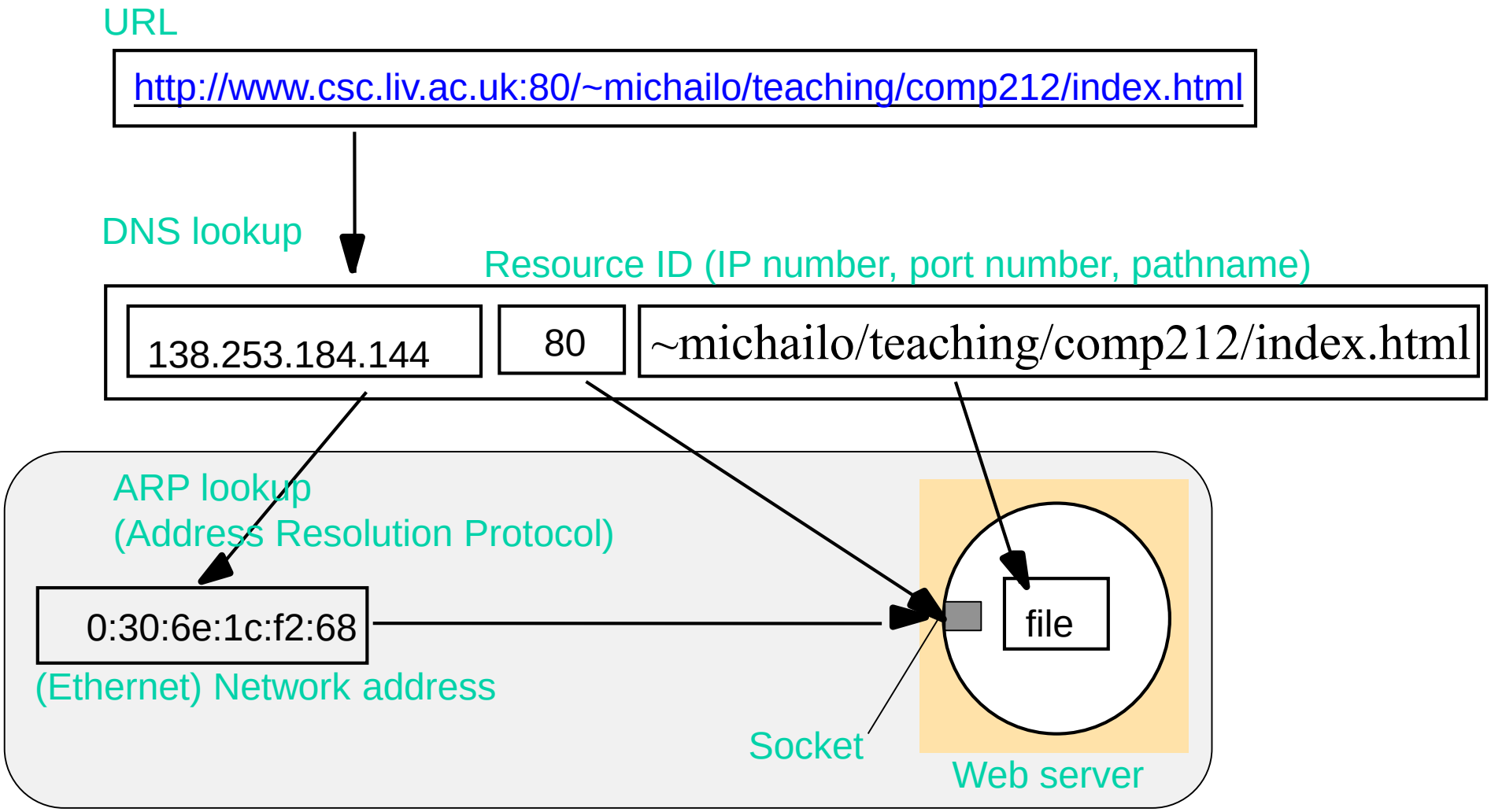
- A name is human-readable value (usually a string) that can be resolved to an identifier or address
  - Internet domain name, file pathname, process number
    - E.g. /etc/passwd, http://www.csc.liv.ac.uk/
- For many purposes, names are preferable to identifiers
  - because the binding of the named resource to a physical location is deferred and can be changed
  - because they are more meaningful to users
- Resource names are resolved by name services
  - to give identifiers and other useful attributes
  - e.g., DNS

Uniform Resource Locator (URL)

Used for identifying resources in the Internet

- Typed by the protocol field (http, ftp, nfs, etc)
- Part of the name is service-specific
- Resources cannot be moved between domains

Name Resolution



Namespaces

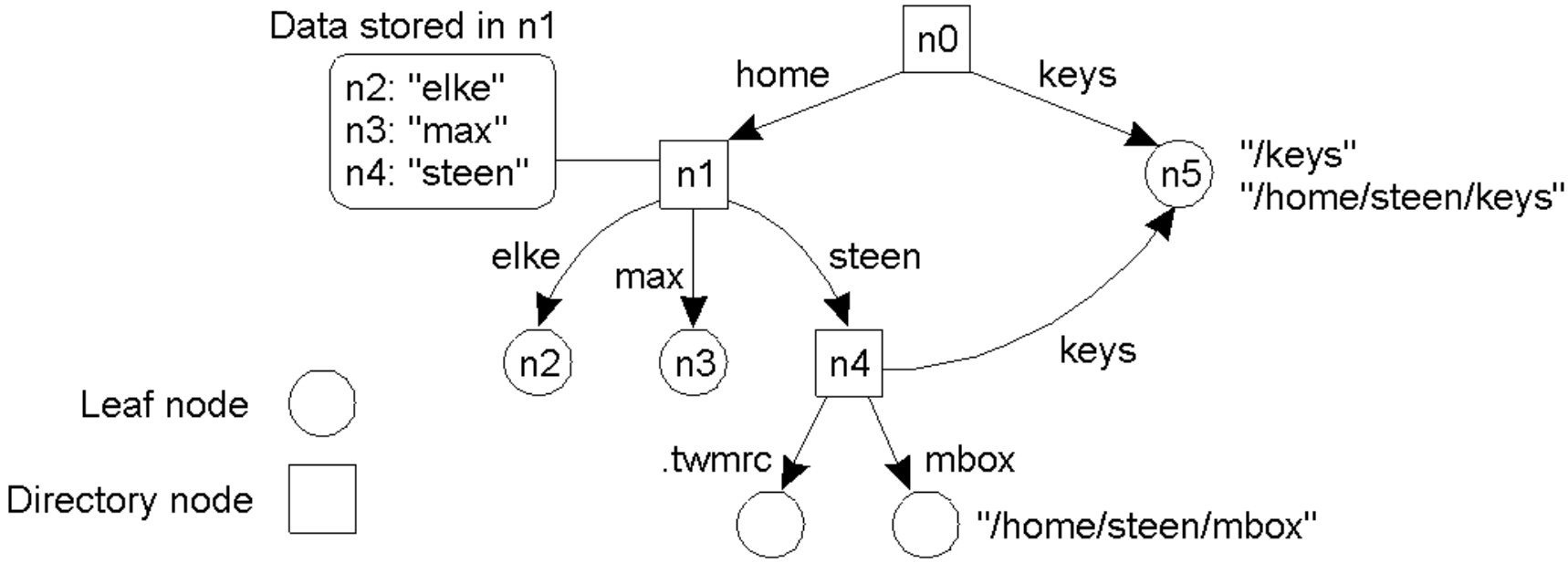
Names are often organised into namespaces

