INP Grenoble
DÉPARTEMENT TÉLÉCOMMUNICATIONS

Application Layer: DNS

Prof. Andrzej Duda
duda@imag.fr

http://duda.imag.fr

1

Overview

- Learn about protocols by examining popular application-level protocols
  - DNS - name service

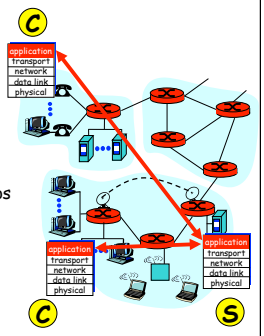
2

Applications and application-layer protocols

Application: communicating, distributed processes
- running in network hosts in "user space"
- exchange messages to implement applications
- e.g., email, file transfer, the Web

Application-layer protocols
- define messages exchanged by apps and actions taken
- one "piece" of an app
- user services provided by lower layer protocols

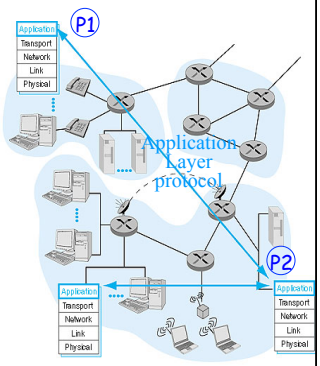
Typical application-layer has two pieces: *client* and *server*



3

Network applications: some definitions

- A process is a program that is running within a host.
- Within the same host, two processes communicate with interprocess communication defined by the OS.
- Processes running in different hosts communicate with an application-layer protocol
- A user agent is an interface between the user and the network application.
  - Web: browser
  - E-mail: mail reader
  - streaming audio/video: media player

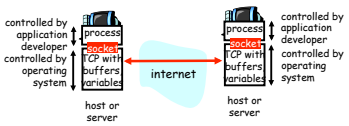


4

Application-layer protocols (cont).

API: application programming interface
- defines interface between application and transport layer
- socket: Internet API
  - two processes communicate by sending data into socket, reading data out of socket

How does a process "identify" the other process with which it wants to communicate?
- IP address of host running other process
- "port number" - allows receiving host to determine to which local process the message should be delivered

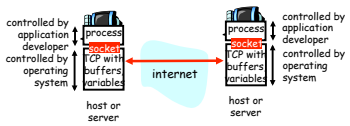


5

Application-layer protocols (cont).

API: application programming interface
- defines interface between application and transport layer
- socket: Internet API
  - two processes communicate by sending data into socket, reading data out of socket

How does a process "identify" the other process with which it wants to communicate?
- IP address of host running other process
- "port number" - allows receiving host to determine to which local process the message should be delivered



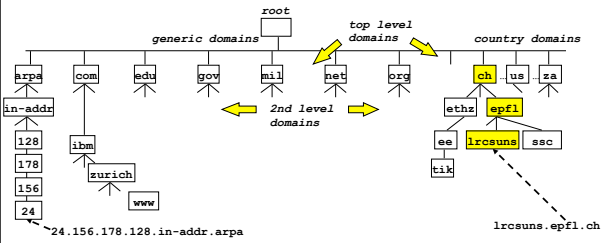
6

DNS (Domain Name System)

- Distributed database
  - relation **`name - IP address`**
  - delegation of authority - who updates the database?
  - name servers
    - primary, secondary - authoritative data (sources autorisées)
    - cache - non-authoritative data (sources non autorisées)
  - resolver:
    - **`gethostbyname`**
    - **`gethostbyaddr`**
- Hierarchical name space
  - similar to Unix pathnames, but reversed
  - label separator: . instead of /

7

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`**)

8

DNS names

- Node
  - label <= 63 characters (letters, digits, and -), lower case = upper case
- Name
  - list of labels separated by .
    - **`www.epfl.ch.`** *(fully qualified domain name)*
    - **`lcawww`** (local name - evaluated with respect to the local domain)
- 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

