Domain Name System & Email

Unit 05 - Hands-On Networking - 2018

Prof. Dr.-Ing. Thorsten Herfet, Andreas Schmidt, Pablo Gil Pereira

Telecommunications Lab, Saarland Informatics Campus, 20th Feb. 2018

Recap

- **Addresses** are an important aspect of networks.
- Some representations in networking are **human-readable**, but most aren't.
- Application layer is where **services** are implemented.
- **URLs** point to **named hosts**.
- **Cache** things that do not change frequently.

Real-World Identifiers

Which ones do you know?

Numbers

- Passport Number
- Social Security Number
- Phone Number

Names

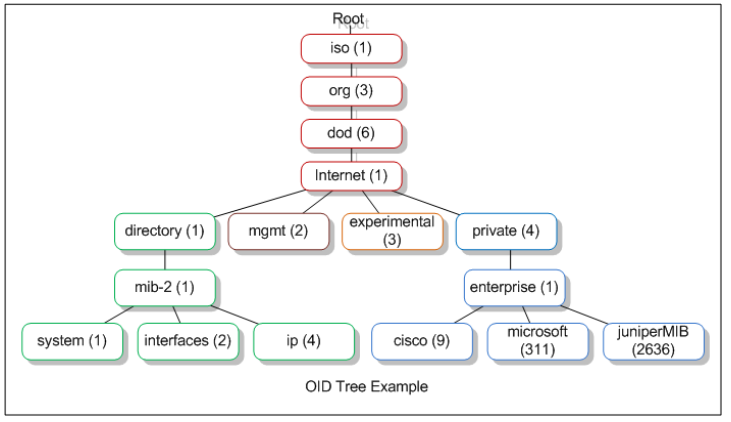
- Full Name
- Postal Address
  (Street, City, Country)
- Facebook Handle

Any disadvantages you can think of?

- Numbers are **hard to remember** for humans.
- Names are often more convenient but **seldomly unique**.

Networking-World Identifiers

- IP addresses:
  - IPv4: four 8-bit decimal number
  - IPv6: eight 16-bit hex number
- MAC addresses (6-byte hex number)
- Port numbers (0-65535)
- SNMP MIB numbers, e.g. 1.5.2.3.7
- SHA Hashes



SNMP (http://bit.ly/293GA96)

Problems

- Numbers or cryptic letters only:
  - Straightforward for machines (fixed length, fixed memory consumption).
  - Space-efficiency high for numerical presentation.
- Again hard to remember for humans (usability).
- IP addresses are nowadays only temporarily used.

Memorize or Lookup

❓ How do we solve the *remember* problem in phone networks etc.?

The First Yellow Page(s) for Networks

- Late 1960s: First funding of the *ARPAnet*, the first wide-area computer network by *US Department of Defense's Advanced Research Projects Agency* (DARPA)
- 1970s: ARPAnet only contained a few hundred hosts. Mapping name-to-address was done using `HOSTS.TXT` (compiled into `/etc/hosts` on UNIX):

```
192.168.2.4     weatherwax.arpa.net
192.168.2.5     vimes.arpa.net
192.168.2.7     moist.arpa.net
192.168.2.10    susan.arpa.net
...
```

- **Distribution:** Using FTP downloads and uploads.
- **Updates / Changes:** Using email.
  (sent to Stanford Research Institute Network Information Center (SRI-NIC))

Can you think of any problems?

- Scalability (consumption and updating).
- Collision probability.
- Consistency.

Domain Name System (DNS) (RFC1035)

1983: RFC882 defines the concept and facilities for *Domain Names*.

1987: RFC1035 introduces one of the most important services on the Internet.

Intuition: DNS = *distributed* database of *host information*, indexed by *domain names*.

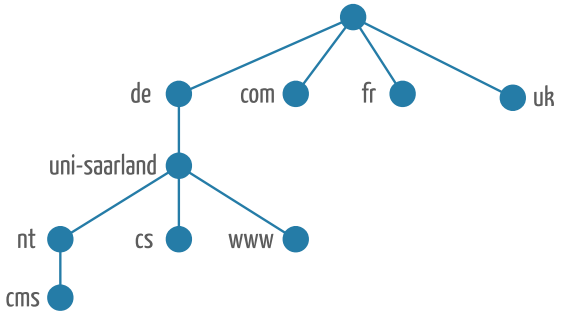
Features (abriged...)

- Address Lookup
  Translate a name (e.g. `uni-saarland.de`) to an IP address (`134.96.7.179`)
- Aliasing
  Provide **multiple** names for the same IP address.
- Service Lookup
  Retrieve IP and port values for a certain name.
- Reverse Lookup
  Find the names for a given IP address.

The Domain Namespace

Namespace: Tree.

- Nodes labeled (e.g. `com`).
  At max 63 characters.
- *Root* labeled "" (empty).
  Similar to `/` in file system.
- Height limited to 127.
  Practically never exceeded.



