Lecture 9: Name and Directory Servers

CDK4: Chapter 9

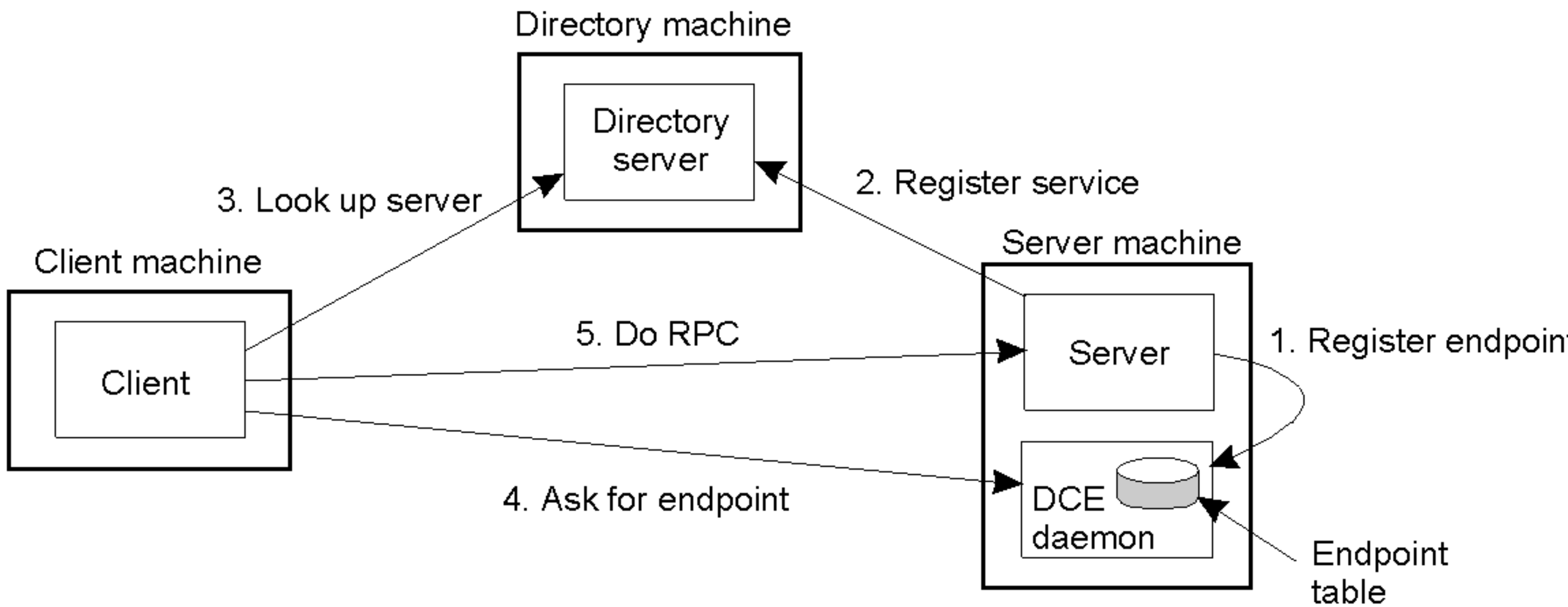
CDK5: Chapter 13

TVS: Chapter 5

Binding RPC Client to Server

- Not desirable to hardwire the machine name and port number used by a server into a client
- Use a directory server instead (to find machine) – directory server machine known
- Use a local daemon on that machine to find port to use (daemon uses known port)

Binding a Client to a Server



TVS: Figure 4-13 Client-to-server binding in a Distributed Computing Environment (endpoint – commonly known as a port)