- Allow simple but meaningful names to be used
- Potentially infinite number of names
- Structured
  - to allow similar subnames without clashes
  - to group related names
- Allow re-structuring of names
  - for some types of change, old programs should continue to work
- Management of trust

Name Graphs

- Within distributed systems, a namespace is represented by a labelled, directed graph with two types of nodes:
  - leaf nodes: information on an entity
  - directory nodes: a collection of named outgoing edges (which can lead to any other type of node)
- Each namespace has at least one root node
- Nodes can be referred to by path names (with absolute or relative)
- File systems are a classic example …

Example 1: File System



- Unix File system is a classic example

Variations

- Global vs local name
  Global name denotes the same entity (always interpreted with respect to the same directory node)
  Local name its interpretation depends on where the name is being used
- More than one path to a node
- Naming graph: tree (hierarchical), more than one root, Directed Acyclic Graph (DAG)

Merging Name Spaces

- Merging name spaces problem: we have different name spaces that we wish to access from any given name space
- Possible Solution: Mounting
- A directory node called a mount point stores the id of (or all the necessary information for identifying and accessing) a directory node from a foreign name space called mounting point

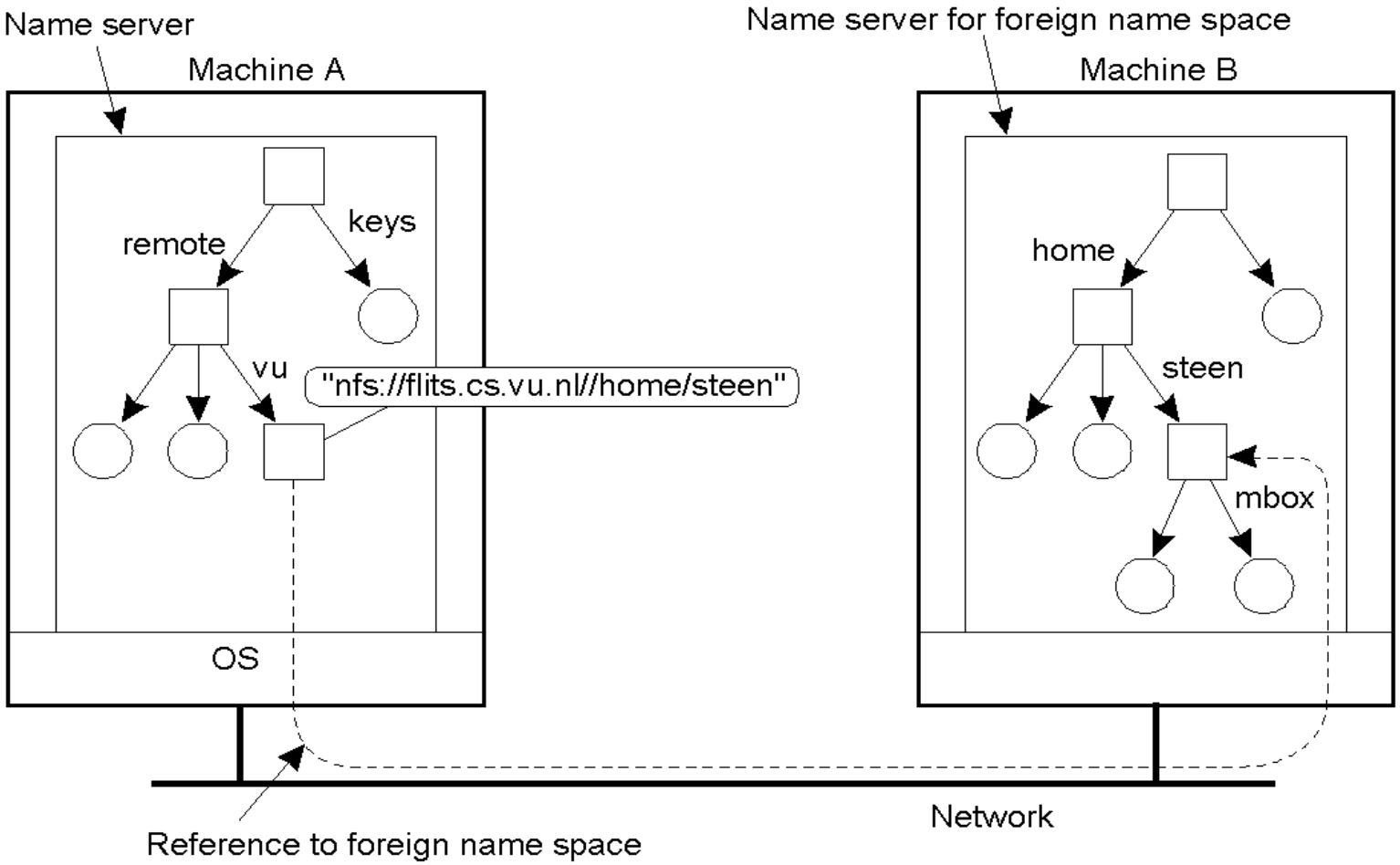
Mounting (1)

The following information is required:

1. Name of the access protocol (resolved to the implementation of a communication protocol)
2. Name of the server (resolved to an address where the server can be reached)
3. Name of the mounting point (resolved to a node in the foreign name space, by the foreign server)

e.g.
nfs://flits.cs.vu.nl//home/steen

Mounting (2)



- Mounting remote name spaces through a specific process protocol (in this case Sun's Network File System protocol - NFS)

Example 2: Internet Name Server

- A good naming server should provide
  - Scalability
  - Decentralised maintenance
  - Robustness, fault-tolerance
  - Global scope
    - Names mean the same thing everywhere

Why not Centralised Server?