Domain Name: Path along tree.

- Traverse from leaf to root.
- Dots separate labels.
- Example: `www.cs.uni-saarland.de`
- Absolute, if a dot is put in the end:
  `www.cs.uni-saarland.de.`
- Also called: FQDN.
  Fully Qualified Domain Name

Domain: Subtree of the namespace.

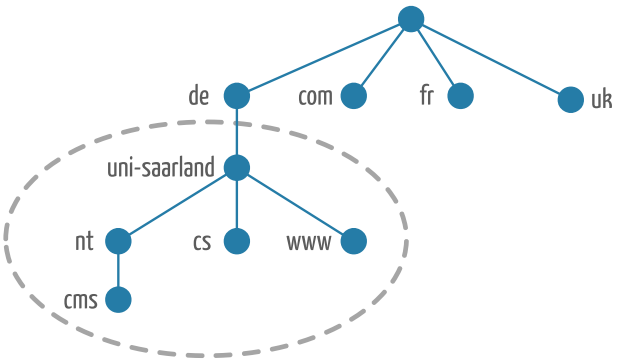
- Name of domain is the *Domain Name* of topmost node in the subtree.

Nameserver: Stores and serves domain information.

Quizzes

❓ What is the *domain name* of the marked subtree?

`uni-saarland.de`

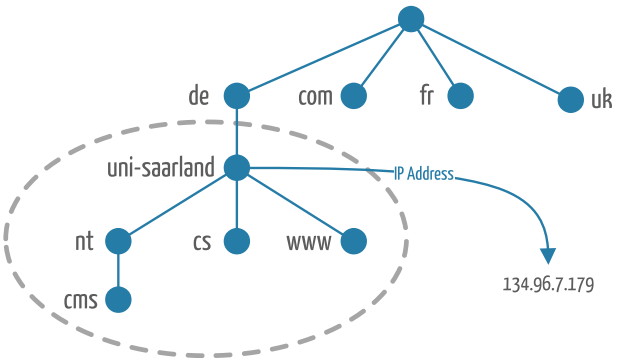


❓ `cms.nt.uni-saarland.de.` is part of which domain(s)?

- `nt.uni-saarland.de`
- `uni-saarland.de`
- `de`

Versatile Domain Names

- Domain Names at *leaves*
  - usually refer to individual hosts
  - point to network address or
  - hardware information or
  - mail-routing information.
- *Interior* Nodes can
  - name hosts and
  - point to domain information.



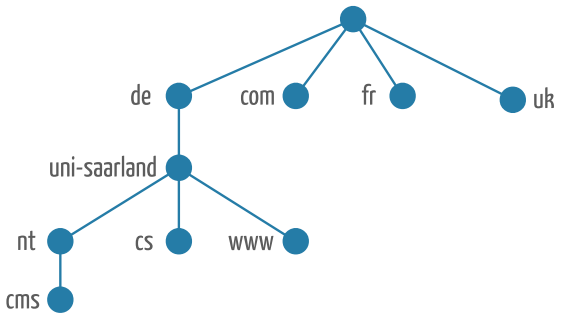
HON

Subdomains and Levels

- Subtrees of a domain are called **subdomain**.
- Domains are assigned to a **level** indicating how far away from *root* they are.
  - *Top-Level Domain* (TLD): Child of root.
  - *First-Level Domain*: Child of root.
    But often referred to as TLD.
  - *Second-Level Domain*: Child of a first-level domain.
  - *nth-Level Domain*: Child of a (n-1)-level domain.

Quiz

❓ What are subdomains of the only second-level domain in the graphic?



A: `de`

B: `uni-saarland.de`

C: `www.uni-saarland.de`

D: `cms.nt.uni-saarland.de`

⚠ Answer:

✖ A: Actually the parent of it.

? B: The domain itself (pathological).

✔ C: Exactly!

✔ D: Subsubdomain = Subdomain

Resource Records `(name, value, type, ttl)`

`type=A`:

- `name`: hostname
- `value`: IPv4 address

`type=AAAA`:

- `name`: hostname
- `value`: IPv6 address

`type=MX`:

- `name`: domain
- `value`: name and priority of mailserver

`type=NS`:

- `name`: domain (e.g. example.com)
- `value`: hostname of authoritative name server for this domain

`type=CNAME`:

- `name`: alias for some canonical (real) name
- `value`: canonical name
- Example: `www.uni-saarland.de` for `webuni.rz.uni-saarland.de`

`type=PTR`:

- `name`: IP address (e.g. `134.96.7.179`)
- `value`: domain name (`webuni.rz.uni-saarland.de`)

The Internet Domain Namespace

⚠ **Labels** have no particular meaning associated with them!
Administration of namespace decides on semantics and naming schemes.

On higher levels, there are certain **traditions**.

