Chapter 13: Name Services

Learning objectives

- To understand the need for naming systems in distributed systems
- To be familiar with the design requirements for distributed name services
- To understand the operation of the Internet naming service - DNS
- To be familiar with the role of directory and discovery services

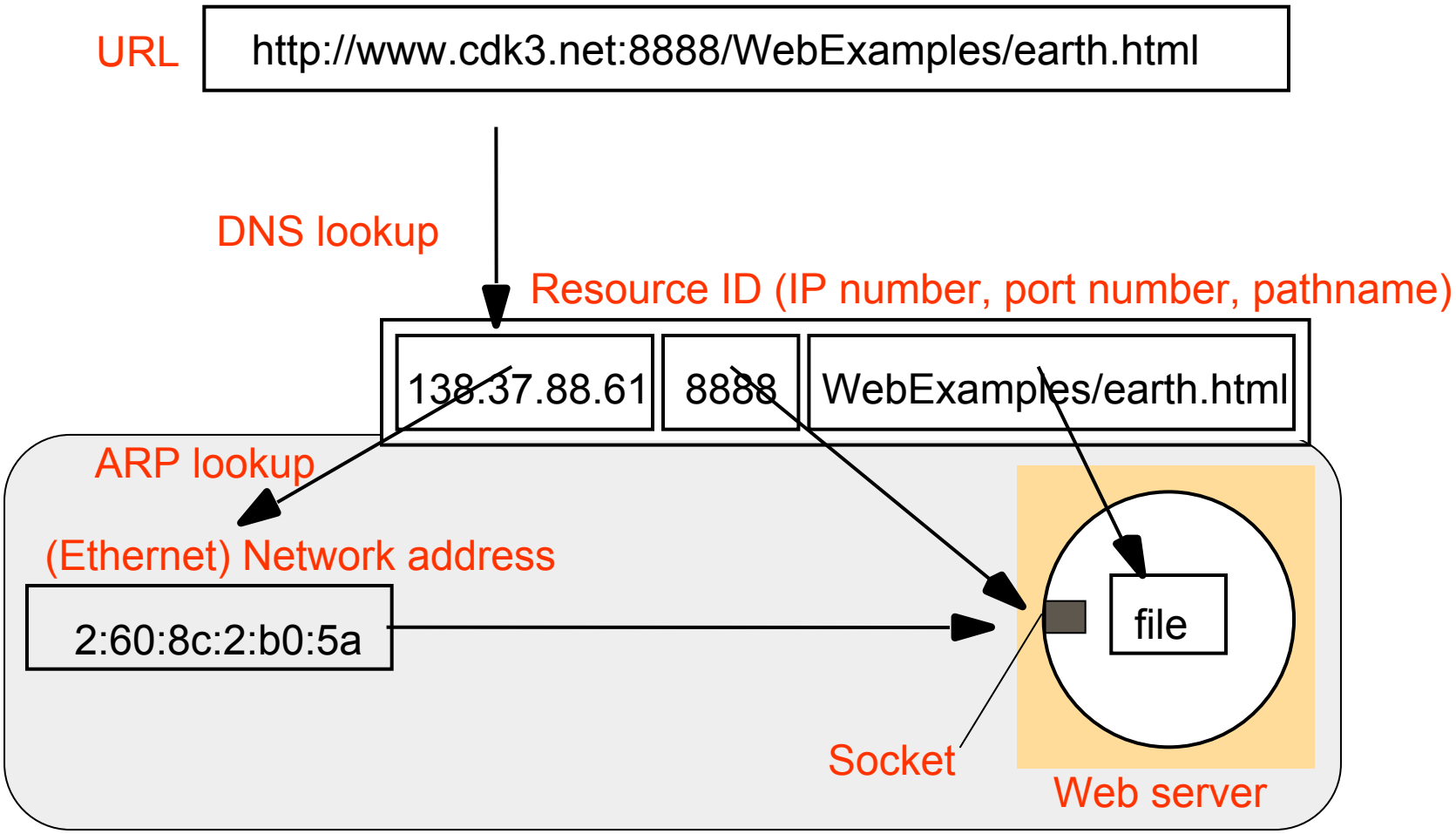
The role of names

- 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/*
- 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