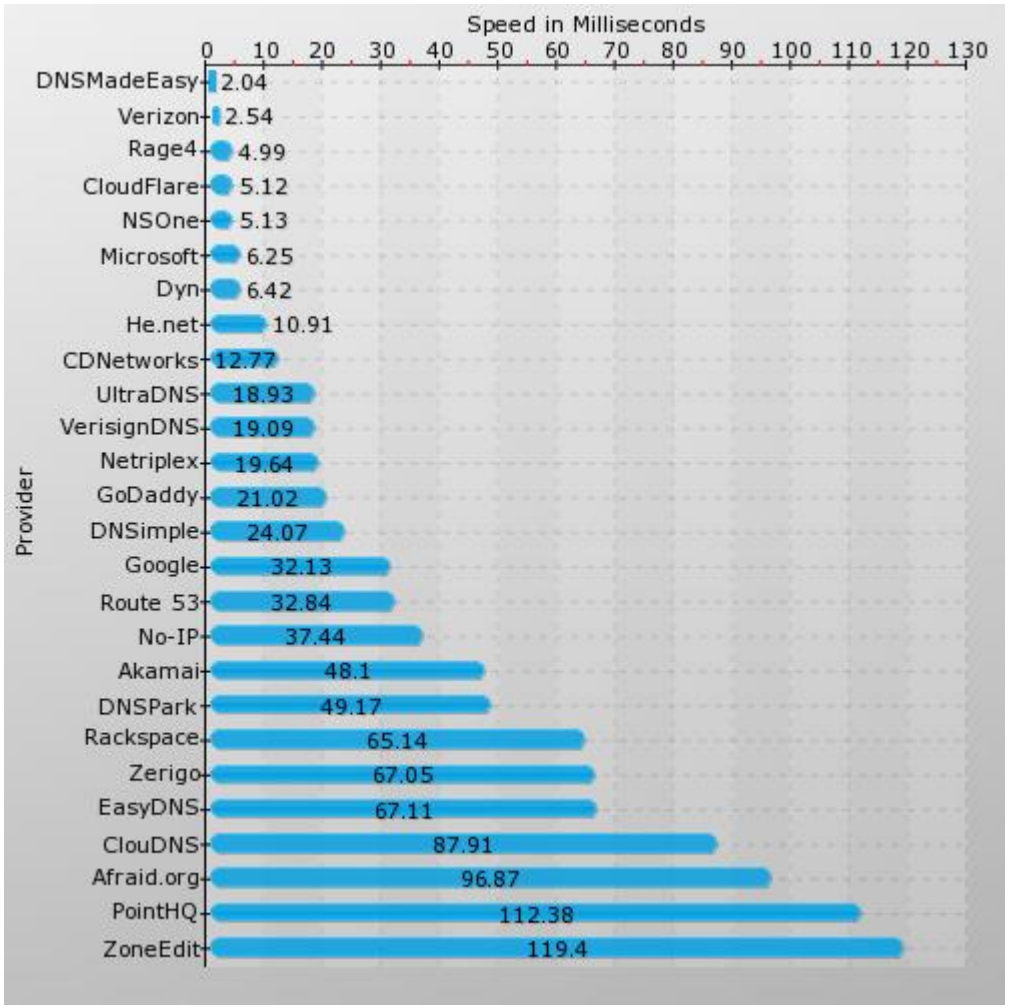
- Single server with all name-to-IP address mappings
  - single point of failure
  - traffic volume
  - distant centralised database (performance)
  - maintenance
  - doesn’t scale!

DNS
(Domain Name System)

DNS (Domain Name System)

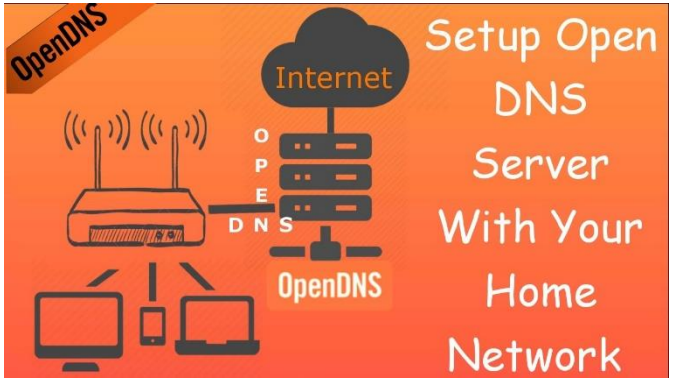
- Distributed database implemented in hierarchy of many name servers
- Decentralised control and management of data
- Application-layer protocol used by hosts and name servers
  - Communicate to resolve names (name/address translation)
  - Core Internet function implemented as application-layer protocol