Top-Level Domains

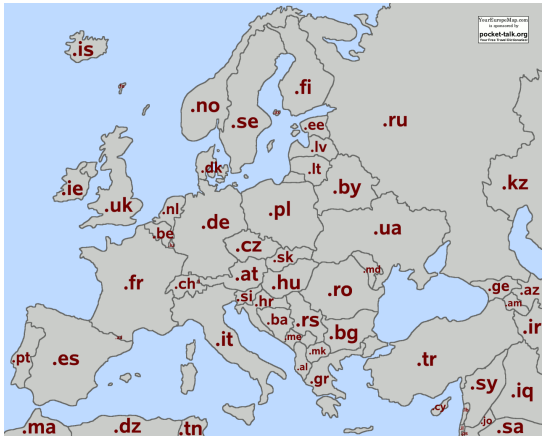
- **com** (commercial)
  e.g. google.com, facebook.com
- **edu** (educational)
  e.g. berkley.edu, mit.edu
- **gov** (governmental)
  e.g. nasa.gov, whitehouse.gov
- **int** (international)
  e.g. nato.int, esa.int
- **mil** (military)
  e.g. army.mil, navy.mil
- **net** (network)
  Originally: Network Infrastructure.
  Today: open for all (e.g. asp.net).
- **org** (organizational)
  Originally: Non-commercials.
  Today: open for all.

Original 7 called *generic top-level domains* or *gTLDs*.

More Top-Level Domains and Traditions

Country-Code TLDs (ccTLDs)

- Top-level domains were reserved for individual countries.
  But not necessarily created.
- Domain names follow ISO 3166.
  Except Great Britain - who wonders?
  - which uses `uk` instead of `gb`.



Source

New TLDs

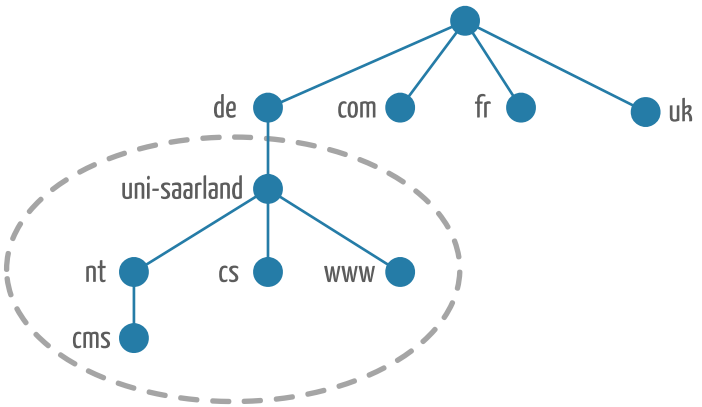
- 2000: Introduction of new gTLDs. `aero`, `biz`, `coop`, `info`, `museum`, `name`, `pro`
- 2005: Sponsored TLDs introduced. `jobs`, `travel`, `mobi`
- New TLDs have been created over the last years: `.saarland`, `.berlin`

Traditions

Follow the US example, e.g. `edu.au`, `com.au`, `ac.uk`, and `co.uk`.

Delegation

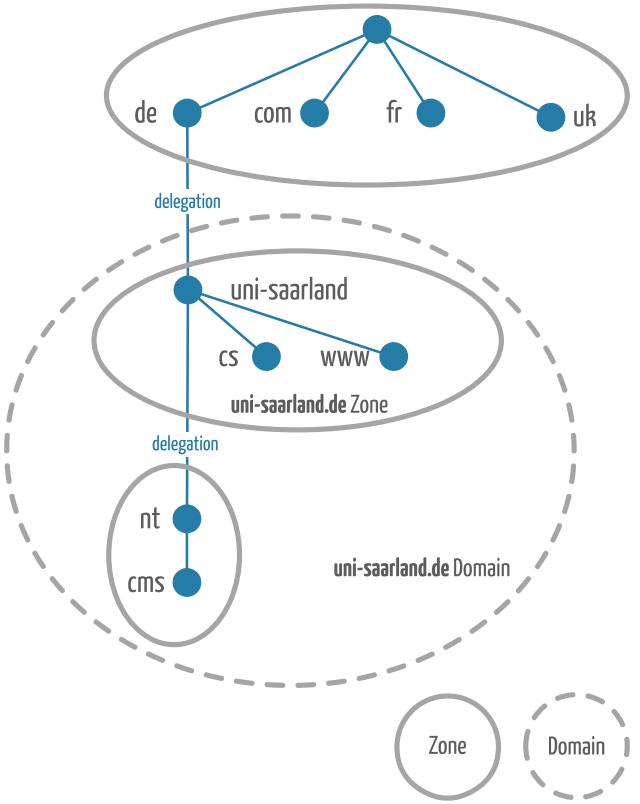
- Each subdomain can be *delegated* to another server.
- `uni-saarland.de` managed by Hochschul-Informations Zentrum (HIZ):



- Typically, organizations manage some domain names on their own and delegate certain subdomains to others.
- **Delegation**: Assigning responsibility for a subdomain to another organization or organizational unit.