9

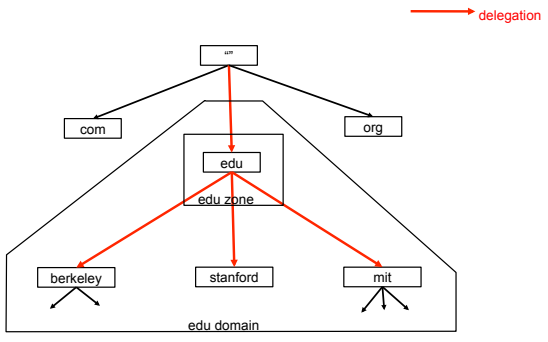
Name management

- Zone = a connected subset of nodes
  - property: a zone has one single node closest to the root (top node, used to name the zone)
  - 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.`**
  - at least 1 name server per zone (port 53)
    - primary, secondary - copy of the primary
    - **`/etc/resolv.conf`**:
    - replication - secondary servers
    - cache - data kept for 1 day

```
nameserver 128.178.15.7
domain epfl.fr
```

10

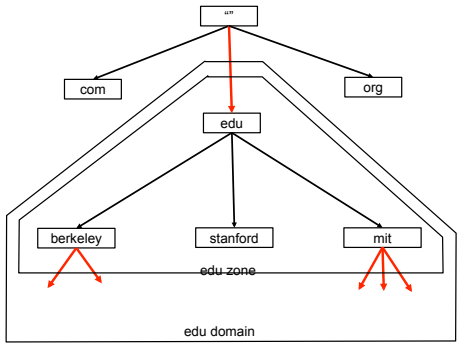
Zones and Domains



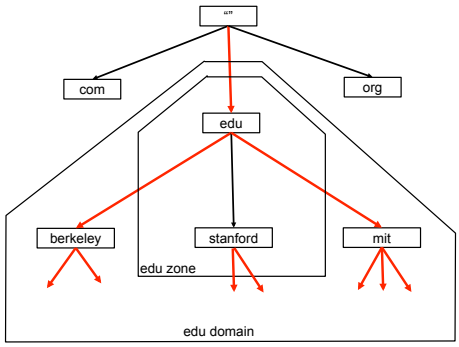
Zones and Domains

- Domains:
  - subtrees of the name space
  - domain x.y.z contains all nodes below x.y.z
  - independent of delegation relationships
- Zones:
  - nodes in name tree under single administrative control
  - zone x.y.z does not contain those nodes below x.y.z for which the zone delegates to another zone
  - delegation relationships define its boundaries

Zones and Domains

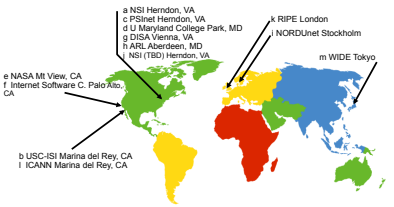


Zones and Domains



DNS Root Name Servers

- Contacted by local name server that can not resolve name
- Root name server:
  - contacts authoritative name server if name mapping not known
  - gets mapping
  - returns mapping to local name server



13 root name servers worldwide

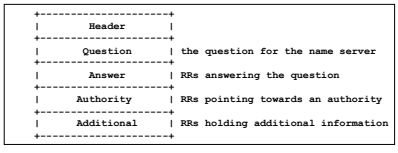
Query Processing and Cached Data

- 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
    - example: question "`www.zurich.ibm.com. A`" from a node at EPFL; response is a pointer to a name server responsible for zone `ibm.com.`
- 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 cache information
  - cached data is not authoritative

16

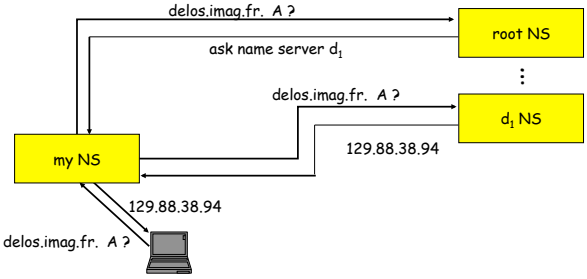
DNS Resource Record Types and Message Formats

