Name Services

CAT3053/N Distributed Computing

Week 10

- Which one is easy for humans and machines? and why?
- 74.125.237.83 or **google.com**
- 128.250.1.22 or distributed systems website
- 128.250.1.25 or Prof. Buyya
- Disk 4, Sector 2, block 5 OR /usr/raj/hello.c

Introduction

- In a distributed system, names are used to refer to a wide variety of resources such as: Computers, services, remote objects, and files, as well as users.
- Naming is fundamental issue in DS design as it facilitates communication and resource sharing.
  - A name in the form of URL is needed to access a specific web page.
  - Processes cannot share particular resources managed by a computer system unless they can name them consistently
  - Users cannot communicate within one another via a DS unless they can name one another, with email address.
- Names are not the only useful means of identification: descriptive attributes are another.

What are Naming Services?

- How do Naming Services facilitate communication and resource sharing?
  - An URL facilitates the localization of a resource exposed on the Web. *e.g., abc.net.au means it is likely to be an Australian entity?*
  - A consistent and uniform naming helps processes in a distributed system to interoperate and manage resources. *e.g., commercials use .com; non-profit organizations use .org*
  - Users refers to each other by means of their names (i.e. email) rather than their system ids
  - Naming Services are not only useful to locate resources but also to gather additional information about them such as attributes

What are Naming Services?

- In a Distributed System, a Naming Service is a specific service whose aim is to provide a consistent and uniform naming of resources, thus allowing other programs or services to localize them and obtain the required metadata for interacting with them.
- **Key benefits**
  - Resource localization
  - Uniform naming
  - Device independent address (e.g., you can move domain name/web site from one server to another server seamlessly).

Role of names and name services

- Resources are accessed using *identifier or reference*
  - An identifier can be stored in variables and retrieved from tables quickly
  - Identifier includes or can be transformed to an address for an object *E.g. NFS file handle, CORBA remote object reference*
  - 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.cdk3.net/*

Role of names and name services

- 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

Requirement for name spaces

- 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 name trees
  - for some types of change, old programs should continue to work
- Management of trust