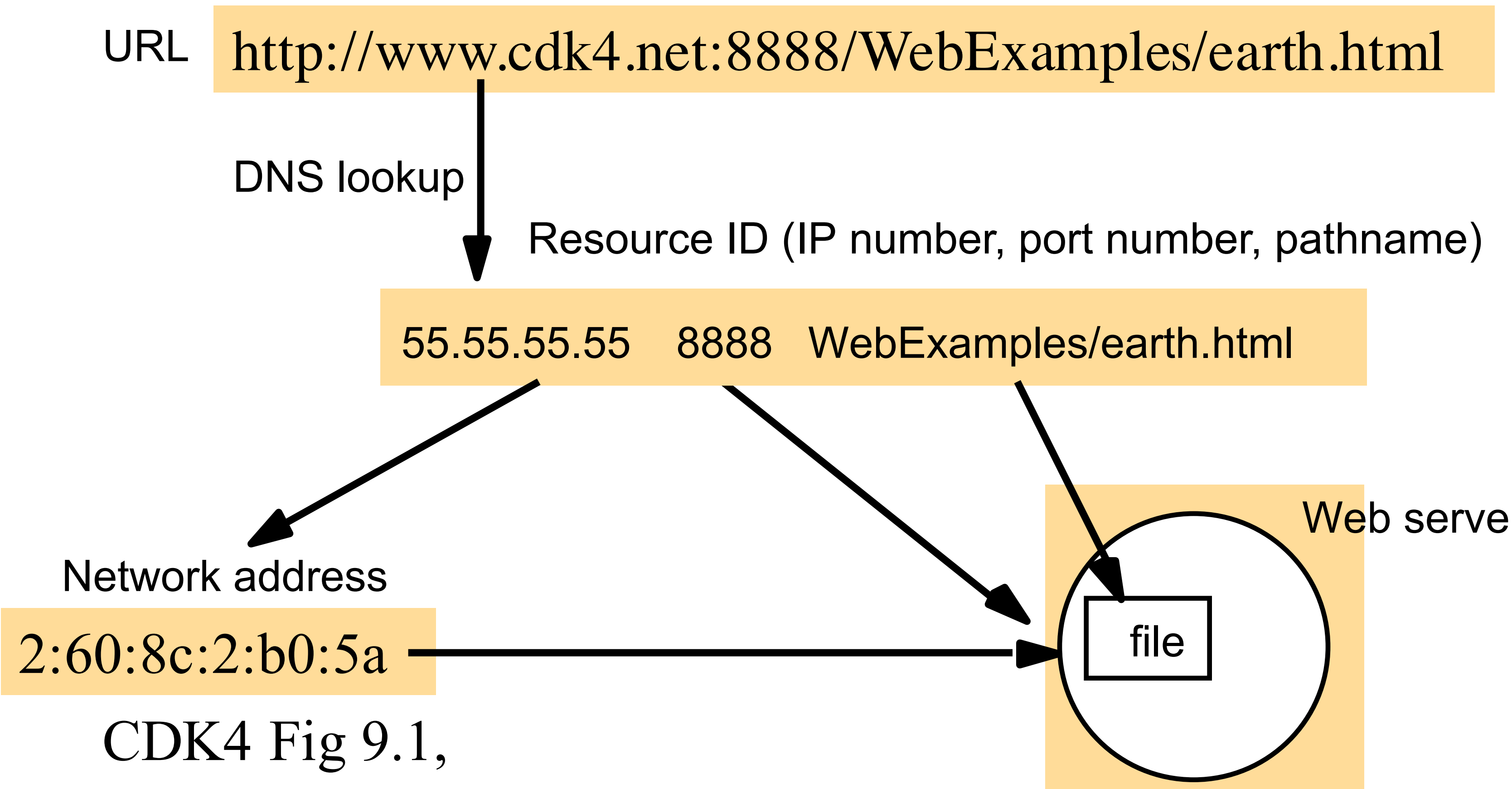
Names

- *Pure* names contain no information about the item they are associated with (Needham)
- Other names may either tell you what sort of object they refer to, or something about where it can be found
- An address is an extreme example of a non-pure name

Name resolution

- A name is *resolved* when it is translated into data about the item
- Names are *bound* to *attributes* (i.e. values of properties – such as addresses!)
- A name has a namespace or domain …
- You can compose names to make bigger ones (e.g. URLs)