TYPE	value and meaning
A	1 Ipv4 address
NS	2 an authoritative name server
CNAME	5 the canonical name for an alias
SOA	6 marks the start of a zone of authority
PTR	12 a domain name pointer
HINFO	13 host information
MINFO	14 mailbox or mail list information
MX	15 mail exchange
TXT	16 text strings
AAAA	28 IPv6 address



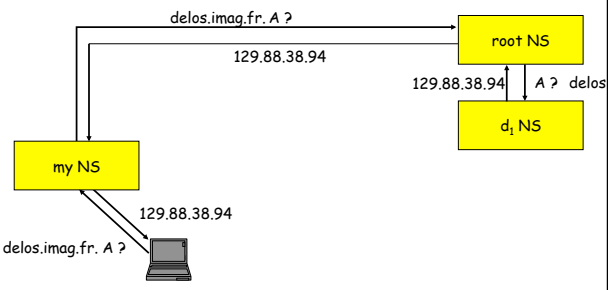
17

Iterative search



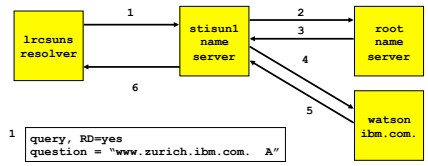
18

Recursive search



19

Example: Query Processing



1
```
query, RD=yes
question = "www.zurich.ibm.com.  A"
```

2,4
```
query, RD=no
question = "www.zurich.ibm.com.  A"
```

3
```
answer
question = "www.zurich.ibm.com.      A"
answer = ""
autority= "ibm.com.     NS  watson.ibm.com.
                        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
question = "www.zurich.ibm.com.     A"
answer = "www.zurich.ibm.com.     A  193.5.61.131"
```

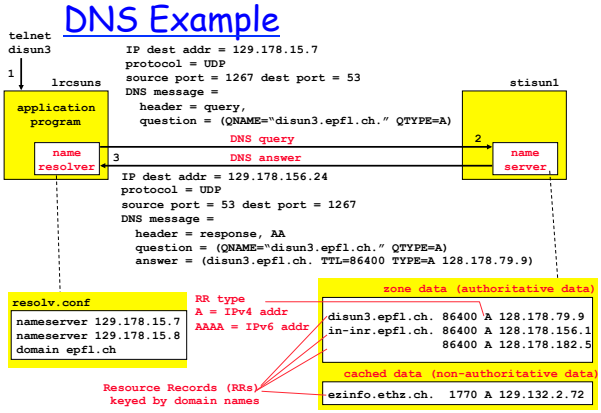
20

Replication

- Zone data is replicated
  - primary server holds master file on disk
  - secondary servers poll primary servers (ex: every 3 hours)
    - using the SERIAL field in the 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

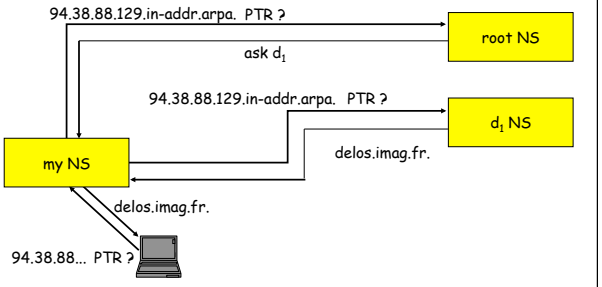
21

DNS Example



22

Pointer query



23

Examples: Queries/ Answers

1
```
$ 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
```

2
```
$ 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.ch
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
```

3
```
$ 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.ch
61.5.193.in-addr.arpa   nameserver = ns1.zurich.ibm.ch
61.5.193.in-addr.arpa   nameserver = scsnms.switch.ch
61.5.193.in-addr.arpa   nameserver = swidir.switch.ch
ns1.zurich.ibm.ch       internet address = 193.5.61.131
scsnms.switch.ch        internet address = 130.59.10.30
scsnms.switch.ch        internet address = 130.59.1.30
swidir.switch.ch        internet address = 130.59.72.10
```

24

DNS

- Name servers all over the world
- Scalable
  - distribution and authority delegation
  - caches for efficiency
  - replication for fault tolerance
- One of the key features of the Internet

25

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
- Root servers
  - 13 servers distributed all over the world
  - any primary server needs to know their addresses

26