Zones

- **Zone:** Part (not subtree) of domain namespace.
  Note: Zones are not domains.
- Nameservers manage one or more zones, hence beeing **authoritative**.
  Data loaded from file or other server.
- Zones contain
  - all the domain names the domain with the same domain name contains
  - except delegated subdomains
- *Delegating* a domain involves a pointer to the nameservers that are *authoritative* for the subdomain.



Resolving Names

In order to resolve a domain name (e.g. `cms.nt.uni-saarland.de`) we require:

Resolvers

- Client applications that access nameservers.
  - Compose query to nameserver.
  - Interpret response.
  - Return and display information.
- Knows **nothing** about domain namespace.
  Therefore, sometimes also called *stub resolver*.

Nameservers

- Potentially knows about parts of the domain namespace.
- Definitely knows who to ask for the answers.
- Typically contain a **resolver**.

Protocol Messages

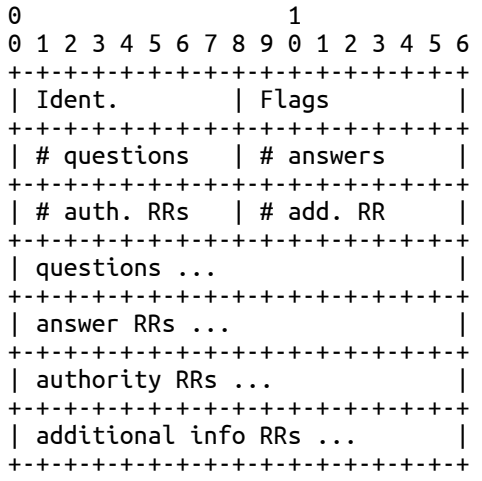
Messages

Query and Reply (use the same format).

Header Contents

- Identification (16bit): Number to relate queries and replies.
- Flags (16bit):
  - Query or Reply.
  - Recursion desired.
  - Recursion available.
  - Reply is authoritative.
- Questions: Name, Type pairs.
- Answers: RR in response to query.
- Authority: records for authoritative servers.
- Additional Information: further RRs that might be helpful.

Header Format



Quiz

Which transport layer protocol is used for DNS?

A: TCP (100% reliable, throughput limited, no time bounds).

B: UDP (unreliable, throughput unlimited, no time bounds).

C: Other Transport Protocol.

Answer:

A: Possible, but seldomly used, (RFC7766). Operations such as "DNS Zone Transfers" use it.

C: Nope.

B: True. Reliability is implemented by resending single packetized messages. No need for more sophisticated error control mechanisms.

Local Nameserver

- Not necessarily managing a zone (infact most of the time not).
- Each ISP (residential ISP, company, university) has one (default nameserver).
- Resolver's DNS query message is sent to this server first.
- Provides **cache** of recent lookups (may be out of date).
- Acts as **proxy**, forwards query into the DNS hierarchy.

Root Nameservers

In case a local name server can not resolve a name, contact **root nameserver**.

Root Name Server (13 different ones):

- First point of inquiry for a completely unknown name mapping.
- IP addresses of top-level-domain (TLD) DNS servers are known.
- Replicated to ensure reliable and stable operation.

Source

Resolving Process - Iterative

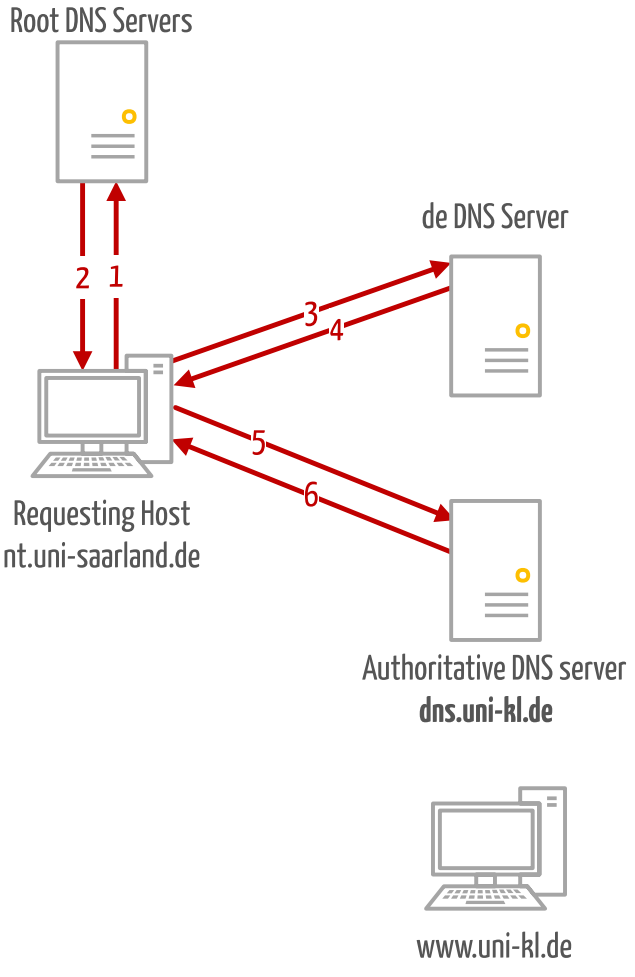
Scenario: Host `nt.uni-saarland.de` wants IP address of `www.uni-kl.de`.