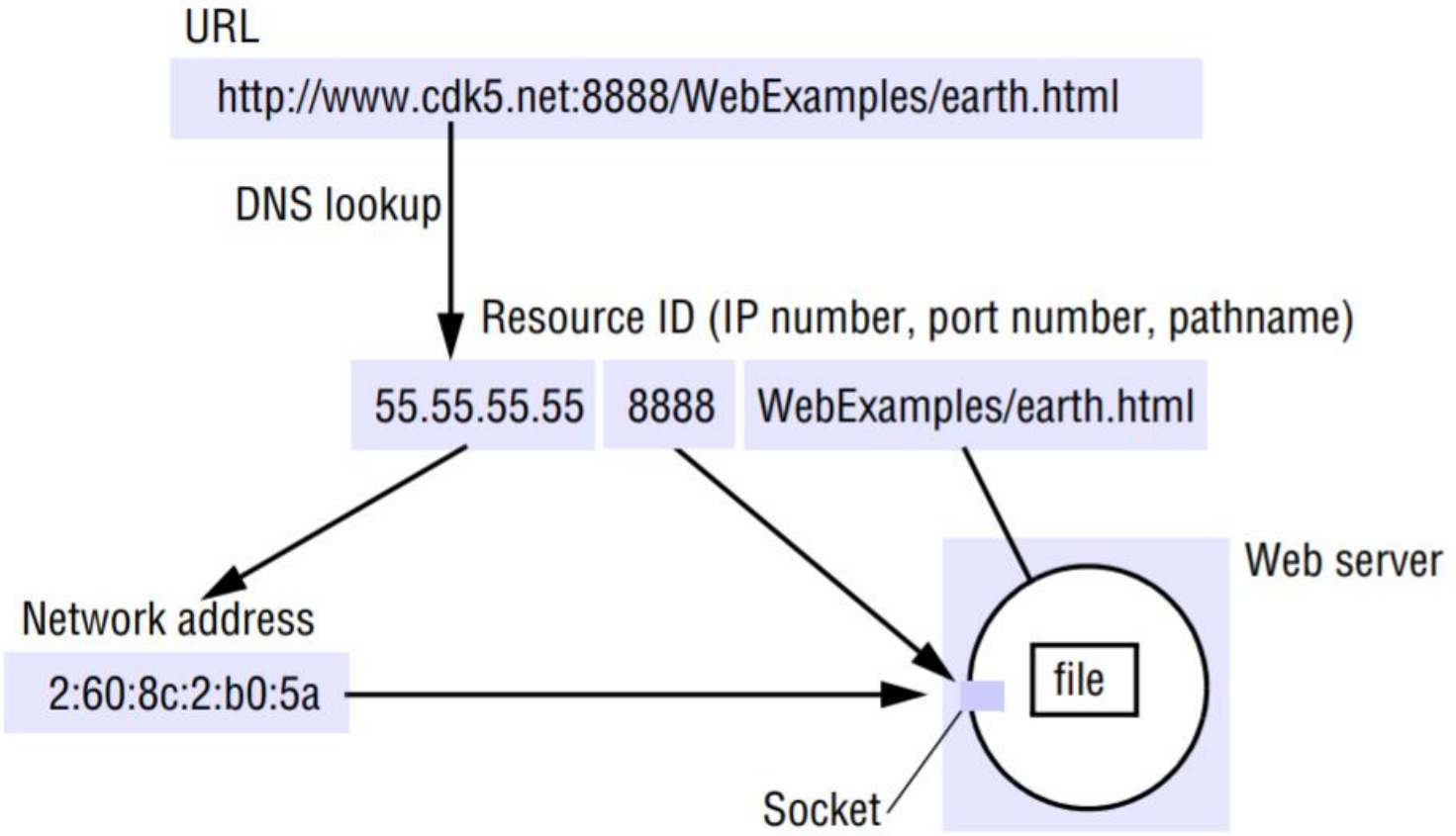
Names, addresses and other attributes

- A key attribute of an entity that is usually relevant in a distributed system is its address.
  - The DNS maps domain names to the attributes of a host computer: its IP address, the type of entry (for example, a reference to a mail server or another host) and, for example, the length of time the host's entry will remain valid.

Names, addresses and other attributes

- The X500 directory service can be used to map a person's name onto attributes including their email address and telephone number.
- The CORBA Naming Service maps the name of a remote object onto its remote object reference.
- Trading Service maps the name of a remote object onto its remote object reference, together with an arbitrary number of attributes describing the object in terms understandable by human users.

Composed naming domains used to access a resource from a URL



Name Services and the Domain Name System

- A name service stores a collection of one or more naming contexts, sets of bindings between textual names and attributes for objects such as computers, services, and users.
- The major operation that a name service supports is to resolve names.

Uniform Resource Identifiers

- Uniform Resource Identifiers (URIs) came about from the need to identify resources on the Web, and other Internet resources such as electronic mailboxes.
- An important goal was to identify resources in a coherent way, so that they could all be processed by common software such as browsers.
- URIs are ‘uniform’ in that their syntax incorporates that of indefinitely many individual types of resource identifiers (that is, URI schemes), and there are procedures for managing the global namespace of schemes.
- The advantage of uniformity is that it eases the process of introducing new types of identifier, as well as using existing types of identifier in new contexts, without disrupting existing usage.

Uniform Resource Locators

- Some URIs contain information that can be used to locate and access a resource; others are pure resource names.
- The familiar term Uniform Resource Locator (URL) is often used for URIs that provide location information and specify the method for accessing the resource.

