Internet Directory Services

Overview

mapping: query ➠ responses

- ARP: IP $\rightarrow$ MAC
- DNS: host name, mail domain $\rightarrow$ IP address(es)
- server location protocol: local service $\rightarrow$ hosts
- X.500: general OSI directory service
- LDAP: directory access (X.500)
- whois: simple query, response
- finger: local, unstructured user query only
- whois++: with referrals
- URN

external interface; may use SQL, …internally

Directory Services

	content	linked
ARP	addresses	no
DNS	RR	yes
srvloc	hosts	no
LDAP	structured	access only
X.500	structured	yes
whois	text (domains)	no
whois++	structured	yes
finger	text	no

X.500 — “The Directory”

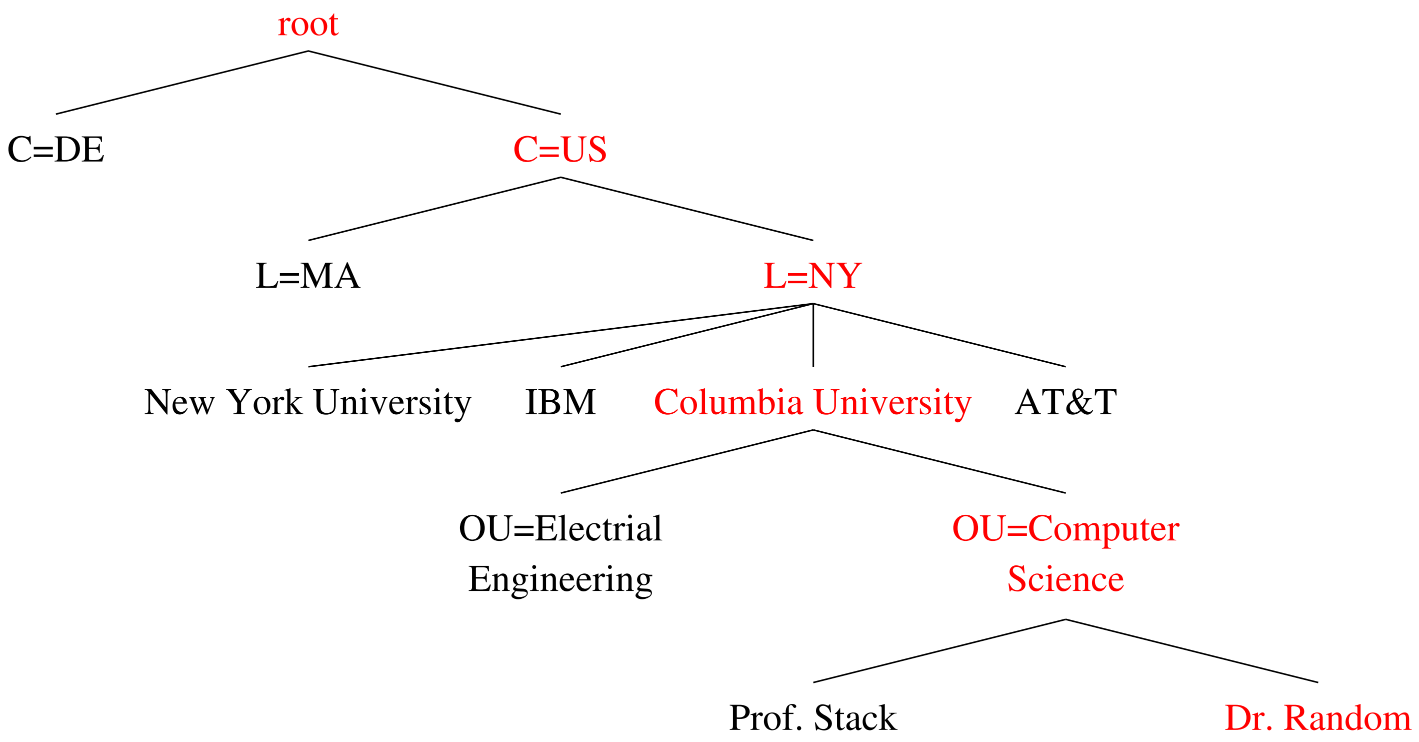
- distributed, hierarchical directory service
- “white pages”: DNS, whois
- “yellow pages”: YP, . . .
- needs full OSI protocol stack (transport, session, presentation)
- referrals