Iterated Query:

- Contacted server replies with name of next server to contact.
- Does not know about complete name, but where to find suffix.
- Forwarding from server to server.
- Responses cached along the way.

Caching:

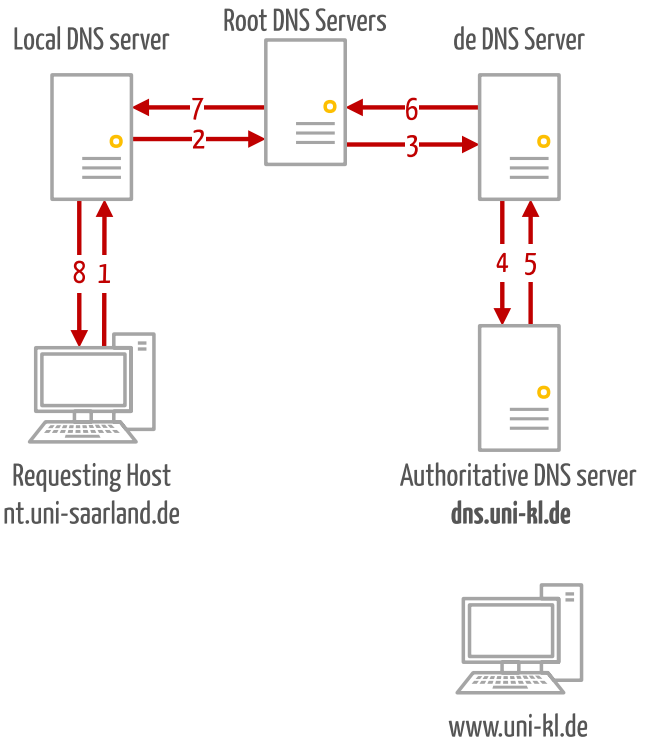
- Responses cached along the way.
- Another query for `cs.uni-kl.de` will immediately be answered by the local name server.



Resolving Process - Recursive

Recursive Query:

- Ask any other server to do the lookup for one.
- Puts the resolution burden on the contacted server.
- Introduces heavy load at upper layers of hierarchy.
- Not ideal.



Quiz

Why is the DNS system implemented as a distributed database?

Answer:

Consider a centralized version instead:

- Single point of failure.
- High traffic volume, as everyone is looking for answers.
- Central database might be spatially distant.
- Maintaining it would be hard, as everyone needs to write to a central instance.

Problem: Does not scale.

Reverse Lookup

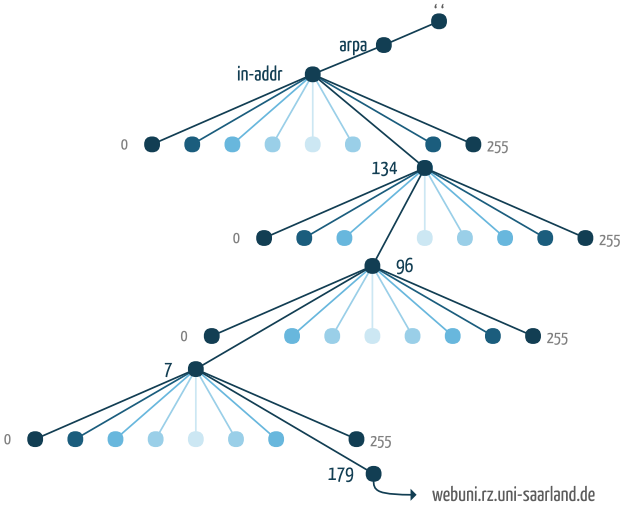
Assume you run a server that provides you with an access log (incl. IP addr.).

How do you find out more about e.g. the IP address `134.96.7.179`?

Lookup in the `in-addr.arpa` domain.

- **Resource Records:**
  - *Name*: IP address
  - *Value*: DNS Name
  - *Type:* `PTR`
- **Domain Name:** `179.7.96.134.in-addr.arpa`
- **Command:**

```
$ dig +short ptr 179.7.96.134.in-addr.arpa
webuni.rz.uni-saarland.de
```



IPv6: uses `ip6.arpa`, e.g.

```
0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
```

Register New Domain

- **Example:** "HON"
- Register name `hon.de` at **domain registrar**.
- Provide names, IP addresses of authoritative nameservers (primary, secondary, ...).
- Registrar inserts records of the following format at .de TLD server:
  - `(hon.de, dns1.hon.de, NS)`
  - `(hon.de, dns2.hon.de, NS)`
  - `(hon.de, 192.0.2.17, A)`
  - `(hon.de, 192.0.2.18, A)`
- Creates additional records, e.g. `www.hon.de`

Time-To-Live (TTL) Pattern

Context: Certain information is only valid for a given time or should not be distributed unconditionally.

