The Domain Name System
(DNS)

Prof. Jean-Yves Le Boudec
Prof. Andrzej Duda
ICA, EPFL

CH-1015 Ecublens
http://lcawww.epfl.ch

Domain Name System: DNS

- Objective of DNS
  - support user friendly naming of resources: computers, printers, mailboxes,…
  - hide IP address changes
  - distribute naming authority
  - distribute the database
  - used primarily for system names and email

- Names and addresses
  - “domain name”: high level identifier; eg. lrcsuns ssc.epfl.ch
  - “IP address”: low level identifier
    - related to routing, physical topology
  - “level 2 addresses”: low level identifiers
    - MAC address: serial number of communication interface
    - ATM address: combination of MAC and route related address
  - “email address”: high level address used for email
    - eg. gwen.nedeleg@ssc.epfl.ch
    - email addresses are mapped to domain names:
      gwen.nedeleg@ssc.epfl.ch -> gwen.nedeleg.ssc.epfl.ch
DNS Example

application program

name resolver

DNS query

1

1

2

stisunl

nameserver 129.178.15.7
nameserver 129.178.15.8
domain epfl.ch

IP dest addr = 129.178.15.7
protocol = UDP
source port = 1267 dest port = 53
DNS message =
  header = query,
  question = (QNAME="disun3.epfl.ch." QTYPE=A)

2

IP dest addr = 129.178.156.24
protocol = UDP
source port = 53 dest port = 1267
DNS message =
  header = response, AA
  question = (QNAME="disun3.epfl.ch." QTYPE=A)
  answer = (disun3.epfl.ch. TTL=86400 TYPE=A 128.178.79.9)

resolv.conf

nameserver 129.178.15.7
nameserver 129.178.15.8
domain epfl.ch

Resource Records (RRs)
keyed by domain names

zone data (authoritative data)

disun3.epfl.ch. 86400 A 128.178.79.9
in-inr.epfl.ch. 86400 A 128.178.156.1
86400 A 128.178.182.5

zone data (non-authoritative data)
ezinfo.ethz.ch. 1770 A 129.132.2.72

Domain Name Tree

every node on the tree represents one or a set of resources
every node on the tree has a label (lrcsuns) and a domain name (lrcsuns.epfl.ch)
domain name = sequence of labels, \leq 64 bytes per label
  examples: www.zurich.ibm.com, lrcsuns.epfl.ch, ezinfo.ethz.ch, ee.ethz.ch
  names have the same syntax for subdomains or individual resources
Name Authority

- hierarchical name authority
  - top level: Internic
  - any organization can apply to become authority for a subdomain examples:
    - SWITCH for ch. and li.
    - EPFL for epfl.ch.
  - any authority can create subdomains and delegate recursively unilaterally

- zones:
  - definition: zone = a connected subset of nodes
    - property: a zone has one single node closest to the root (top node, used to name the zone)
  - definition: zone \( Z_1 \) is a subzone (or child) of zone \( Z_0 \) iff the top node of \( Z_1 \) is connected to a node in \( Z_0 \);
  - name authority matches zone boundaries:
    - names and subzones, can be created and deleted by the authority responsible for a zone; examples:
      - zurich.ibm.com is a subzone of ibm.com
      - zone zurich.ibm.com has authority delegation from ibm.com.

Fully Qualified Domain Names

- complete domain name = fully qualified domain name (FQDN)
  - ends with a period (".")
  - trailing period usually hidden by the user interface software
- incomplete names are completed by local resolver
  - or add local domain suffix: lrcsuns -> lrcsuns.epfl.ch.
The DNS distributed database

DNS offers one distributed world-wide database
- distributed according to the zone concept: every zone has a master file describing all records under the zone’s authority
- name servers hold their part of the database
  - for one zone, at least two name servers have the zone information, copied from master file
    - example: stisun1.epfl.ch, stisun2.epfl.ch; dns1.ethz.ch, dns2.ethz.ch
  - zone information held by the name server is called authoritative data
  - one name server may hold zone data for one or more zones
- zone data contains pointers to name servers holding authoritative data for subzones
- all name servers know IP addresses of root servers (name servers for the top level zones)

Query Processing and Cached Data

query processing
- resolver associated with an application sends a query to a name server
- name server responds with answer or with pointer to another server

query processing can be
- iterative
- recursive: server responds with final answer
  - server acts as an intermediate resolver
  - recursive operation only if requested in query and server accepts it
  - root servers never support recursive operation

name servers usually cache some information for nodes outside their zones
- recently obtained information is cached when acting recursively
- every record has a TTL field (ex: 1 day) used for cache management
- cached data is not authoritative
Example: Query Processing