Composed naming domains used to access a resource from a URL



CDK4 Fig 9.1,
CDK5 Fig 13.1

URIs, URLs, and URNs

- Uniform Resource Identifiers identify resources on the Web. Start by identifying the URI scheme – e.g. http:, ftp:, etc.
- Uniform Resource Locators – a subset of URIs which give a location for a resource
- Uniform Resource Names – URIs which are not URLs, e.g. urn:ISBN:0-201-62433-8

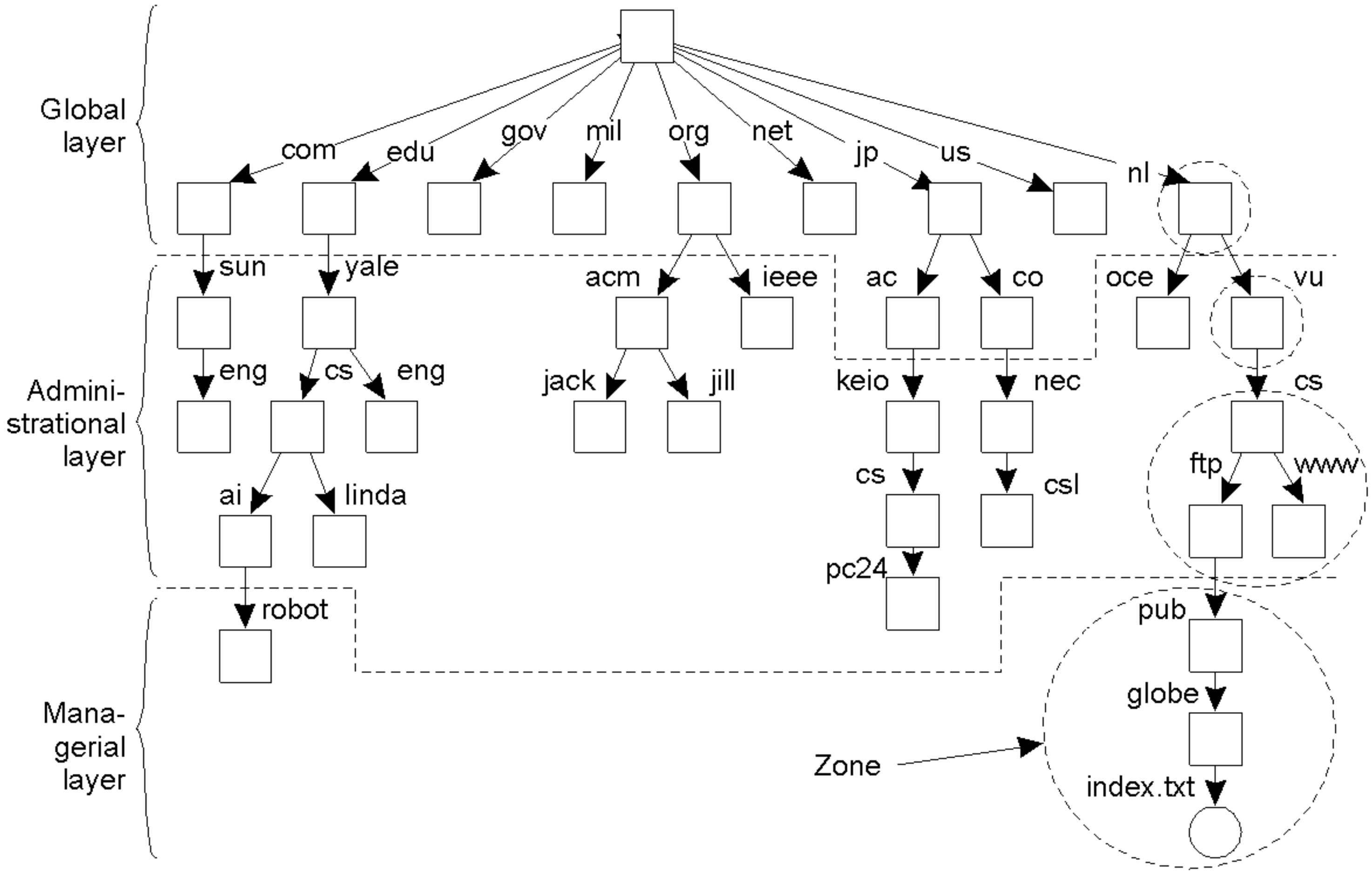
Namespaces

- Can be flat (e.g. a number, a string)
- Or structured (i.e. hierarchic) e.g. a Unix file name
- If hierarchic, each part of the name is resolved in a different context

DNS

- Domain Name System – names computers across the Internet
- Uses replication and caching