Implementation:

- Add a TTL field to packets or data entries.
- Provide a mechanism to detect expiry:
  - Explicitly decrease TTL after some operation.
  - Use a clock and compare time information.

Variants:

- Hop-Limit (sometimes Hop = Time)

Related:

- Cache Pattern

Attention:

- TTL value is Design Parameter
  - Performance vs. Consistency
  - Actuality vs. Robustness
  - Range of Influence vs. Duration

DNS Caching

Motivation

- Most RR don't change often.
- Resolving takes *time* and *bandwidth*.
  Wasted if information already known.
- DNS lookup when connecting to unknown host.
- Multiple hosts in one network access same names → no need to resolve the complete domain name again.
- cmp. *Cache Pattern*

Solution

- Answers are cached locally.
  Including "intermediate" responses.
- Entry removed after TTL expired.

Problems

- Entries might be out of date.
- Typically DNS changes need up to 24h to propagate.
  Typical TTL default value: 86400

RFC2136 introduces an update/notify mechanism to actively change entries.

Common Server Implementations

- **Berkley Internet Name Domain (BIND)**
  One of the oldest, still maintained DNS servers, developed at the University of California Berkley and now managed by the Internet Systems Consortium.

- **dnsmasq**
  Started as DNS forwarder, now also includes server functionality. Newer software (published 2001), which also provides DHCP server functionality. Used in most SOHO routers / network appliances.

- **unbound**
  Validating, recursive, and caching DNS resolver. Replacing BIND in most open source systems as a default name server.

- ... see Wikipedia for more

BIND `example.com.rr.zone`

```
$ORIGIN example.com.
$TTL 86400
@       IN    SOA    dns1.example.com.    hostmaster.example.com. (
              2017022801 ; serial
              21600      ; refresh after 6 hours
              3600       ; retry after 1 hour
              604800     ; expire after 1 week
              86400 )    ; minimum TTL of 1 day

        IN    NS    dns1.example.com.
        IN    NS    dns2.example.com.


        IN    MX    10    mail.example.com.
        IN    MX    20    mail2.example.com.


dns1    IN    A     10.0.1.1
dns2    IN    A     10.0.1.2


server1 IN    A     10.0.1.5
server2 IN    A     10.0.1.6


ftp     IN    A     10.0.1.3
        IN    A     10.0.1.4

mail    IN    CNAME     server1
mail2   IN    CNAME     server2


www     IN    CNAME     server1
```

dnsmasq `/etc/dnsmasq.conf`

```
auth-server=dns1.example.com,eth0
auth-soa=2017022801,hostmaster.example.com,21600,3600,604800
auth-ttl=86400
auth-zone=example.com,10.0.1.0/24

mx-host=example.com,mail.example.com,10
mx-host=example.com,mail2.example.com,20

host-record=dns1.example.com,10.0.1.1
host-record=dns2.example.com,10.0.1.2

host-record=server1.example.com,10.0.1.5
host-record=server2.example.com,10.0.1.6

host-record=ftp.example.com,10.0.1.3
host-record=ftp.example.com,10.0.1.4

cname=mail.example.com,server1
cname=mail2.example.com,server2

cname=www.example.com,server1
```

Attacking DNS

Distributed Denial of Service (DDoS)

- Bombard root servers with traffic:
  - Not successful yet.
  - Local DNS cache requests so no big deal.
- Bombard TLD or ISP servers:
  - More dangerous, because there are fewer replicated servers.
  - e.g. incident at Dyn in Oct 2016

Redirect

- Man-in-the-middle: intercept and manipulate queries and replies.
- DNS poisoning: send wrong replies to DNS servers that cache them.

Exploit DNS for DDoS

- Send queries with spoofed source IP so that the target gets the traffic.

Defending DNS

Problem: DNS was not designed with security in mind.

Attacks

- Packet Interception:
  - Man-in-the-middle attacks.
  - Eavesdropping.
  - Manipulate answers.
  - Requires access to packets in transit.
- Query Prediction:
  - Relatively easy to guess the parameters of a DNS query (IPs, Ports, IDs).
  - Allows to do the same manipulation as before, without access to transit.
- For more see RFC3833.

Defenses

- DNSSec (RFC2535).
  - Records are signed using a hierarchy of trust.
  - Queries are authenticated.
- Queries cannot be tampered with.
- Responses cannot be tampered with.
- Attacks to feed in inappropriate DNS entries (e.g. a fake IP address for a banking URL) are avoided.
- Remaining issues:
  - No encryption, so eavesdropping is still possible.
  - ...