Resolving and binding of names

- A name is *resolved* when it is translated into data about the named resource or object
  - often in order to invoke an action upon it
- The association between a name and an object is called a *binding*
  - In general, names are bound to attributes of the named objects
  - An attribute is the value of a property associated with an object
  - E.g., DNS maps domain names to the attributes of a host computer: its IP address

Composed naming domains used to access a resource from a URL



- The domain name portion of a URL resolved first via the DNS into an IP address and then via ARP to an Ethernet address
- The last part of the URL is resolved by the file system on the web server to locate the relevant file

Names and resources

Currently, different name systems are used for each type of resource:

resource	*name*	*identifies*
file	pathname	file within a given file system
process	process id	process on a given computer
port	port number	IP port on a given computer

Uniform Resource Identifiers (URI) offer a general solution for any type of resource. There two main classes:

URL *Uniform Resource Locator*

- typed by the protocol field (http, ftp, nfs, etc.)
- part of the name is service-specific
- resources cannot be moved between domains

URN *Uniform Resource Name*

- requires a universal resource name lookup service - a DNS-like system for all resources

More on URNs

format: urn:<nameSpace>:<name-within-namespace>

examples:

a) urn:ISBN:021-61918-0

b) urn:dcs.qmul.ac.uk:TR2000-56

resolution:

a) send a request to nearest ISBN-lookup service - it would return whatever attributes of a book are required by the requester

b) send a request to the urn lookup service at dcs.qmul.ac.uk - it would return a url for the relevant document

Name services

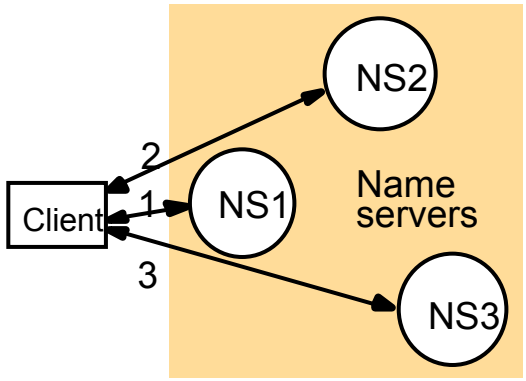
- A ***name service*** stores a collection of one or more naming contexts – sets of bindings between textual names and attributes for objects such as users, computers, services and remote objects.
- The major operation is to *resolve* a name
  - that is, to *look up* attributes from a given name
- General requirements for name services
  - To handle an essentially arbitrary number of names and to serve an arbitrary number of administrative organizations
  - High availability

Name spaces

- A name space is the collection of all valid names recognized by a particular service
- Allow simple but meaningful names to be used
- Potentially infinite number of names
- Structured
  - to allow same subnames without clashes
  - to group related names
- Aliases: one domain name is defined to stand for another
  - http://espn.go.com/ and http://www.espn.com
- Naming domains: is a name space for which there exists a single overall administrative authority for assigning names within it
  - Domains in DNS are collections of domain names; syntactically, a domain's name is the common suffix of the domain names within it
  - The administration of domains may be devolved to sub-domains