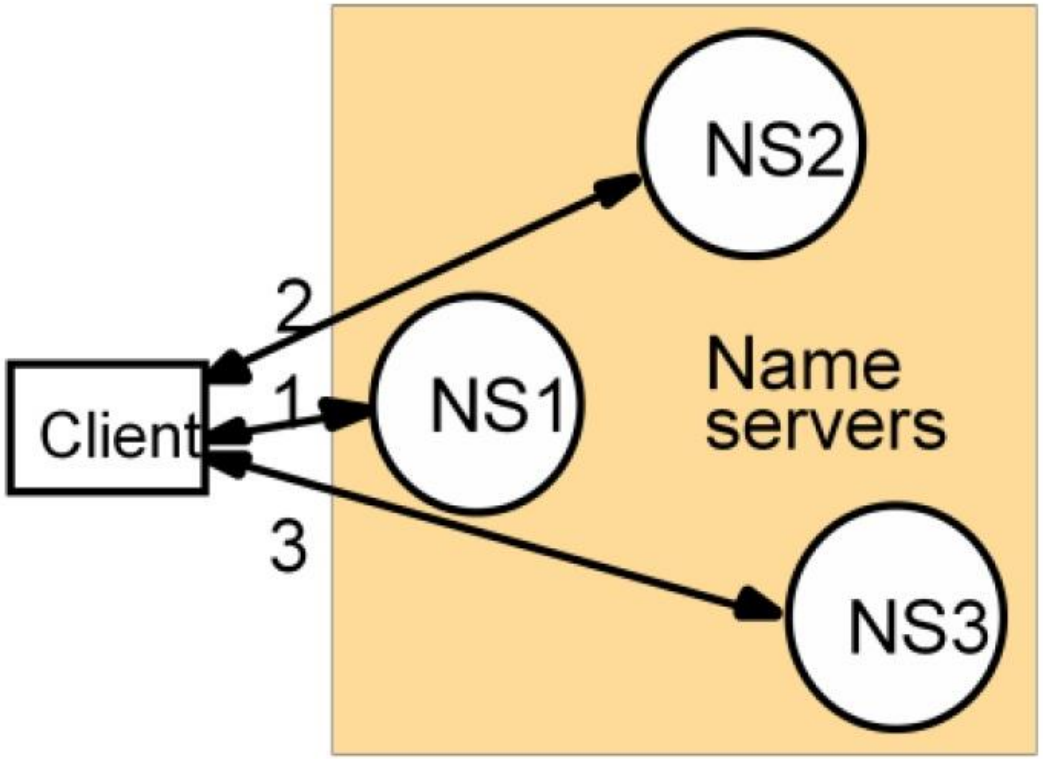
Uniform Resource Names

- Uniform Resource Names (URNs) are URIs that are used as pure resource names rather than locators. For example, the URI:
  - mid:0E4FC272-5C02-11D9-B115-000A95B55BC8@hpl.hp.com

Navigation

- Navigation is the act of chaining multiple Naming Services in order to resolve a single name to the corresponding resource.
- Namespaces allows for structure in names.
- URLs provide a default structure that decompose the location of a resource in
  - protocol used for retrieval
  - internet end point of the service exposing the resource
  - service specific path
- This decomposition facilitates the resolution of the name into the corresponding resource
- Moreover, structured namespaces allows for iterative navigation.

Iterative navigation



Client iteratively contacts name servers NS1 – NS3 in order to resolve a name.

Iterative navigation

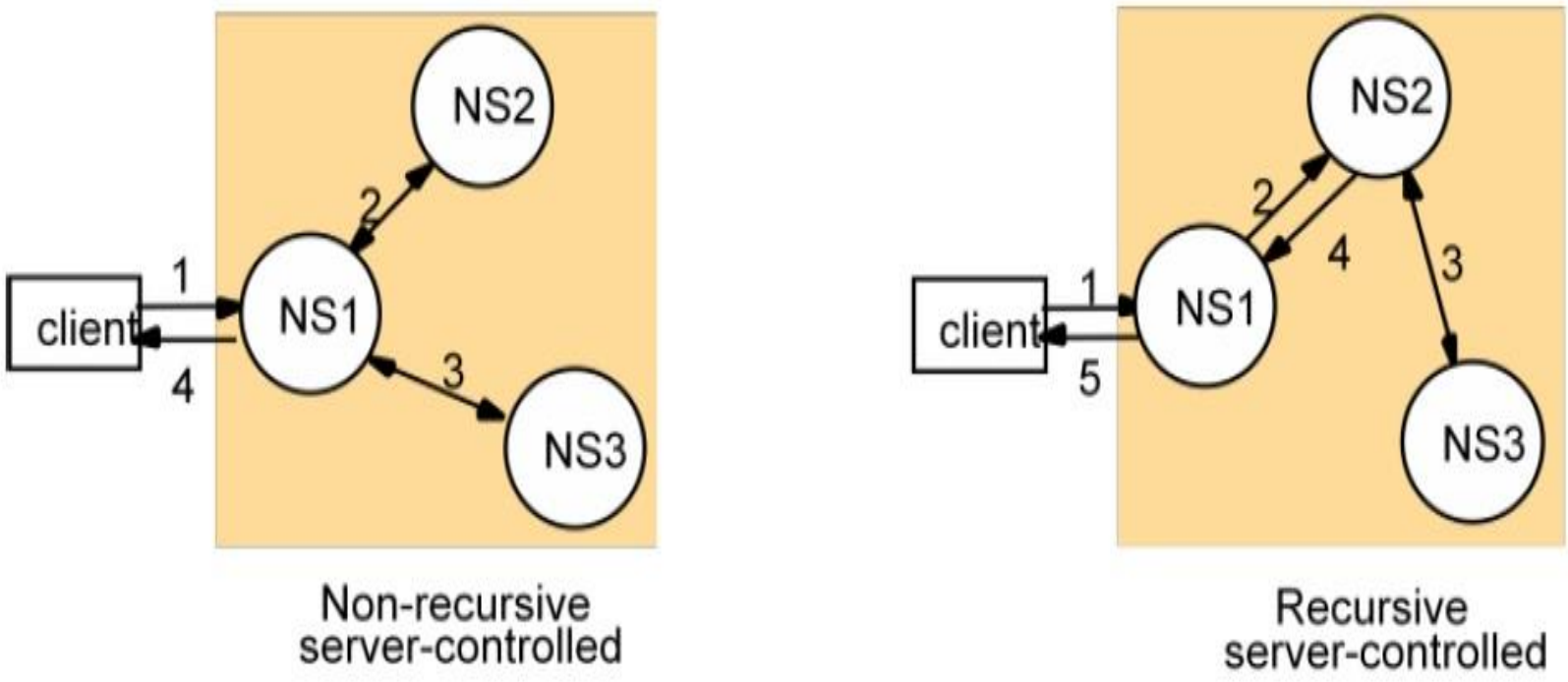
Reason for NFS iterative name resolution