Further Uses of DNS

A) Load Balancing

Problem:

- Services might be popular and hence consumed by many users.
- Buying expensive servers is an option, but does not scale well.
- Providing the same service using multiple servers is easier.

Solution:

- Use DNS in a way that the same name resolves to different IP addresses.
- When accessing `service.company.com`, provide IP addresses round-robin so that different users are sent to different servers.

B) Content-Delivery Networks

- DNS also plays an important role in this context... see later.

Must I Use DNS?

DNS comes at a **cost** (maintain zones, nameservers, ...), but brings **value** and can effectively reduce management effort.

Guidelines:

- *If you are connected to the Internet...*
  You must! Not necessarily using own servers, but at least join the global DNS system with your hosts.

- *If you have your own multi-site TCP/IP-based network...*
  You probably want to! DNS will most likely simplify structures and make your life easier by assigning names and not using numbers (see later: DHCP).

- *If you have your own local area network or site network...*
  You could well do without a DNS server. Other products (WINS, mDNS) can provide this functionality. But consider evolvability of your network.

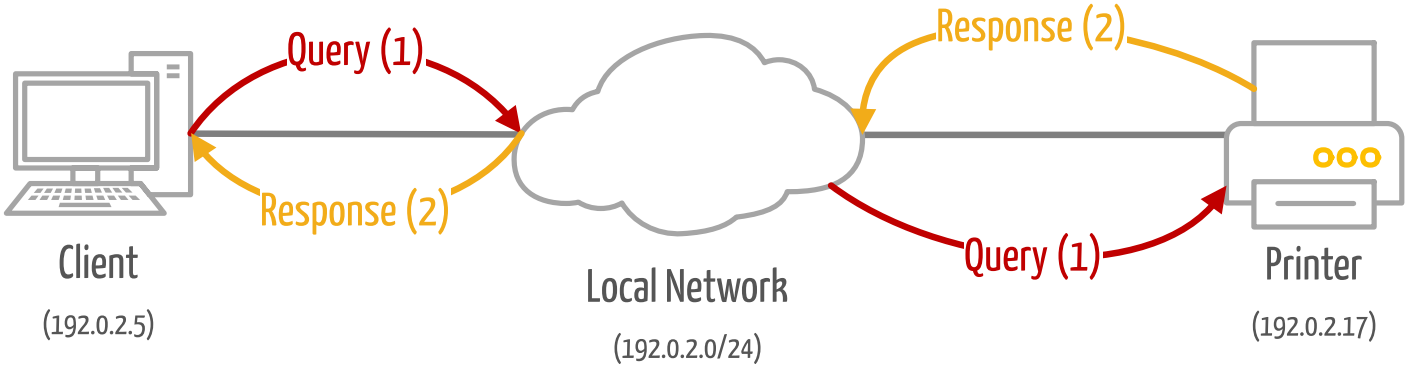
Multicast DNS (mDNS) (RFC6762)

- DNS using **multicast transmission** on the **local link**.
- **No DNS server** - hosts answer queries directly ( ).
- Packet format identical to server-based DNS.
- All hosts on the link share the same domain: `.local`
- Used in Zero-Configuration Networking.

Popular Implementations

- Avahi ( )
- Apple Bonjour (,)

mDNS in Action



Query (1)

```
Src: 192.0.2.5
Dst: 224.0.0.251
Question RRs: 1
Questions:
  printer.local, type A, class IN
```

Response (2)

```
Src: 192.0.2.17
Dst: 224.0.0.251
Answer RRs: 1
Answers:
  printer.local, type A, class IN, addr 192.0.2.17
```

Electronic Mail (Email)

Email | Motivation

First means to provide a **letter-like experience** over **computer networks**.

- **Fast:** Even though propagation takes time, it is way quicker than "snail" mail.
- **Async:** Messages are composed, sent and at some point in the future delivered to the recipient.
- **Organization:** Have a mailbox of letters, sort and archive them.
- **Versatile:** Using attachments, arbitrary files can be sent along.

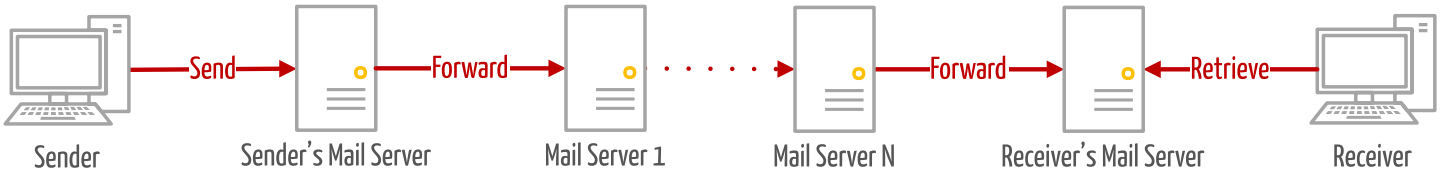
While it is more and more replacing physical mail...