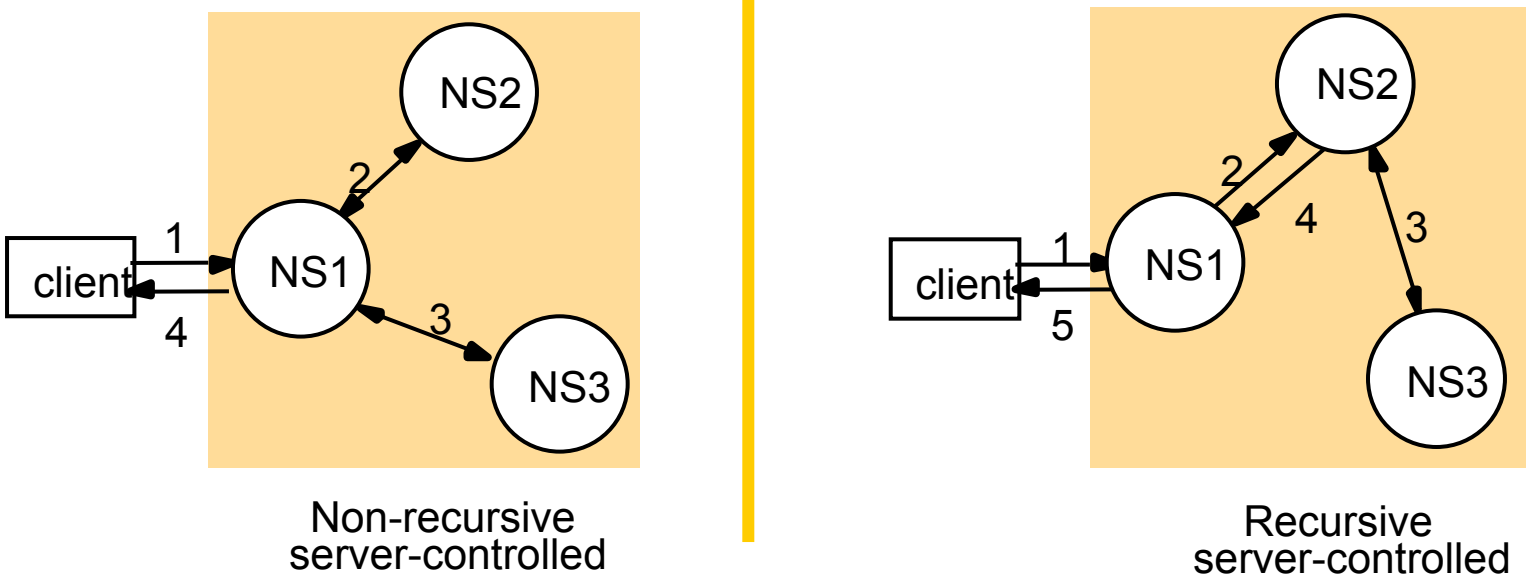
Name resolution: iterative navigation

A client iteratively contacts name servers NS1–NS3 in order to resolve a name



- The partitioning of data implies that the local name server cannot answer all enquiries without the help of other name servers
- The process of locating naming data from among more than one name server in order to resolve a name is called ***navigation***
- DNS supports the model known as ***iterative navigation***: to resolve a name, a client presents entire name to servers, starting at a local server, NS1. If NS1 has the requested name, it is resolved, else NS1 suggests contacting NS2 (a server for a domain that includes the requested name)

Non-recursive and recursive server-controlled navigation



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

- An alternative model that DNS supports: a name server coordinates the resolution of the name and passes the result back to the user agent
  - Non-recursive server-controlled: any name server may be chosen by the client.
  - Recursive server-controlled: the client contacts a single server
- Recursive navigation must be used in domains that limit client access to their DNS information for security reasons

DNS - The Internet Domain Name System

- DNS is a name service design whose main naming database is used across the Internet
  - Derived in 1987 to replace the original Internet naming scheme, in which all host names and addresses were held in a single central master file and downloaded by FTP to all computers that required them
- Major shortcomings of old scheme:
  - did not scale
  - 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
- 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 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*):
  - com: commercial organizations
  - edu: universities and other educational institutions
  - gov: US governmental agencies
  - mil:US military organizations
  - net: major network support centers
  - org: organizations not mentioned above
  - int: international organizations
- New top-level domains were added in the early 2000s
- In addition, every country has its own domains:
  - us, uk, fr, ca, cn