DNS Providers Examples

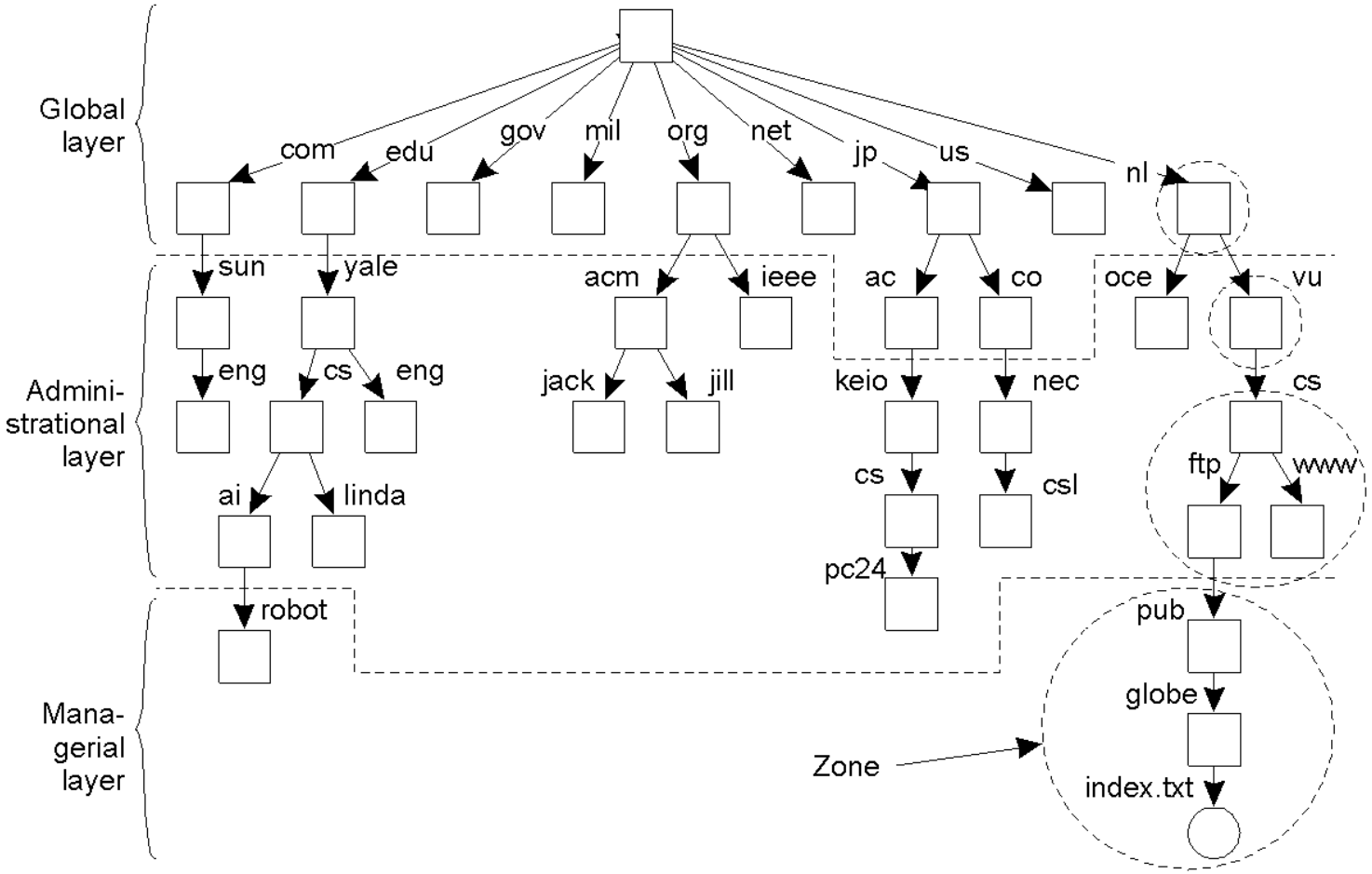


DNS MADE EASY

Google™ DNS
8.8.8.8
8.8.4.4



DNS Name Space



- An example partitioning of the DNS name space, including Internet-accessible files, into the three name space layers
- A zone in DNS is a non-overlapping part of the namespace that is implemented by a separate name server

DNS Name Servers

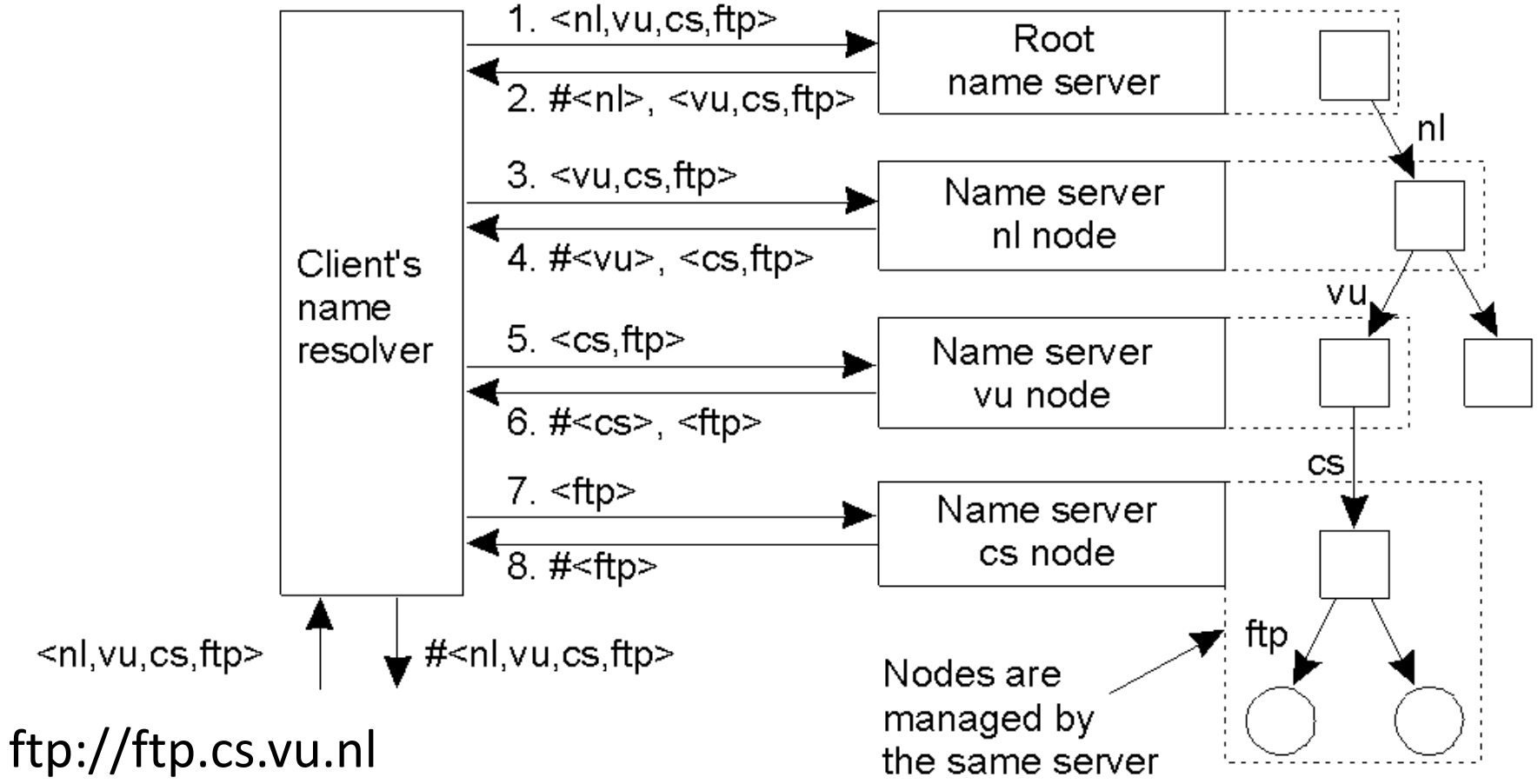
- Authoritative name servers store parts of the database
- Names assigned to authoritative name servers
  - For a host, authority stores that host's IP address, name
  - Responds to queries for host's IP address
  - Perform name/address translation for that host's name
- Root name server knows authoritative servers for particular subdomains
  - Hierarchically organises authoritative name servers
  - Reserving a domain gives you control of entry in root name server for particular names