- ... there are still legal issues (esp. in Germany) so that things must be printed.
  - Archiving documents to have accountability.
  - Digitally signed documents are still not allowed for important processes.
    Security doubts, lacking infrastructure, large number of citizens, ...
- ... some people don't like it and prefer printed letters.

Email | Systems Perspective

Electronic Mail uses the following major components:

- User Agents
  User interfaces (graphical, console, ...) or software components which are able to send and receive email messages.
- Mail Servers
  Software components that can store and relay email messages.
- Simple Mail Transfer Protocol (SMTP)
  Well-defined messages and behaviours to transmit mail from sender to receiver.
- Mail Access Protocols
  Well-defined messages and behaviours to allow user agents to retrieve/delete/move messages residing in mail server.



Email | Software Componentss

User Agents

- Purpose:
  - Reading and Searching.
  - Composing and Editing.
  - Structuring and Archiving.
- Examples:
  - Outlook
  - Thunderbird
  - Alpine
  - ...

Mail Servers

- Purpose:
  - Store messages for users (provide mailbox).
  - Relay messages to other servers.
  - Classify messages as spam.
- Examples:
  - Postfix
  - Exim
  - Microsoft Exchange
  - ...

Quiz

Which transport layer protocol is used for mail transfer protocols (e.g. SMTP)?

A: TCP (100% reliable, throughput limited, no time bounds).

B: UDP (unreliable, throughput unlimited, no time bounds)

C: Other Transport Protocol

Answer:

✔ A: True. Full reliability is required and we have no timing bounds.

✖ B: Possible, but never used.

✖ C: Nope.

Bonus Question: What do you think why UDP is not used?

SMTP (RFC5321)

- SMTP := Simple Mail Transfer Protocol
- TCP Port 25 (UDP port 25 also reserved by IANA).
- Directly transfers mails from one server to another.
- Phases:
  - Handshake
  - Transfer
  - Closure
- Encoding:
  - Pure ASCII text.
  - Message itself (header and body) in 7-bit ASCII.
- Commands (Requests)
- Responses: Status Code + Phrase
- Server uses `CRLF.CRLF` to determine message end.
  Lines with just a . have to be escaped.

SMTP | Sample Interaction

```
S: 220 python.co.uk
C: HELO uni-saarland.de
S: 250 Hello uni-saarland.de, pleased to meet you
C: MAIL FROM: <hon@nt.uni-saarland.de>
S: 250 hon@nt.uni-saarland.de... Sender ok
C: RCPT TO: <john@python.co.uk>
S: 250 john@python.co.uk... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: And now for something completely different.
C: A man with a tape recorder up his nose.
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 python.co.uk closing connection
```

`S`: Server, `C`: Client

See also: SMTP Protocol Tutorial.

HON

Email | Message Format

- Messages follow RFC5322 (superseding the original RFC822).
- Header Lines:
  - `To:` Email-address of recipient.
  - `From:` Email-address of sender.
  - `Reply-To:` Email-address to which responses should be sent.
  - `Subject:` Topic for a message.
- Body:
  - Contains the actual message.
  - Only ASCII characters allowed.

`To` and `From` are different than the parameters used for SMTP commands ( ).

Email | Access Protocols

Post Office Protocol (POP)

Authorization and transaction protocol, defined in RFC1939.

- *Download-and-delete:* once deleted mails cannot be downloaded by other clients.
- *Download-and-keep:* copies messages on different clients (no sync).
- Stateless across sessions.

Internet Mail Access Protocol (IMAP)

Advanced protocol (superset of POP) defined in RFC3501.

- Keeps all messages at the server.
- Allows users to organize messages in folders.
- Keep user state across sessions.
  Folder names, messages ids, read/unread state.

HTTP (e.g. Gmail, Outlook.com, etc.)

- Integrates email access and manipulation within a web application.
- Highest degree of technical integration, as users no longer have to bother with mail-server names etc.

Wrap-Up

HON

Questions?

Take-Home Messages

- **Domains names** are **human readable**, convenient **identifiers**.
- The **DNS** system provides a vital structure for the Internet.
- **Domains** are subtrees of this global structure.
- **Zones** are domain names managed by one **administrative entity**.
- **Nameservers** are responsible for **resolving domain names** and **serving a zone**.
- **Caching** makes DNS efficient and scalable.
- **Securing** DNS is still a hot topic, because it is a crucial service.

Further Reading

- Kurose-Ross "Computer Networking"
  - Sec. 2.4 (Email)
  - Sec. 2.5 (DNS)

- Liu & Albitz: DNS and BIND
  - Chapter 1, 2

Copyright and Acknowledgement

- Some examples and parts of the content are taken from the book Computer Networking as well as the slide deck by James Kurose and Keith Ross.
- Some examples and parts are inspired by DNS and BIND 5th Edition, by Cricket Liu and Paul Albitz.
- The **material is copyrighted**. Please treat the slides accordingly and do not share them.