1. query, RD=yes

2, 4. query, RD=no

3. answer
   answer = ""
               NS  ns.austin.ibm.com.
               NS  ns.almaden.ibm.com."
   additional="watson.ibm.com.    A 192.35.232.34
             ns.austin.ibm.com. A 129.34.139.4
             ns.almaden.ibm.com A 198.4.83.134"

5, 6. answer
   answer = "www.zurich.ibm.com.  A 193.5.61.131"

Replication

- zone data is **replicated** in several servers responsible for the zone
  - primary server holds master file on disk
  - secondary servers poll primary servers (ex: every 3 hours)
    - using the SERIAL field in zone data
    - copying is called **zone transfer**; uses TCP (queries usually use UDP)
  - changes in zone data by system manager:
    - update master file
    - signal primary name server to reload; new value of SERIAL field automatically created
    - secondary servers will discover the change automatically
- zone data in secondary servers is authoritative

example: in which name servers can these RRs appear as zone or cache data:

disun3.epfl.ch. 86400 A 128.178.79.9
### Resource Record Types and Message Formats

<table>
<thead>
<tr>
<th>TYPE</th>
<th>value and meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1 IPv4 address</td>
</tr>
<tr>
<td>NS</td>
<td>2 an authoritative name server</td>
</tr>
<tr>
<td>CNAME</td>
<td>5 the canonical name for an alias</td>
</tr>
<tr>
<td>SOA</td>
<td>6 marks the start of a zone of authority</td>
</tr>
<tr>
<td>PTR</td>
<td>12 a domain name pointer</td>
</tr>
<tr>
<td>HINFO</td>
<td>13 host information</td>
</tr>
<tr>
<td>MINFO</td>
<td>14 mailbox or mail list information</td>
</tr>
<tr>
<td>MX</td>
<td>15 mail exchange</td>
</tr>
<tr>
<td>TXT</td>
<td>16 text strings</td>
</tr>
<tr>
<td>AAAA</td>
<td>28 IPv6 address</td>
</tr>
</tbody>
</table>

---

#### Examples of Records

- **MX records**: used by email application
  - example:
    ```
    di.epfl.ch. MX 10 dimail.epfl.ch.
    di.epfl.ch. MX 20 disunmm2.epfl.ch.
    ```
  - possible use: ?
- **PTR records**: inverse mapping IP addr -> domain name
  - example:
    ```
    5.182.178.128.in-addr.arpa PTR in-addr
    6.5.0.0.0.0.0.0.0.0.0.0.0.0.0.5.6.3.0.0.0.1.4.0.0.e.w.4.0 PTR lrcpc3
    ```
  - used for verifying names
  - zone date should contain PTR records for all systems in the zone
- **other records**: ISDN number, ATM address (proposed)
$ nslookup -querytype=PTR 193.5.61.131
Server: stisun1.epfl.ch
Address: 128.178.15.8
131.61.5.193.in-addr.arpa name = uetliberg.zurich.ibm.com
61.5.193.in-addr.arpa nameserver = ns1.zurich.ibm.com
61.5.193.in-addr.arpa nameserver = scsnms.switch.ch
61.5.193.in-addr.arpa nameserver = swidir.switch.ch
ns1.zurich.ibm.com internet address = 193.5.61.131
scsnms.switch.ch internet address = 130.59.10.30
scsnms.switch.ch internet address = 130.59.72.10
swidir.switch.ch internet address = 130.59.130.30

$ nslookup www.zurich.ibm.com
Server: stisun1.epfl.ch
Address: 128.178.15.8
Non-authoritative answer:
Name: www.zurich.ibm.com
Address: 193.5.61.131

$ nslookup -querytype=NS zurich.ibm.com 129.34.139.4
Server: watson.ibm.com
Address: 129.34.139.4
zurich.ibm.com nameserver = ns1.zurich.ibm.com
zurich.ibm.com nameserver = watson.ibm.com
ns1.zurich.ibm.ch internet address = 193.5.61.131
watson.ibm.com internet address = 129.34.139.4

Example of Zone Data (ch.)

ch.epfl.ch

- where are the servers responsible for that zone?
- what are the data indicating authority delegation?
- what records are authoritative
- what would change if ch. and switch.ch were in the same zone?
- what is the answer to a query “scsnms.ethz.ch A”
Example of Zone Data (epfl.ch.)

