Business and Logistics

☐ Problem set 2 due
  ☑ Answers to be posted tonight

☐ Exam study guide posted on course webpage

☐ Paper presentation instructions and schedule sent in email message
Acknowledgements

- Some material taken from author’s teaching slides
- Some figures taken from Distributed Systems: Concepts and Design book
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 discovery services in mobile and ubiquitous computer systems
The Role of Names

- Names in distributed systems can refer to a variety of resources as well as users.
- Names facilitate communication and resource sharing.
- Must be able to name things consistently.
- Names are not the only useful means of identification:
  - Descriptive attributes are another.
  - Use some information to describe.
- Name services provide clients with data about named objects.
- Directory and discovery services provide data about objects that satisfy a given description.
Names, Addresses, Attributes

- Any process must possess a name or identifier to access a specific resource
  - File names, URLs, Internet domain names
- *Identifier* is sometimes used to refer to names interpreted by programs
  - Remote object references, NSF file handles
- *Pure names* are uninterpreted bit patterns
  - Need to be looked up
- *Addresses* are inadequate for identification
Names, Addresses, Attributes (continued)

- Resources are accessed using identifier or reference
  - Identifiers can be stored in variables
  - Includes or can be transformed to an address
  - Names are is human-readable value (usually a string) that can be resolved to an identifier or address

- 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
Names, Addresses, Attributes (continued)

- Association between a name and an object is called a binding.
- Names are bound to attributes of the named objects rather than the implementation of the objects themselves.
- An attribute is the value of a property associated with an object.
  - Key attribute is its address.
- Addresses may be considered names to be looked up to get others addresses.
  - IP address looked up to get network address.
Requirements 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
Composed Naming Domains for URL Resource Access

URL


DNS lookup

Resource ID (IP number, port number, pathname)

55.55.55.55  8888  WebExamples/earth.html

ARP lookup

Network address

2:60:8c:2:b0:5a

Web server

Socket

file
Names and Services

- Many names are specific to some service
- Clients use names as part of requesting a service
  - File name or process identifier
  - Use only in context
- Also use names to refer to entities in a distributed system beyond the scope of any single service
  - Users, computers, servers, …
Names and Resources

Different name systems used for each type of resource:

<table>
<thead>
<tr>
<th>Resource</th>
<th>Name</th>
<th>Identifies</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>pathname</td>
<td>file within a given file system</td>
</tr>
<tr>
<td>process</td>
<td>process id</td>
<td>process on a given computer</td>
</tr>
<tr>
<td>port</td>
<td>port number</td>
<td>IP port on a given computer</td>
</tr>
</tbody>
</table>

Uniform Resource Identifiers (URI) offer a general solution for any type of resource

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
URNs

- **Format**
  - `<urn:<nameSpace>:<name-within-namespace>`

- **Examples**
  - `<urn:ISBN:021-61918-0>`
  - `<urn:dcs.qmul.ac.uk:TR2000-56>`

- **Resolution**
  - Send a request to nearest ISBN lookup service
    - returns whatever attributes of a book required by requester
  - Send a request to the urn lookup service at `dcs.qmul.ac.uk`
    - returns a url for the relevant document
Name Services

- Stores a collection of one or more naming contexts
  - Set of bindings between textual names and attributes
- Major operation supported is name resolution
  - Look up attributes from a given name
- Other operations
  - Create new bindings, delete bindings, list bound names
  - Add and delete contexts
- Name management separated from other services
  - Unification: resources managed by different services use same naming scheme (e.g., URLs)
  - Integration: across administrative domains
General Name Service Requirements

- Name-mapping problem
  - Number of names
  - Number of administrative organizations
  - Lifetime
  - Availability
  - Fault isolation
  - Tolerance of mistrust

- Examples
  - Global Name Service (DEC)
  - Globe name service
  - Internet Domain Name System (DNS)
Name Spaces

- Collection of all valid names recognized by a particular service
  - A valid name is one that the service will attempt to look up
- Name spaces require a syntactic definition
- Names may have a flat structure
- Names may have internal structure representing their position in a hierarchic name space
  - Advantage is that each part of a name is resolved relative to a separate context
  - Same name may be used with different meanings in different contexts
Name Spaces (continued)

- Hierarchic names are potentially infinite and can grow
- Different contexts can be managed separately
- URLs have hierarchic structure and can use relative names
- DNS name space also has hierarchic structure
  - DNS names are called *domain names*
  - Consist of one or more strings called *name components* or *labels* separated by ‘.’
  - DNS names are case insensitive
  - DNS servers do not recognize relative names
- Aliases can be used for transparency and generality
Naming Domain

- A naming domain 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.
- Administration of domains may be devolved to sub-domains.
- Responsibility for a naming domain goes hand in hand with responsibility for managing and maintaining the corresponding part of the database stored in an authoritative name server and used by the name service.
- DNS provides a global and homogeneous name space.
Name Resolution