- Strict cache consistency not vital
- Very large amount of data – partitioned by domain
- In general need to involve >1 name server in full name resolution

Name Space Distribution (1)



TVS: Fig. 5-13. An example partitioning of the DNS name space, including Internet-accessible files, into three layers.

Name Space Distribution (2)

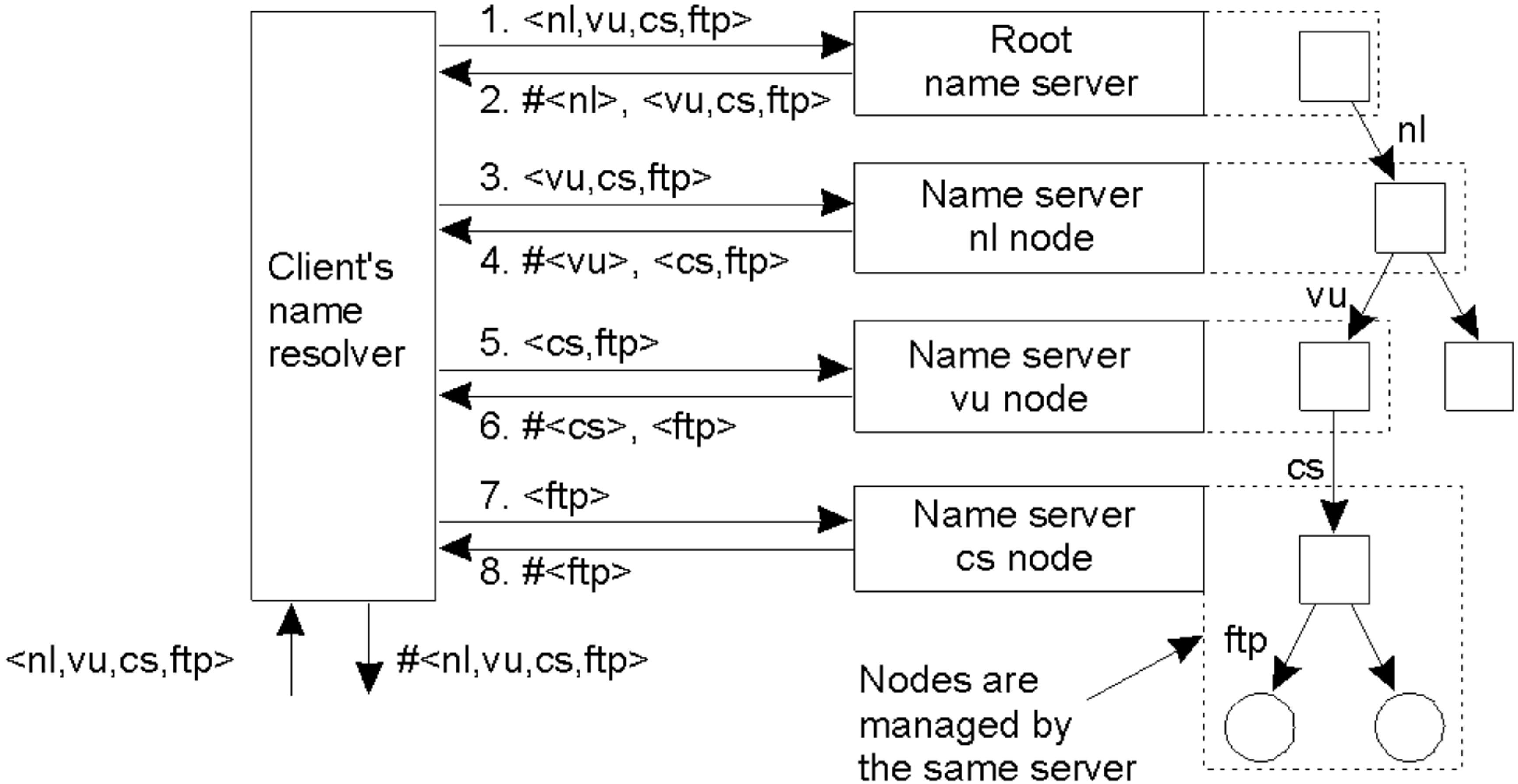
Item	Global	Administrational	Managerial
Geographical scale of network	Worldwide	Organization	Department
Total number of nodes	Few	Many	Vast numbers
Responsiveness to lookups	Seconds	Milliseconds	Immediate
Update propagation	Lazy	Immediate	Immediate
Number of replicas	Many	None or few	None
Is client-side caching applied?	Yes	Yes	Sometimes