- This is because the file service may encounter a symbolic link (i.e. an *alias) when resolving a* name.
- A symbolic link must be interpreted in the client's file system name space because it may point to a file in a directory stored at another server.
- The client computer must determine which server this is, because only the client knows its mount points.

Server Controlled Navigation

- In an alternative model, name server coordinates naming resolution and returns the results to the client. It can be:
- – Recursive:
  - *it is performed by the naming server*
  - *the server becomes like a client for the next server*
  - *this is necessary in case of client connectivity constraints*
- – Non recursive:
  - *it is performed by the client or the first server*
  - *the server bounces back the next hop to its client*

Non-recursive and recursive server-controlled navigation



A name server NS1 communicates with other name servers on behalf of a client

DNS - The Domain Name System

- The Domain Name System is a name service design whose main naming database is used across the Internet.
- The original Internet name scheme was soon seen to suffer from three major shortcomings:
  - It did not scale to large numbers of computers.
  - Local organizations wished to administer their own naming systems.
  - A general name service was needed – not one that serves only for looking up computer addresses.

DNS – The Domain Name System

- A distributed naming database (specified in RFC 1034/1305)
- Name structure reflects administrative structure of the Internet
- Rapidly resolves domain names to IP addresses
  - exploits caching heavily
  - typical query time ~100 milliseconds
- Scales to millions of computers
  - partitioned database
  - caching
- Resilient to failure of a server
  - Replication

Domain Names

- The DNS is designed for use in multiple implementations, each of which may have its own name space.
- In practice, however, only one is in widespread use, and that is the one used for naming across the Internet.
- The Internet DNS name space is partitioned both organizationally and according to geography.
- The names are written with the highest-level domain on the right.
- The original top-level organizational domains (also called *generic domains)* in use across the Internet were:

Domain Names (Cont.)

- *com – Commercial organizations*
- *edu – Universities and other educational institutions*
- *gov – US governmental agencies*
- *mil – US military organizations*
- *net – Major network support centres*
- *org – Organizations not mentioned above*
- *int – International organizations*

Domain Names (Cont.)

- New top-level domains such as *biz and mobi have been added since the early 2000s.*
- *A full list of* current generic domain names is available from the Internet Assigned Numbers Authority [www.iana.org I].
- In addition, every country has its own domains:

Domain Names (Cont.)

