

Einführung  
DNS

Lukas Beeler  
Huber+Monsch  
Interne Schulung

---

1. November 2003  
*Rev : 9*

## Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>3</b>
1.1	Geschichtliches . . . . .	3
<b>2</b>	<b>Aufbau</b>	<b>4</b>
2.1	Root Server . . . . .	4
2.2	Die TLD Server . . . . .	5
2.3	Die 2nd Level Domain Server . . . . .	6
2.4	Delegation durch die 2nd Level DNS Server . . . . .	7
<b>3</b>	<b>Theoretisches</b>	<b>8</b>
3.1	Resource Record Typen . . . . .	8
3.1.1	SOA . . . . .	8
3.1.2	NS . . . . .	8
3.1.3	A . . . . .	8
3.1.4	PTR . . . . .	8
3.1.5	MX . . . . .	9
3.1.6	SRV . . . . .	9
3.2	Noch mehr Resource Records . . . . .	9
3.2.1	CNAME . . . . .	9
3.2.2	AAAA . . . . .	9
3.2.3	TXT . . . . .	10
3.2.4	LOC . . . . .	10
<b>4</b>	<b>Query-Typen</b>	<b>11</b>
4.1	Iterativ . . . . .	11
4.2	Rekursiv . . . . .	11
<b>5</b>	<b>DNS-Server Typen</b>	<b>12</b>
5.1	Resolver . . . . .	12
5.2	Authoritativer Nameserver . . . . .	12
5.3	Die Realität . . . . .	12
<b>6</b>	<b>Glue, oder Leimschnüffeln für Anfänger</b>	<b>13</b>
6.1	Delegation mit Glue . . . . .	13
6.2	Delegation ohne Glue . . . . .	14
6.3	Wieso ist Glueless schlecht? . . . . .	14
6.4	Wie mache ich eine Zone mit Glue? . . . . .	14
6.5	Zusätzliches . . . . .	14
<b>7</b>	<b>Delegationen</b>	<b>15</b>
7.1	. . . . .	15
7.2	ch. . . . .	15
7.3	CNO . . . . .	15
7.4	de. . . . .	15
<b>8</b>	<b>Spezielles</b>	<b>16</b>
8.1	Load Balancing (Webserver) . . . . .	16
8.2	Load Balancing (Mail) . . . . .	16
8.3	TTL . . . . .	16

<b>9</b>	<b>Praktisches</b>	<b>17</b>
9.1	Zonentransfer . . . . .	17
9.1.1	AXFR . . . . .	17
9.1.2	Ablaufschema . . . . .	17
9.1.3	LDAP . . . . .	17
9.1.4	rsync over ssh . . . . .	17
<b>10</b>	<b>Verständnisfragen</b>	<b>18</b>
10.1	Erster Teil . . . . .	18
10.2	Zweiter Teil . . . . .	18
<b>11</b>	<b>Übungen</b>	<b>20</b>
11.1	Kleinfirma . . . . .	20
11.2	Mittlere Firma, mit drei Filialen . . . . .	20
11.3	Switch Nameserver . . . . .	21

# 1 Einführung

DNS, das Domain Name System, sorgt für menschenlesbare Adressierung in IP-basierten Netzwerken. Das DNS ist hierarchisch aufgebaut, und erlaubt deswegen die verteilte Speicherung riesiger Datenmengen. DNS ist ein relativ komplexes Protokoll, wobei die Komplexität durch Kombination vieler simpler Elemente erreicht wird.

## 1.1 Geschichtliches

Damals, als ich und meine Grossmutter noch kleine Mädchen waren, gab es das ARPANET. Und im ARPANET gab es eine Datei, /etc/hosts, (Ja, damals hat man noch überall anständige Betriebssysteme verwendet). Diese Datei wurde mittels FTP zwischen den verschiedenen Rechnern hin und her kopiert, und beinhaltete sämtliche Rechner, die am ARPANET teilnahmen. Natürlich wurde diese Datei von Monat zu Monat grösser, und irgendwann wurde klar, dass dieser Ansatz nicht mehr ausreichen würde.

Also kam die Idee auf, ein verteiltes Datenbanksystem für diese Auflösung zu benutzen. Es sollte hierarchisch sein, um auch International noch flexibel zu bleiben, und um Konflikte zu vermeiden. Wo man gerade sowieso schon dabei war, das Rad neu (und runder ;) zu erfinden, kam man auf die Idee, dieses neue System auch flexibel und erweiterbar zu machen, und diverse Spezialfälle zu regeln.

Man schuf eine Implementation, die halbwegs funktionierte. Diese Implementation hiess BIND. Zu diesem Zeitpunkt existierte noch kein Standard, also fing man an, einen Standard zu verfassen. Man nahm die momentane Funktionsweise von BIND, und definierte diese als Standard. Dies hatte zur Folge, dass DNS einiges an Altlasten mit sich trägt, die sich damals in Bind befanden.

**ACHTUNG:** DNS ist ein enorm flexibles Protokoll. Nachher folgende Beispiele zeigen die gängige Praxis, nicht alles was theoretisch möglich ist.

Das Verfassen dieses Dokumentes wurde von der Firma Huber + Monsch gesponsort.

## 2 Aufbau

Bei DNS wird ein Name von rechts nach links aufgebaut. Der 'Root' (Wurzel, Anfang), ist also die '.' Zone. Dieses Beispiel folgt dem Beispiel des Internets.