DNS Lookup

1. Iterative Name Resolution.
   - Server responds with as much as it knows (i.e. name of server to contact next)
     - "I don't know this name, but ask this server"
   - Client iteratively queries additional servers

2. Recursive Name Resolution.
   - Server goes out and searches for more info on behalf of the client (recursive)
   - Only returns final answer or "not found"
   - Puts burden of name resolution on contacted name server

Iterative Name Resolution



ftp://ftp.cs.vu.nl

- The name resolver queries each name server (at each layer) in an iterative fashion
- Note: the client is doing all the work here (and generating a lot of traffic, too)

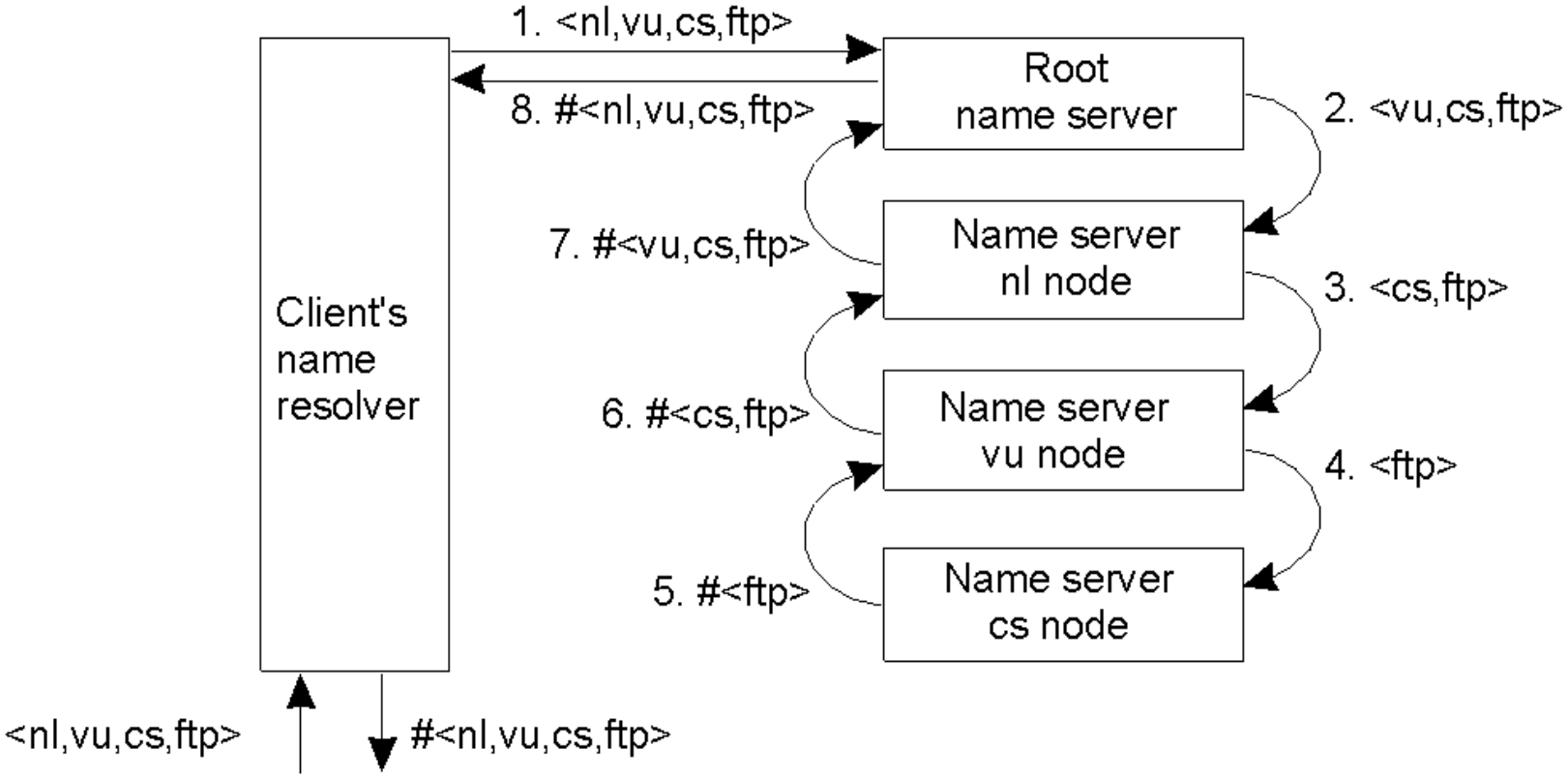
Iterative Name Resolution

```
; <<>> DiG 9.9.5-3ubuntu0.9-Ubuntu <<>> +trace www.liv.ac.uk
;; global options: +cmd
.                       336735  IN      NS      h.root-servers.net.
.                       336735  IN      NS      k.root-servers.net.
.                       336735  IN      NS      i.root-servers.net.
.                       336735  IN      NS      l.root-servers.net.
.                       336735  IN      NS      m.root-servers.net.
.                       336735  IN      NS      d.root-servers.net.
.                       336735  IN      NS      j.root-servers.net.
.                       336735  IN      NS      f.root-servers.net.
.                       336735  IN      NS      g.root-servers.net.
.                       336735  IN      NS      e.root-servers.net.
.                       336735  IN      NS      c.root-servers.net.
.                       336735  IN      NS      b.root-servers.net.
.                       336735  IN      NS      a.root-servers.net.
;; Received 239 bytes from 127.0.1.1#53(127.0.1.1) in 263 ms

uk.                     172800  IN      NS      dns1.nic.uk.
uk.                     172800  IN      NS      nsd.nic.uk.
uk.                     172800  IN      NS      nsa.nic.uk.
uk.                     172800  IN      NS      nsb.nic.uk.
uk.                     172800  IN      NS      dns2.nic.uk.
uk.                     172800  IN      NS      nsc.nic.uk.
uk.                     172800  IN      NS      dns3.nic.uk.
uk.                     172800  IN      NS      dns4.nic.uk.
uk.                     86400   IN      DS      43876 8 2 A107ED2AC1BD14D924173B
C7E827A1153582072394F9272BA37E2353 BC659603
uk.                     86400   IN      RRSIG   DS 8 1 86400 20170308050000 2017
0223040000 61045 . g7tL4lKHd87lKdgqTLhWFOqbjrAWB4y0FD6LiO7SToAJxjnpn7i4qoMR XuS5
TMLxlzpOP6d9CakMs2gsXFrPZl3oXARqeYcx7qm45Tx78RJNPsGe vw5Cg8FFwke8efOkiOzdumYchZG
q/PCdmkKJo7HWdzWVdxExMpcfsRPU pVtWTXufeNBjU5ixonJfdIttDRWj0Y1mHeuZK2VR0u43wE2cUI
lppXKm AA466/ecGnIxOyHj8M+07/IuXnQeqp7OcCHgc75HdaOh02WjDzRqKbqy QKYAGiQPO2OuspHC
0BgYQTv8SR5WHG/MH0ryB54KBoGAONciEn51hsU9 aZRkgQ==
;; Received 797 bytes from 192.112.36.4#53(g.root-servers.net) in 123 ms
```