- *us – United States*
- *uk – United Kingdom*
- *fr – France*
- *Etc…*

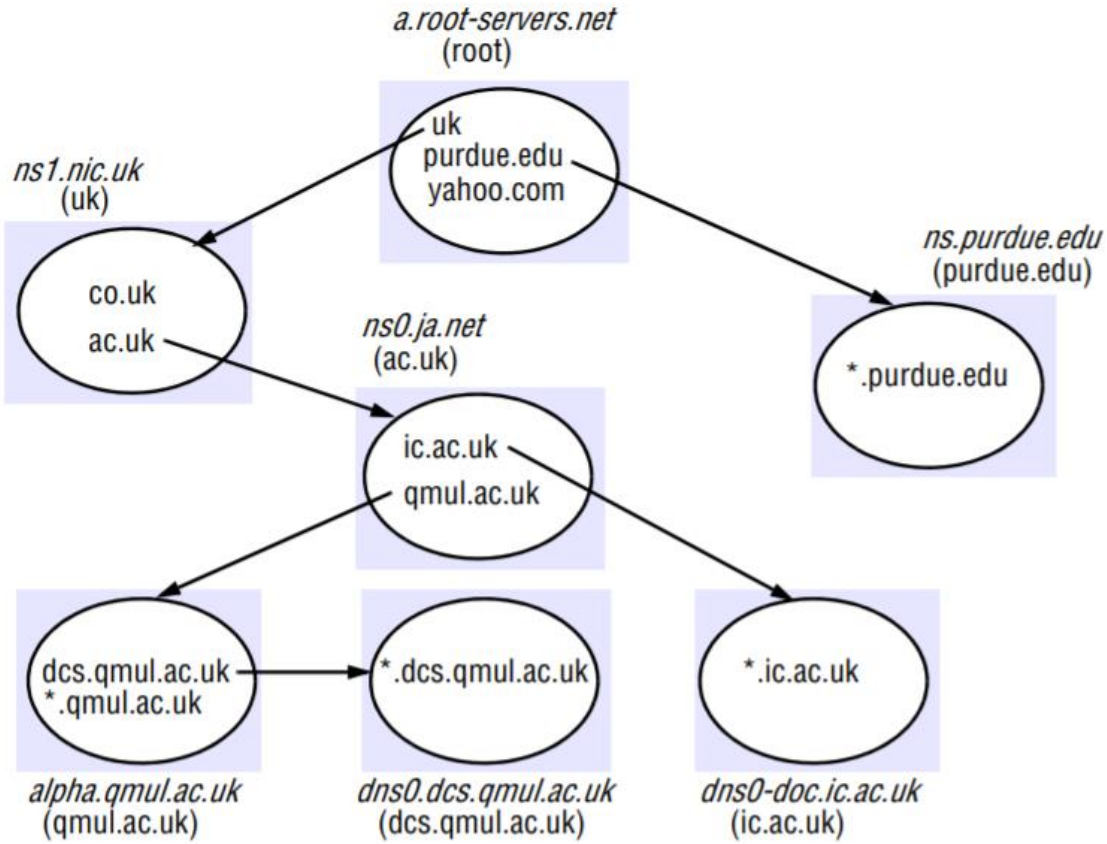
DNS Queries

- The Internet DNS is primarily used for simple host name resolution and for looking up electronic mail hosts:
  - *Host name resolution*
  - *Mail host location*
  - *Reverse resolution*
  - *Host information*

Basic DNS algorithm for name resolution (domain name -> IP number)

- Look for the name in the local cache
- Try a superior DNS server, which responds with:
  - another recommended DNS server
  - the IP address (which may not be entirely up to date)

DNS name servers: Hierarchical organisation



Name server names are in italics, and the corresponding domains are in parentheses. Arrows denote name server entries

Navigation and query processing

- The DNS architecture allows for recursive navigation as well as iterative navigation.
- The resolver specifies which type of navigation is required when contacting a name server.
- However, name servers are not bound to implement recursive navigation.
- As was pointed out above, recursive navigation may tie up server threads, meaning that other requests might be delayed.

DNS Resource records

Record type	*Meaning*	*Main contents*
A	A computer address (IPv4)	IPv4 number
AAAA	A computer address (IPv6)	IPv6 number
NS	An authoritative name server	Domain name for server
CNAME	The canonical name for an alias	Domain name for alias
SOA	Marks the start of data for a zone	Parameters governing the zone
PTR	Domain name pointer (reverse lookups)	Domain name
HINFO	Host information	Machine architecture and operating system
MX	Mail exchange	List of <*preference*, *host*> pairs
TXT	Text string	Arbitrary text