TVS: Fig. 5-14. A comparison between name servers for implementing nodes from a large-scale name space partitioned into a global layer, as an administrational layer, and a managerial layer.

Name Resolution

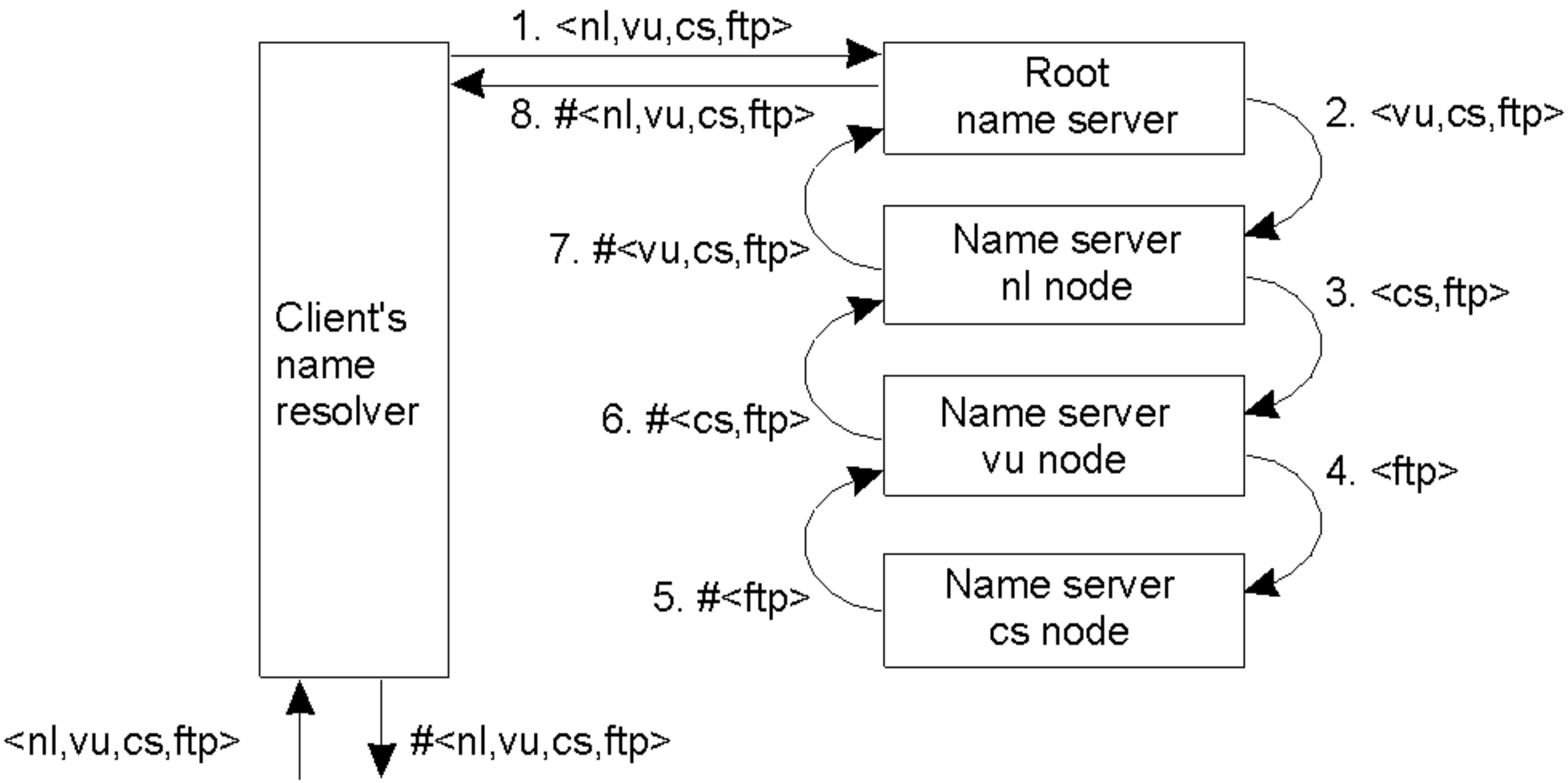
- Each client has a local name resolver
- It can work
  - Iteratively
  - Recursively
- Use #<xx> to mean the address of the name server for handling names in the node <xx>