- Iterative process whereby a name is repeatedly presented to naming contexts
  - Either maps a given name onto a set of primitive attributes directly
  - Or maps it onto a further naming context and a derived name to be presented to that context
- Resolution iterates as long as further contexts and derived names are output
- Name services are provided by name servers
- Names services are replicated for high availability
- Name data is partitioned into servers based on domain
Navigation

- Locating naming data from among more than one name server in order to resolve a name
- DNS supports the model of iterative navigation

A client iteratively contacts name servers NS1–NS3 in order to resolve a name
Navigation (continued)

- DNS database is partitioned between servers in such a way as to allow many queries to be satisfied locally
- Others are satisfied without needing to resolve each part of the name separately
- NFS employs iterative navigation in resolution of file names because it may encounter symbolic links
Non-recursive / Recursive Server-Controlled

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

- Client name resolution software and servers maintain a cache of the results of previous name resolutions
- Client request for name lookup are first checked for in the name resolution software’s cache
  - If it holds a recent result, it returns it to the client
  - Otherwise, it sets about finding it from a server
- Server may cache data from other servers
- Caching is the key to performance
  - Also assists in maintaining availability
- Successful because names change infrequently
- Cached information though can get out of date
DNS - The Internet Domain Name System

- A distributed naming database
- 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
DNS Algorithm for Name Resolution

- Domain name → IP number

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

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

path to lookup: 
jeans-pc.dcs.qmw.ac.uk
DNS in Typical Operation

Without caching

1. IP: dns0.dcs.qmw.ac.uk
2. IP: ns0.ja.net
3. IP: dns0.dcs.qmw.ac.uk
DNS server functions and configuration

☐ 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)
# DNS resource records

<table>
<thead>
<tr>
<th>Record type</th>
<th>Meaning</th>
<th>Main contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A computer address</td>
<td>IP number</td>
</tr>
<tr>
<td>NS</td>
<td>An authoritative name server</td>
<td>Domain name for server</td>
</tr>
<tr>
<td>CNAME</td>
<td>The canonical name for an alias</td>
<td>Domain name for alias</td>
</tr>
<tr>
<td>SOA</td>
<td>Marks the start of data for a zone</td>
<td>Parameters governing the zone</td>
</tr>
<tr>
<td>WKS</td>
<td>A well-known service description</td>
<td>List of service names and protocols</td>
</tr>
<tr>
<td>PTR</td>
<td>Domain name pointer (reverse lookups)</td>
<td>Domain name</td>
</tr>
<tr>
<td>HINFO</td>
<td>Host information</td>
<td>Machine architecture and operating system</td>
</tr>
<tr>
<td>MX</td>
<td>Mail exchange</td>
<td>List of <code>&lt;preference, host&gt;</code> pairs</td>
</tr>
<tr>
<td>TXT</td>
<td>Text string</td>
<td>Arbitrary text</td>
</tr>
</tbody>
</table>
DNS Issues

- Name tables change infrequently, but when they do, caching can result in the delivery of stale data.
  - Clients are responsible for detecting and recovering
- Its design makes changes to the structure of the name space difficult
  - Merging previously separate domain trees
  - Moving subtrees to a different part of the structure
- These issues have been addressed in other systems
Directory and Discovery Services

- Directory service
  - “Yellow pages” for the resources in a network
  - Retrieves the set of names that satisfy a given description
  - X.500, LDAP, MS Active Directory Services

- Discovery service
  - 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: Jini discovery service, the “service location protocol,” “simple service discovery protocol,” “secure discovery service”
Service Discover in Jini

- Jini services register their interfaces and descriptions with the Jini lookup services in their scope
- Clients find the Jini lookup services in the scope
- Lookup service searches by attribute or interface type

1. ‘finance’ lookup service?
2. Here I am: ..... 
3. Request printing
4. Use printing service

Diagram:
- Client
- Printing service
- Corporate infoservice
- Network
- Lookup service
- admin
- finance
Topics Not Covered

- GNS case study (Section 9.4)
  - An early research project (1985) that developed solutions for the problems of:
    - large name spaces
    - restructuring the name space

- X.500 and LDAP (Section 9.5)
  - A hierarchically-structured standard directory service designed for world-wide use
  - Accommodates resource descriptions in a standard form and their retrieval for any resource (online or offline)
  - Never fully deployed, but the standard forms the basis for LDAP, the Lightweight Directory Access Protocol, which is widely used

- Trading services (see Section 17.3)
  - Directories of services with retrieval by attribute searching
  - Brokers negotiate the contract for the use of a service, including negotiation of attribute such as quality and quantity of service
Summary

- **Name services:**
  - Defer binding of resource names to addresses (and 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