Iterative Name Resolution

```
ac.uk.                  172800  IN      NS      ns0.ja.net.
ac.uk.                  172800  IN      NS      ns1.surfnet.nl.
ac.uk.                  172800  IN      NS      ns2.ja.net.
ac.uk.                  172800  IN      NS      ns3.ja.net.
ac.uk.                  172800  IN      NS      ns4.ja.net.
ac.uk.                  172800  IN      NS      auth03.ns.uu.net.
ac.uk.                  172800  IN      NS      ws-fra1.win-ip.dfn.de.
ac.uk.                  300     IN      DS      45874 8 2 A1DC9AF78C8BF46CF074F6
0E5EC1CBDB5C29A40991FF54C78ABDD3B9 9CD49D2F
ac.uk.                  300     IN      RRSIG   DS 8 2 300 20170309054004 201702
23053444 43056 uk. sY9g+sLYxGO4I7EOv9VF8VtomX5Ea07h9HXwAmSITNit6VdKRlLKRaX/ oUJv
a8Ky2La25G3eFs/VTg0uubSNAxYJ96zjn2Io0WPrn9plKOsCcHAd wND4UfZ3qRWvEZhFOwa3/6BD+D7
x+umON4bF//JKZDyE1SmZkNdfEa7D oAo=
;; Received 420 bytes from 213.248.220.1#53(dns3.nic.uk) in 102 ms

liv.ac.uk.              86400   IN      NS      ns0.york.ac.uk.
liv.ac.uk.              86400   IN      NS      dns1.liv.ac.uk.
liv.ac.uk.              86400   IN      NS      dns0.liv.ac.uk.
liv.ac.uk.              86400   IN      NS      dir.mcc.ac.uk.
46HLCJCI11G8N2F6REST7JKSVUI0A0IH.ac.uk. 14400 IN NSEC3 1 0 10 - 46RPD722C6J62M1J
BDBEL5GROILH0BUO NS
46HLCJCI11G8N2F6REST7JKSVUI0A0IH.ac.uk. 14400 IN RRSIG NSEC3 8 3 14400 201703241
40708 20170222140708 62709 ac.uk. r/T0GbuMH+P1cQig/kDQ0aV0O5ajW2jBms92KPJkgMn93b
dUI7PKye+i e70bYYN0Za6o6CID6Ql+cPJBsAzWzajB8CVfGosKrsR4D8up0lXepJ4q UBPYapgONxQB
f9KSvZXwBd1or5S2ZronLstN75GBrCFXD5hwXVprRPkr lOY=
;; Received 440 bytes from 128.86.1.20#53(ns0.ja.net) in 926 ms

www.liv.ac.uk.          86400   IN      CNAME   vs-www.liv.ac.uk.
vs-www.liv.ac.uk.       86400   IN      A       138.253.13.50
;; Received 79 bytes from 138.253.31.3#53(dns1.liv.ac.uk) in 1 ms
```

Recursive Name Resolution



- The name resolver starts the process, then each server temporarily becomes a client of the next name server until the resolution is satisfied. The results are then returned to the client.

Caching and Recursive Name Resolution

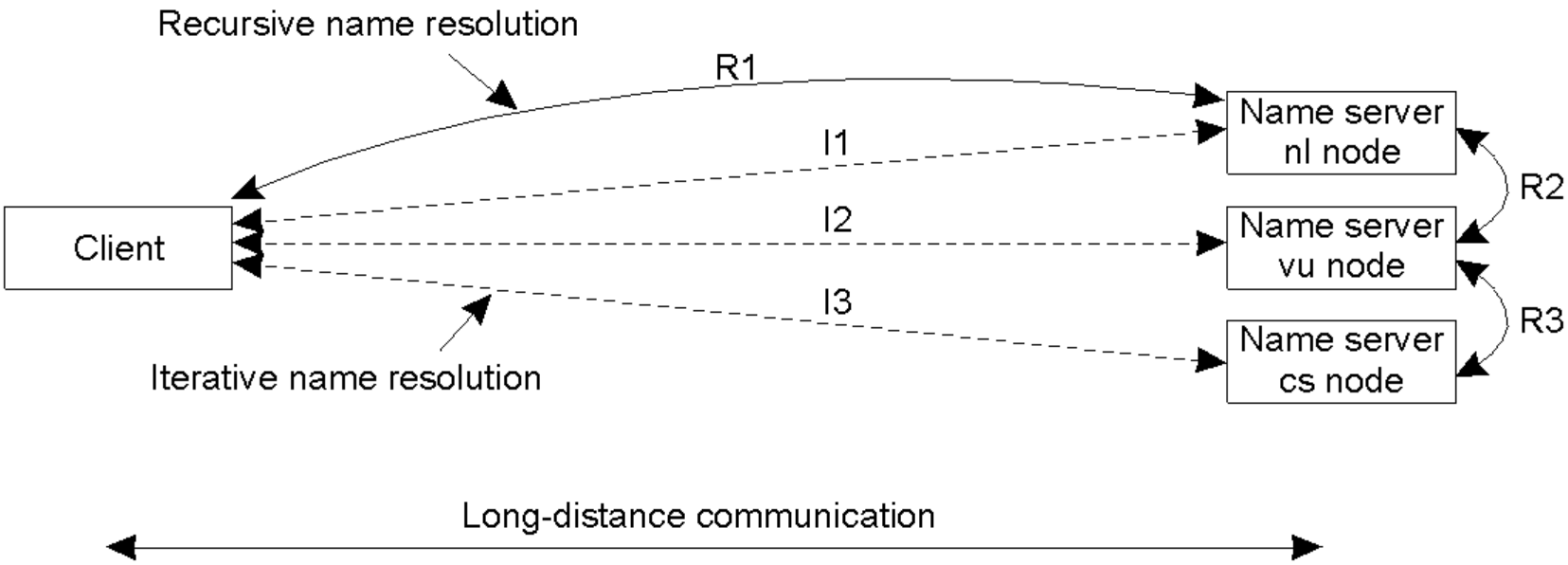
Server for node	Should resolve	Looks up	Passes to child	Receives and caches	Returns to requester
cs	<ftp>	#<ftp>	--	--	#<ftp>
vu	<cs,ftp>	#<cs>	<ftp>	#<ftp>	#<cs> #<cs, ftp>
ni	<vu,cs,ftp>	#<vu>	<cs,ftp>	#<cs> #<cs,ftp>	#<vu> #<vu,cs> #<vu,cs,ftp>
root	<ni,vu,cs,ftp>	#<nl>	<vu,cs,ftp>	#<vu> #<vu,cs> #<vu,cs,ftp>	#<nl> #<nl,vu> #<nl,vu,cs> #<nl,vu,cs,ftp>