Implementation of Name Resolution (1)



TVS: Fig 5-15 Iterative Name Resolution

Implementation of Name Resolution (2)



TVS: Fig 5-16: recursive name resolution.

Iterative vs Recursive Resolution

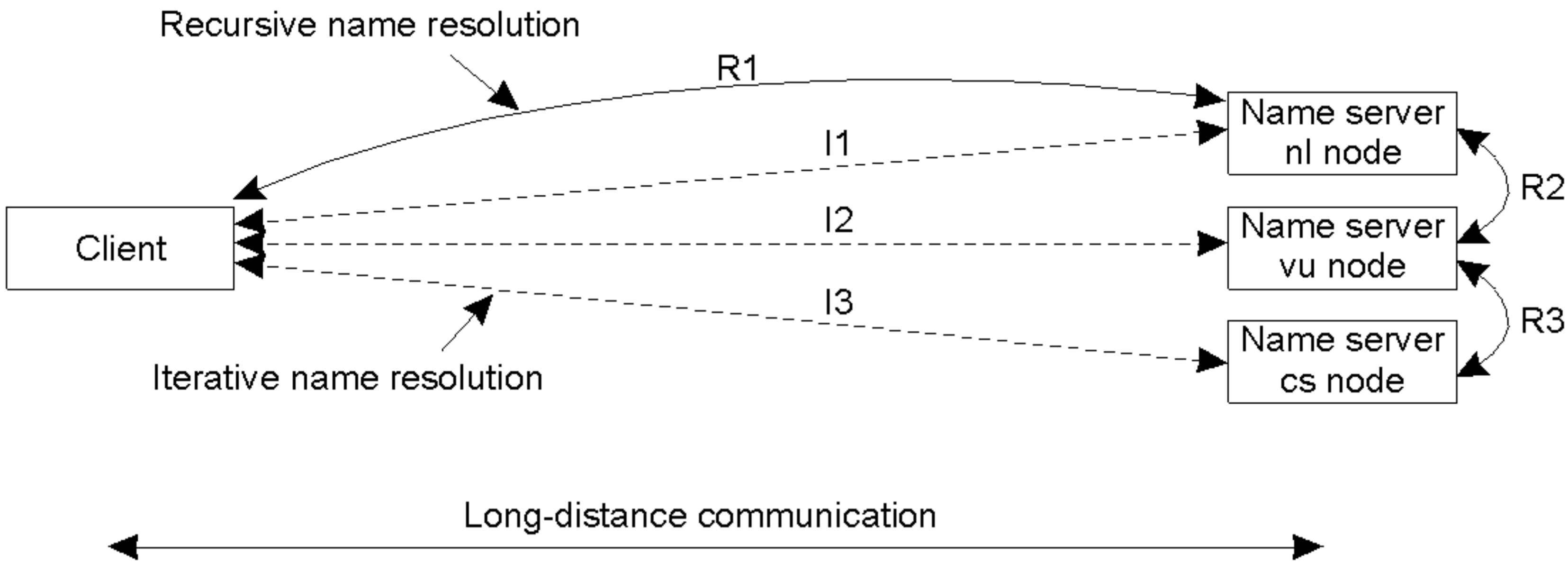
- Recursive resolution puts more burden on a name server – global layers support only iterative resolution
- Recursive resolution makes caching more effective ….
- Communication costs may be lower for recursive resolution