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>stisun1.epfl.ch</td>
<td>NS</td>
<td>128.178.100.7</td>
</tr>
<tr>
<td>stisun2.epfl.ch</td>
<td>NS</td>
<td>128.178.15.7</td>
</tr>
<tr>
<td>stisun1.epfl.ch</td>
<td>A</td>
<td>128.178.100.7</td>
</tr>
<tr>
<td>stisun2.epfl.ch</td>
<td>A</td>
<td>128.178.15.7</td>
</tr>
<tr>
<td>lrcsuns.epfl.ch</td>
<td>A</td>
<td>128.178.156.24</td>
</tr>
<tr>
<td>lrcwww.epfl.ch</td>
<td>CNAME</td>
<td>lrcsuns.epfl.ch</td>
</tr>
<tr>
<td>lrcftp.epfl.ch</td>
<td>CNAME</td>
<td>lrcsuns.epfl.ch</td>
</tr>
<tr>
<td>ssc.epfl.ch</td>
<td>MX</td>
<td>10 sicmail.epfl.ch</td>
</tr>
<tr>
<td>*.di.epfl.ch</td>
<td>MX</td>
<td>10 sicmail.epfl.ch</td>
</tr>
</tbody>
</table>

24.156.178.128.in-addr.arpa PTR lrcsuns.epfl.ch

... (other records) ...

Exercise:
- where are the servers responsible for that zone?
- what are the data indicating authority delegation?
- what records are authoritative?
- what is the answer to a query "lrcwww.epfl.ch A"

RFC 1034 says:

1. Set or clear the value of recursion available in the response depending on whether the name server is willing to provide recursive service. If recursive service is available and requested via the RD bit in the query, go to step 5, otherwise step 2.

2. Search the available zones for the zone which is the nearest ancestor to QNAME. If such a zone is found, go to step 3, otherwise step 4.

3. Start matching down, label by label, in the zone. The matching process can terminate several ways:
   a. If the whole of QNAME is matched, we have found the node.
      If the data at the node is a CNAME, and QTYPE doesn't match CNAME, copy the CNAME RR into the answer section of the response, change QNAME to the canonical name in the CNAME RR, and go back to step 1. Otherwise, copy all RRs which match QTYPE into the answer section and go to step 6.
   b. If a match would take us out of the authoritative data, we have a referral. This happens when we encounter a node with NS RRs marking cuts along the bottom of a zone. Copy the NS RRs for the subzone into the authority section of the reply. Put whatever addresses are available into the additional section, using glue RRs if the addresses are not available from authoritative data or the cache. Go to step 4.
c. If at some label, a match is impossible (i.e., the corresponding label does not exist), look to see if a the "*" label exists.

    If the "*" label does not exist, check whether the name we are looking for is the original QNAME in the query or a name we have followed due to a CNAME. If the name is original, set an authoritative name error in the response and exit. Otherwise just exit.

    If the "*" label does exist, match RRs at that node against QTYPE. If any match, copy them into the answer section, but set the owner of the RR to be QNAME, and not the node with the "*" label. Go to step 6.

4. Start matching down in the cache. If QNAME is found in the cache, copy all RRs attached to it that match QTYPE into the answer section. If there was no delegation from authoritative data, look for the best one from the cache, and put it in the authority section. Go to step 6.

5. Using the local resolver or a copy of its algorithm (see resolver section of this memo) to answer the query. Store the results, including any intermediate CNAMEs, in the answer section of the response.

6. Using local data only, attempt to add other RRs which may be useful to the additional section of the query. Exit.

---

**Name Resolution**

0 Application requests name resolution on local host

1 resolver sends query to name server
   /etc/resolv.conf on many systems points to the name server
   if no pointer, then local host activates its own name server
   - resolver usually requests recursive query
   - response is processed until an answer is found

1 name server acting recursively plays the role of a resolver for that query

1 host resolvers usually do not cache responses (stub resolvers), but name servers do (full resolvers)
DNS Components Overview

DNS

References: DNS

- Halsall, chapter 13.2
- RFCs: 1032, 1033, 1034, 1035, 1591
- nslookup, host, resolver, named
NetBIOS

- Windows uses NetBIOS for transactions and distributed file system
- NetBIOS is a programming interface (as Sockets is) which uses NetBIOS names instead of (IP address, port number)
  - example: ICARE, ICA118PC29
- name resolution was done originally by LAN broadcast
- in modern installations, done by NetBIOS name server
  - WINS

DNS: Conclusion

- high level names
  - global world-wide
  - decouple names used by humans from IP addresses
    - names not related to routing
  - decouple logical names from machine names
- distributed database with simple database mechanisms
  - loose consistency in records
  - strict hierarchical database with zone concept
  - high survivability thanks to replication within one zone
  - caching to improve performance
- DNS has become a key component of the Internet
  - survivability and security are key issues