- Recursive name resolution of <nl, vu, cs, ftp>. Name servers cache intermediate results for subsequent lookups. This is seen as a key advantage to the recursive name resolution approach, even though the workload has been moved from the client to the servers. Nevertheless, think about subsequent lookups …

DNS Caching

- DNS responses are cached
  - Quick response for repeated translations
  - Other queries may reuse some parts of lookup
    - NS records for domains
- DNS negative queries are also cached
  - Don’t have to repeat past mistakes, e.g. misspellings
- Cached data periodically times out
  - Lifetime (TTL: Time To Live) of data controlled by owner of data
  - TTL passed with every record
  - TTL affects DNS-based load balancing techniques
- Name tables change infrequently, but when they do, caching can result in the delivery of stale data

Iterative vs. Recursive Resolution



- The comparison between recursive and iterative name resolution with respect to communication costs. Again, the recursive technology is generally regarded to have an advantage in this situation (especially over longer, more expensive WAN links).

DNS Terminology

- A subtree within DNS is referred to as a domain
- A path name is referred to as a domain name
- These can be relative or absolute
- A DNS server operates at each node (except those at the bottom)
- The information at nodes is organised into resource records

DNS functions

- Main function is to resolve domain names for computers, i.e. to get their IP addresses
  - caches the results of previous searches until they pass their 'time to live'
- Other functions:
  - get mail host for a domain
  - reverse resolution - get domain name from IP address
  - Host information - type of hardware and OS
  - Well-known services - a list of well-known services offered by a host
  - Other attributes can be included (optional)

Reverse Lookup

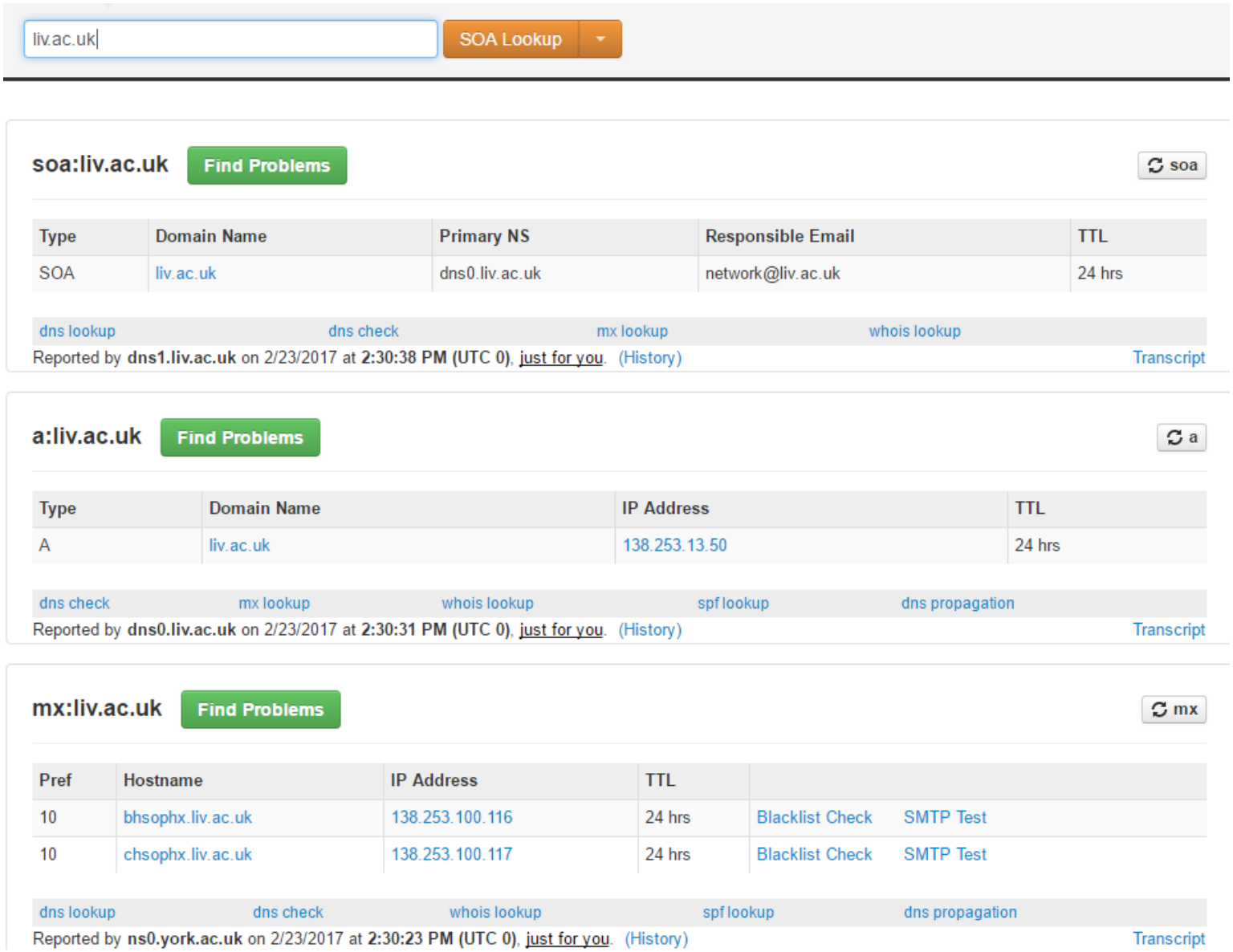
```
othon@othon-Inspiron-5547:~$ host 138.253.13.50
50.13.253.138.in-addr.arpa domain name pointer vs-www.liv.ac.uk.
othon@othon-Inspiron-5547:~$ nslookup 138.253.13.50
Server:         127.0.1.1
Address:        127.0.1.1#53

50.13.253.138.in-addr.arpa      name = vs-www.liv.ac.uk.

othon@othon-Inspiron-5547:~$ dig +noall +answer -x 138.253.13.50
50.13.253.138.in-addr.arpa. 86400 IN    PTR     vs-www.liv.ac.uk.
othon@othon-Inspiron-5547:~$
```