Implementation of Name Resolution (3)

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>
nl	<vu,cs,ftp>	#<vu>	<cs,ftp>	#<cs> #<cs,ftp>	#<vu> #<vu,cs> #<vu,cs,ftp>
root	<nl,vu,cs,ftp>	#<nl>	<vu,cs,ftp>	#<vu> #<vu,cs> #<vu,cs,ftp>	#<nl> #<nl,vu> #<nl,vu,cs> #<nl,vu,cs,ftp>

TVS: Fig. 5-17 Recursive name resolution of <*nl, vu, cs, ftp*>.

Implementation of Name Resolution (4)



TVS: Fig. 5-18. The comparison between recursive and iterative name resolution with respect to communication costs.

Zones

- DNS data divided into zones
- Each contains attribute data for a domain, but not that held in a sub-domain
- Two authoritative name servers for zone
- Names of servers for sub-domains
- Zone management data (e.g. lifetime of cached items ….)

The DNS Name Space

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
PTR	Node	Symbolic link with the primary name of the represented node
CNAME	Host	Contains the canonical name of a host
HINFO	Host	Holds information on the host this node represents
TXT	Any kind	Contains any entity-specific information considered useful

TVS: Fig. 5-19 .The most important types of resource records forming the contents of nodes in the DNS name space.

DNS Implementation

TVS: Figure 5-20. An excerpt from the DNS database for the zone *cs.vu.nl*.

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

Name Server vs Directory Server

- A name server takes a name, and returns one or more attributes of the named object
- A directory server takes attribute values, and returns sets of attributes of objects with those attribute values
- Like telephone directory: white pages vs yellow pages distinction

X.500 Directory Service

- X.500 invented by standards organisations
- Collection of all entries = Directory Information Base (DIB) – portions on different servers (Directory Service Agents, DSAs).
- Clients are Directory User Agents (DUAs)
- Get a Directory Information Tree (DIT)