X.500

- top-down, all interconnected (sure!)
- decentralized: maintain local part
- structured information: attribute (OID)-values: common name, telephone number, PO box ➠ standard + lots of extensions
- one attribute = distinguished name (DN)
- hierarchical global name space
- *chaining:* wait, while asking around (whom?)
- *referral:* provide better DSA to querier ➠ HTTP redirection
- *multicasting:* ask several

X.500 distinguished names

- relative distinguished name (RDN) = unique identifier (key) at each level
- distinguished name (DN) = traversal of RDNs (called directory information tree (DIT))



X.500 naming

- which state does IBM belong in?
- no international registration

➠ Recent (11/26/96) proposal:

- use RFC 822 email address for RDN
- use DNS *domain components* to construct DN ➠ DN: `uid=foo@sales.ibm.com, dc=sales, dc=ibm, dc=com`

Lightweight Directory Access Protocol Overview

- client-server protocol
- access to X.500 *and others*
- entry: person, printer
- entry = attributes = type + value (text, JPEG, . . .)
- does not define data structure itself
- directly over TCP; uses simple ASN.1
- RFC 1777 (v2), v3 in progress ➠ referral
- to be used by Four11, . . .

LDAP operation

- `bind`: authenticate
- `search`:
  - limit time
  - attributes to be returned
- compare assertion
- add, modify, delete object
- `unbind`: terminate session

LDAP queries

```
Filter ::=
         CHOICE {
             and                  [0] SET OF Filter,
             or                   [1] SET OF Filter,
             not                  [2] Filter,
             equalityMatch        [3] AttributeValueAssertion,
             substrings           [4] SubstringFilter,
             greaterOrEqual       [5] AttributeValueAssertion,
             lessOrEqual          [6] AttributeValueAssertion,
             present              [7] AttributeType,
             approxMatch          [8] AttributeValueAssertion
         }
AttributeValueAssertion ::=
    SEQUENCE {
        attributeType          AttributeType,
        attributeValue         AttributeValue
    }
AttributeType ::= LDAPString
                -- text name of the attribute, or dotted
                -- OID representation

AttributeValue ::= OCTET STRING
```

whois++

- RFC 1835 (architecture)
- simple templates = data schema
- servers export "forward knowledge" for their entries
- *centroid*: union of values for each template

```
Record 1                           Record 2
Template: Person                   Template: Person
First-Name: John                   First-Name: Joe
Last-Name: Smith                   Last-Name: Smith
Favourite-Drink: Labatt Beer       Favourite-Drink: Molson Beer

Record 3
Template: Domain
Domain-Name: foo.edu
Contact-Name: Mike Foobar
```

whois++ centroid

➠ centroid for this server

```
Template:         Person
First-Name:       Joe
                  John
Last-Name:        Smith
Favourite-Drink:Beer
                  Labatt
                  Molson

Template:         Domain
Domain-Name:      foo.edu
Contact-Name:     Mike
                  Foobar
```

whois++

```
author=chris and template=user
schoultz and rick;search=lstring  ; leading substring

# FULL USER SERVERHANDLE1 PD45
 Name: Peter Deutsch
 email: peterd@bunyip.com
# END
```

whois++

server structure ➠

- each database has a unique handle
- servers are polled by other to extract centroids
- servers poll others for centroids

➠ not necessarily hieararchical

URNs

- persistent resource *identifiers* (names) $\leftrightarrow$ URLs = location
- for long-lived resources (papers, newspaper article archives, legal records, Dilbert cartoons, $\ldots$)
- URLs, URNs: not always a single file (ASCII, HTML; English, French)
- resolution of URN may be time-dependent (current weather map)
- URN $\stackrel{directory,hints}{\rightarrow}$ URLs, URCs
- examples: ISBN, MIME content-ids (cid)

Uniform Resource Characteristics (URC)

meta information:

- bibliographic citation
- digital signature
- revision history

➠ not currently available

URN syntax

```
[urn:] <NID> ":" <NSS>
```

NID: namespace identifier (ISBN)

NSS: namespace specific syntax

Examples (fictional):

```
urn:isbn:1-23485-8-29
cid:foo@huh.com
```

URN resolution

- single directory won’t scale
- typical: local hints for resolution
- fallback: URN-resolution-service discovery service (UDS)