DNS – Types of Resource Record

Type of record	Associated entity	Description
SOA	Zone	Holds information on the represented zone.
A	Host	Contains an IP address of the host this node represents.
MX	Domain	Refers to a mail server to handle mail addressed to this node.
SRV	Domain	Refers to a server handling a specific service.
NS	Zone	Refers to a name server that implements the represented zone.
CNAME	Node	Symbolic link with the primary name of the represented node.
PTR	Host	Inverse mapping of IP address to host names
HINFO	Host	Holds information on the host this node represents.
TXT	Any kind	Contains any entity-specific information considered useful.

- The most important types of resource records forming the contents of nodes (and maintained by servers) in the DNS name space

Resource Record Lookups



DNS Implementation

Name	Record type	Record value
cs.vu.nl	SOA	star (1999121502,7200,3600,2419200,86400)
cs.vu.nl	NS	star.cs.vu.nl
cs.vu.nl	NS	top.cs.vu.nl
cs.vu.nl	NS	solo.cs.vu.nl
cs.vu.nl	TXT	"Vrije Universiteit - Math. & Comp. Sc."
cs.vu.nl	MX	1 zephyr.cs.vu.nl
cs.vu.nl	MX	2 tornado.cs.vu.nl
cs.vu.nl	MX	3 star.cs.vu.nl
star.cs.vu.nl	HINFO	Sun Unix
star.cs.vu.nl	MX	1 star.cs.vu.nl
star.cs.vu.nl	MX	10 zephyr.cs.vu.nl
star.cs.vu.nl	A	130.37.24.6
star.cs.vu.nl	A	192.31.231.42
zephyr.cs.vu.nl	HINFO	Sun Unix
zephyr.cs.vu.nl	MX	1 zephyr.cs.vu.nl
zephyr.cs.vu.nl	MX	2 tornado.cs.vu.nl
zephyr.cs.vu.nl	A	192.31.231.66
www.cs.vu.nl	CNAME	soling.cs.vu.nl
ftp.cs.vu.nl	CNAME	soling.cs.vu.nl
soling.cs.vu.nl	HINFO	Sun Unix
soling.cs.vu.nl	MX	1 soling.cs.vu.nl
soling.cs.vu.nl	MX	10 zephyr.cs.vu.nl
soling.cs.vu.nl	A	130.37.24.11
laser.cs.vu.nl	HINFO	PC MS-DOS
laser.cs.vu.nl	A	130.37.30.32
vucs-das.cs.vu.nl	PTR	0.26.37.130.in-addr.arpa
vucs-das.cs.vu.nl	A	130.37.26.0

- An excerpt from the DNS database for the zone *cs.vu.nl*.
- The “database” is a small collection of files maintained within each DNS “zone”.

Directory Services

Directory Service

- Directory service:- 'yellow pages' for the resources in a network
  - Retrieves the set of names that satisfy a given description
  - e.g. X.500, LDAP, MS Active Directory Services
    - (DNS holds some descriptive data, but:
      - the data is very incomplete
      - DNS isn't organised to search it)

What Are Directory Services?

- All Directory services use a hierarchical structure that stores information about objects on the network
- What differentiates the various implementations are the types of objects that they track

X.500 and LDAP

- **X.500**
  - Global directory service framework defined by sets of international standards
  - Published by ISO and ITU (International Telecommunication Union)
  - Provides cataloguing service used to arrange information across sites
- **LDAP (Lightweight Directory Access Protocol)**
  - Open standard for directory services
  - Protocol sets for accessing information directories
  - Simpler functions than X.500
  - Queries smaller, quicker response times, lower network traffic

The X.500 Name Space (1)

Attribute	Abbr.	Value
Country	C	NL
Locality	L	Amsterdam
Organization	O	Vrije Universiteit
OrganizationalUnit	OU	Math. & Comp. Sc.
CommonName	CN	Main server
Mail_Servers	--	130.37.24.6, 192.31.231,192.31.231.66
FTP_Server	--	130.37.21.11
WWW_Server	--	130.37.21.11

- A simple example of a X.500 directory entry using X.500 naming conventions.

Practical Use

- Managing large computer networks is a nightmare
- Who are the users?
  - Privileges, profiles, policies,…
- What resources are available?
  - Different default printers in different rooms,…
- Settings for various applications
  - Network settings: default gateway, mail server, firewall…
- Hierarchical organisation
- …

Microsoft Active Directory

- Active directory uses LDAP to send queries to servers
- Active Directory Services Interface (ADSI) is a connector used with the API for LDAP
- ADSI designed to interoperate with other directory service products
- Active directory uses DNS as a locator service to resolve domain, site and services names to an IP address

Other Implementations

- eDirectory Novel's implementation. Support Windows, NetWare, Linux and some Unixes
- Netscape Directory Service
  - RedHat directory service
  - Fedora directory service
- Apache Directory Service
  - Integrates with Java
- OpenLDAP
- …

Discovery Service

- Discovery service is a directory service that also:
  - is automatically updated as the network configuration changes
  - meets the needs of clients in spontaneous networks
  - discovers services required by a client (who may be mobile) within the current scope, for example, to find the most suitable printing service for image files after arriving at a hotel.
  - Examples of discovery services: Jini discovery service (now called Apache River), the 'service location protocol', the 'simple service discovery protocol' (part of UPnP), the 'secure discovery service'.