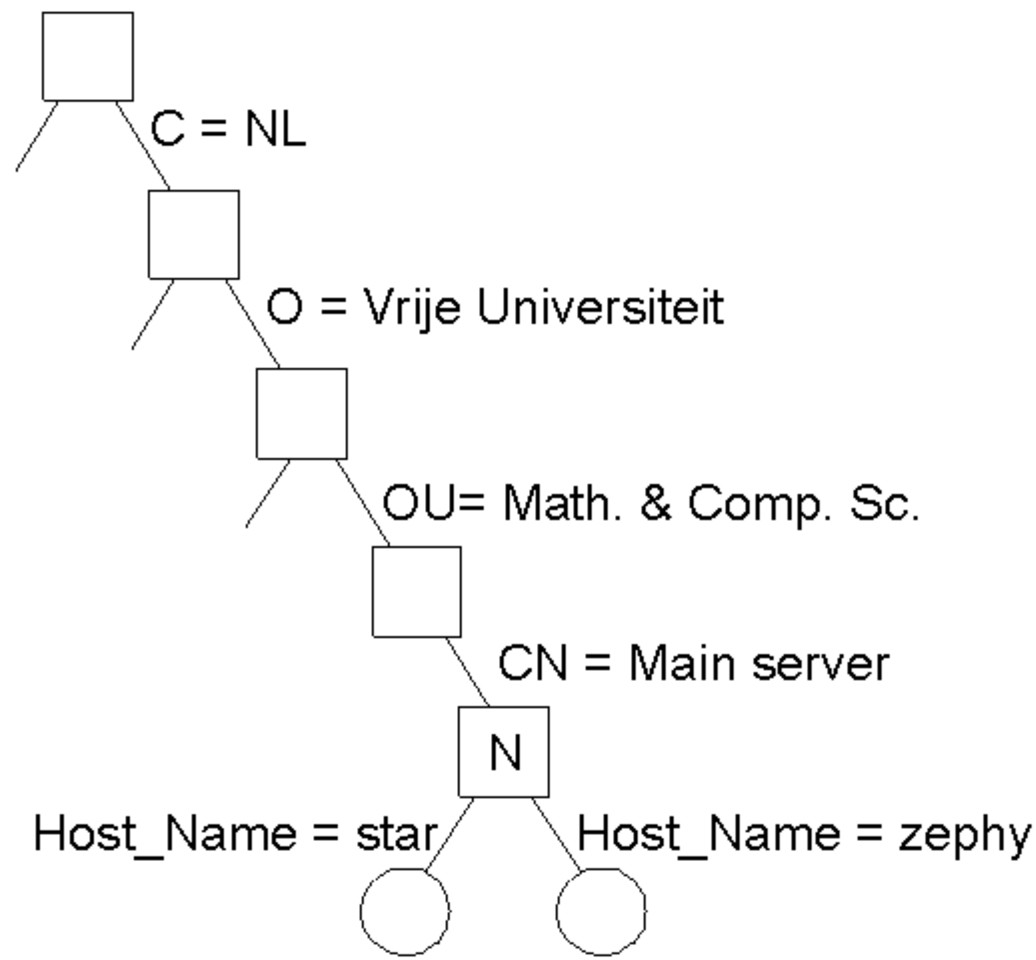
LDAP

- Lightweight Directory Access Protocol - a simple protocol for use with X.500
- Allows more simple directory lookup than X.500 for directory servers which implement
- Widely adopted (e.g. Microsoft's Active Directory Service provides an LDAP interface)

Attribute	Abbr.	Value
Country	C	NL
Locality	L	Amsterdam
Organization	O	Vrije Universiteit
OrganizationalUnit	OU	Comp. Sc.
CommonName	CN	Main server
Mail_Servers	--	130.37.24.6, 192.31.231.42,192.31.231.66
FTP_Server	--	130.37.21.11
WWW_Server	--	130.37.21.11

TVS: Fig. 5-22. A simple example of an LDAP directory entry using LDAP naming conventions.

TVS: Fig. 5-23a Part of the directory information tree.



TVS: Fig. 5-23b. Two directory Entries

Attribute	**Value**
Country	NL
Locality	Amsterdam
Organization	Vrije Universiteit
OrganizationalUnit	Math. & Comp. Sc.
CommonName	Main server
Host_Name	star
Host_Address	192.31.231.42

Attribute	**Value**
Country	NL
Locality	Amsterdam
Organization	Vrije Universiteit
OrganizationalUnit	Math. & Comp. Sc.
CommonName	Main server
Host_Name	zephyr
Host_Address	192.31.231.66

LDAP access

- Entries can be read – enough info has to be provided to navigate the Directory Information Tree (DIT), but other attributes can then be obtained
- Can search, starting from a node in the DIT and using a boolean filter expression to identify targets. This can be quite costly!

- **<u>Next Lecture: Time and Logical Clocks</u>**