DNS Resource records

- The data for a zone starts with an *SOA-type record, which contains the zone parameters that* specify
  - the version number and how often secondary's should refresh their copies.
- This is followed by a list of records of type *NS specifying the name servers for the domain and a* list of records of type *MX giving the domain names of mail hosts, each prefixed by a number* expressing its preference.
  - part of the database for the domain *dcs.qmul.ac.uk at one* point is shown in the following figure where the time to live *1D means 1 day.*

DNS zone data records

domain name	*time to live*	*class*	*type*	*value*
dcs.qmul.ac.uk	*1D*	*IN*	*NS*	*dns0*
dcs.qmul.ac.uk	*1D*	*IN*	*NS*	*dns1*
dcs.qmul.ac.uk	*1D*	*IN*	*MX*	*1 mail1.qmul.ac.uk*
dcs.qmul.ac.uk	*1D*	*IN*	*MX*	*2 mail2.qmul.ac.uk*

The majority of the remainder of the records in a lower-level zone like *dcs.qmul.ac.uk* will be of type *A* and map the domain name of a computer onto its IP address. They may contain some aliases for the well-known services, for example:

domain name	*time to live*	*class*	*type*	*value*
www	*1D*	*IN*	*CNAME*	traffic
traffic	*1D*	*IN*	*A*	138.37.95.150

If the domain has any subdomains, there will be further records of type *NS* specifying their name servers, which will also have individual *A* entries. For example, at one point the database for *qmul.ac.uk* contained the following records for the name servers in its subdomain

dcs.qmul.ac.uk:

domain name	*time to live*	*class*	*type*	*value*
dcs	*1D*	*IN*	*NS*	*dns0.dcs*
dns0.dcs	*1D*	*IN*	*A*	*138.37.88.249*
dcs	*1D*	*IN*	*NS*	*dns1.dcs*
dns1.dcs	*1D*	*IN*	*A*	*138.37.94.248*

DNS issues

- Name tables change infrequently, but when they do, caching can result in the delivery of stale data.
  - Clients are responsible for detecting this and recovering
- Its design makes changes to the structure of the name space difficult. For example:
  - merging previously separate domain trees under a new root
  - moving subtrees to a different part of the structure (e.g. if Scotland became a separate country, its domains should all be moved to a new country-level domain.)

Directory Service

- The name services store collections of <*name, attribute> pairs, and the attributes are* looked up from a name.
- It is natural to consider the dual of this arrangement, in which *attributes* are used as values to be looked up.
- In these services, textual names can be considered to be just another attribute.

Directory Service (Cont.)

- Sometime users wish to find a particular person or resource, but they don't know its name, only some of its attributes.
  - What is the name of the user with a telephone number 03-83441344?
  - What is the name of professor teaching Cloud computing at UniMelb (e.g., ask Google!)
- Sometime users require a service, but they are not concerned with what system entity provides it.
  - Which computers in this building are Macintoshes running the Mac OS X operating system?
  - Where can I print high resolution colour image?

Directory Services (Cont.)

- A service that stores collections of bindings between names and attributes and that looks up entries that match attribute-based specifications is called a *directory service.*
- Examples are Microsoft's Active Directory Services, X.500 and its cousin LDAP, Univers and Profile.

Directory Service (Cont.)

- Directory service: 'yellow pages' for the resources in a network
- Retrieves the set of names that satisfy a given description
- *DNS holds some descriptive data, but:*
  - the data is very incomplete
  - DNS isn't organised to search it

Directory Service (Cont.)

- A directory service returns the sets of attributes of any objects found to match some specified attributes.
  - the request ‘TelephoneNumber = 020 555 9980’ might return {‘Name = John Smith’, ‘TelephoneNumber = 020 555 9980’, ‘emailAddress = john@dcs.gormenghast.ac.uk’, ...}
- The client may specify that only a subset of the attributes is of interest – for example, just the email addresses of matching objects.
- X.500 and some other directory services also allow objects to be looked up by conventional hierarchic textual names.

Discovery Service

- a special case of a directory service
- 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, the 'service location* protocol', the 'simple service discovery protocol' (part of UPnP), the 'secure discovery service'.

Discovery Service (Cont.)

- A database of services with lookup based on service description or type, location and other criteria, E.g.
  1. Find a printing service in this hotel compatible with a Nikon camera
  2. Send the video from my camera to the digital TV in my room.
- Automatic registration of new services
- Automatic connection of guest's clients to the discovery service