DNS queries

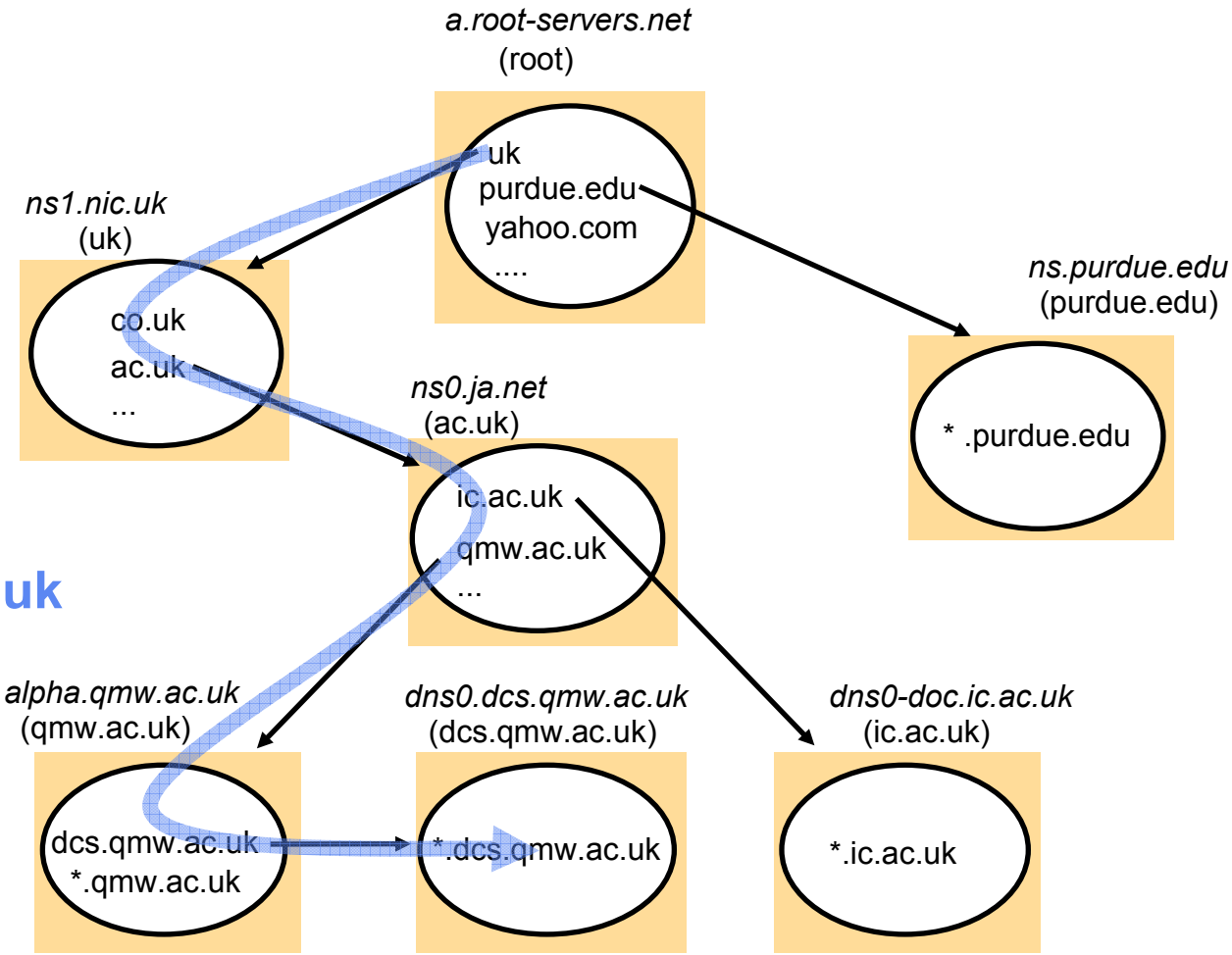
- Main function is to resolve domain names for computers, i.e. to get their IP addresses
  - caches the results of previous searches until they pass their 'time to live'
- Other functions:
  - get *mail host* for a domain
  - reverse resolution - get domain name from IP address
  - Host information -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

DNS name servers

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

authoritative path to lookup:

jeans-pc.dcs.qmw.ac.uk

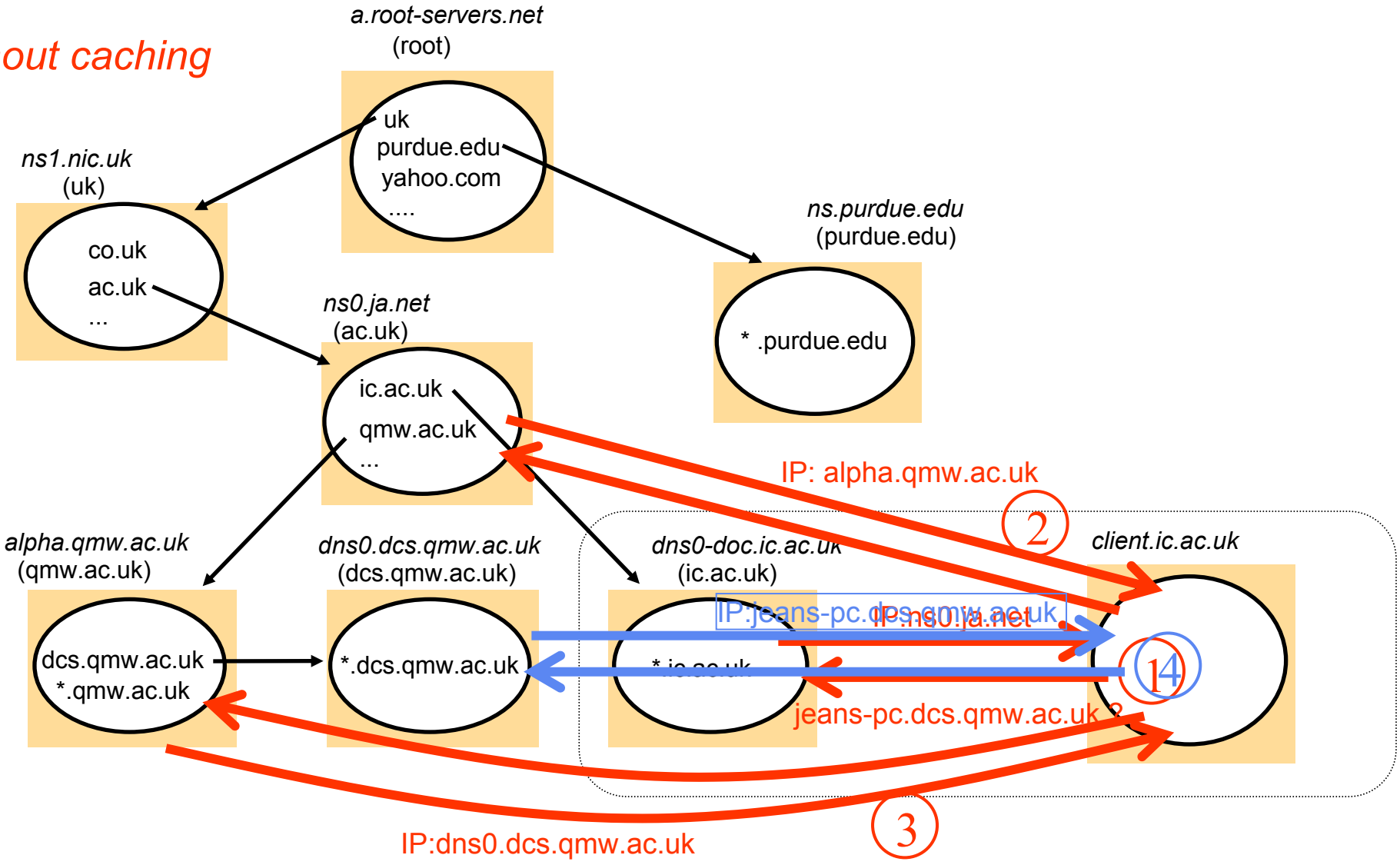


Basic DNS algorithm for name resolution

- **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 in typical operation

Without caching



DNS resource records

Record type	*Meaning*	*Main contents*
A	A computer address	IP 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
WKS	A well-known service description	List of service names and protocols
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

Directory and discovery services

- 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)
- UDDI (universal directory and discovery service) provides both white pages and yellow pages services, slide 18-19 of chapter 19
  - Provide information about organizations and the web services they provide
- Discovery service: a special case of a directory service for services provided by devices in a spontaneous networking environment
  - automatically updated as the network configuration changes
  - 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

Summary

Name services:

- defer the binding of resource names to addresses (and other attributes)
- Names are resolved to give addresses and other attributes
- Goals :
  - Scalability (size of database, access traffic (hits/second), update traffic)
  - Reliability
  - Trust management (authority of servers)
- Issues
  - exploitation of replication and caching to achieve scalability without compromising the distribution of updates
  - navigation methods

Directory and discovery services:

- 'yellow pages' retrieval by attributes
- dynamic resource registration and discovery