Unser Beispiel wird nun folgender Hostname sein:

a.mx.hubermons.ch.

Zu beachten ist der abschliessende Punkt. Dieser ist optional, wird jedoch zu einem besseren Verständnis angefügt. Dieser Name wird nun von rechts nach links aufgelöst. Wer dies macht, wird später erklärt. Wir fragen nach dem 'A' RR (Resource Record). Das 'A' steht für 'Address', also eine IP-Adresse.

### 2.1 Root Server

Als erstes fragt man bei den sogenannten Root-Servern an. Dies ist eine iterative Anfrage, auf welche man immer eine Antwort mit AUTHORITATIVE 1 zurückbekommen muss. Die Root-Server wissen den A RR von a.mx.hubermons.ch. nicht, koennen darauf also nicht autoritativ antworten. Die Root Server wissen jedoch wer für ch. verantwortlich ist. Also antworten sie autoritativ mit den NS (Nameserver) RR's für ch.. Visuell sieht dies so aus:

```
; <<>> DiG 9.2.2 <<>> a.mx.hubermons.ch @a.root-servers.net
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7884
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 7, ADDITIONAL: 7

;; QUESTION SECTION:
;a.mx.hubermons.ch.          IN      A

;; AUTHORITY SECTION:
ch.          172800  IN      NS      NS.APNIC.NET.
ch.          172800  IN      NS      DOMREG.NIC.ch.
ch.          172800  IN      NS      MERAPI.SWITCH.ch.
ch.          172800  IN      NS      TULKU.NIC.AR.
ch.          172800  IN      NS      CCTLD.TIX.ch.

;; ADDITIONAL SECTION:
NS.APNIC.NET.          172800  IN      A      203.37.255.97
DOMREG.NIC.ch.        172800  IN      A      130.59.1.80
MERAPI.SWITCH.ch.     172800  IN      A      130.59.211.10
TULKU.NIC.AR.         172800  IN      A      200.16.97.77
CCTLD.TIX.ch.         172800  IN      A      194.42.48.120

;; Query time: 831 msec
;; SERVER: 198.41.0.4#53(a.root-servers.net)
;; WHEN: Mon Apr 28 11:49:30 2003
;; MSG SIZE rcvd: 334
```

Wie man sieht antwortet der Root-Server mit den NS-RR's für ch., und den A-RR's für diese NS-RR's. Wir wissen nun etwas mehr, und werden uns nun daran machen, diese Server zu befragen.

## 2.2 Die TLD Server

Wieder fragen wir einen der ch. Nameserver nach a.mx.hubermensch.ch. Diese wissen den A RR jedoch auch nicht. Wissen jedoch, wer für hubermensch.ch. zuständig ist. Also Teilen sie uns dessen NS-RR's mit.

```

; <<>> DiG 9.2.2 <<>> a.mx.hubermensch.ch @ns.apnic.net
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22594
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;a.mx.hubermensch.ch.          IN      A

;; AUTHORITY SECTION:
hubermensch.ch.              43200   IN      NS      ns1.v-1.ch.
hubermensch.ch.              43200   IN      NS      ns2.v-1.ch.

;; ADDITIONAL SECTION:
ns1.v-1.ch.                  43200   IN      A       157.161.114.35
ns2.v-1.ch.                  43200   IN      A       157.161.114.36

;; Query time: 1152 msec
;; SERVER: 203.37.255.97#53(ns.apnic.net)
;; WHEN: Mon Apr 28 11:53:44 2003
;; MSG SIZE  rcvd: 109

```

Auch diesmal haben wir NS-RR's aund A-RR's bekommen. Also fragen wir nun bei diesen nach.

### 2.3 Die 2nd Level Domain Server

Wir fragen den hubermensch.ch. Nameserver nach a.mx.hubermensch.ch.. Sagt dieser uns nun die für mx.hubermensch.ch. zuständigen Nameserver?

Nein. Denn mx.hubermensch.ch. wurde nicht weiterdelegiert, also wissen die hubermensch.ch. Nameserver den A RR. Und antworten auch entsprechend.

```
; <<>> DiG 9.2.2 <<>> a.mx.hubermensch.ch @ns1.v-1.ch
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38776
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 4

;; QUESTION SECTION:
;a.mx.hubermensch.ch.          IN      A

;; ANSWER SECTION:
a.mx.hubermensch.ch.         86400   IN      A      194.147.133.161

;; AUTHORITY SECTION:
hubermensch.ch.             43200   IN      NS     ns2.v-1.ch.
hubermensch.ch.             43200   IN      NS     ns1.v-1.ch.

;; ADDITIONAL SECTION:
ns1.v-1.ch.                  43200   IN      A      157.161.114.35
ns2.v-1.ch.                  43200   IN      A      157.161.114.36

;; Query time: 85 msec
;; SERVER: 157.161.114.35#53(ns1.v-1.ch)
;; WHEN: Mon Apr 28 13:02:18 2003
;; MSG SIZE  rcvd: 182
```

## 2.4 Delegation durch die 2nd Level DNS Server

Der 2nd Level DNS Server muss nicht unbedingt die Antwort genau wissen, es ist genauso möglich, das `mx.hubermensch.ch` nochmal woanders hin delegiert ist. Dies ist der Punkt, an dem DNS beginnt ziemlich theoretisch und Abstrakt zu werden. Theoretisch hätten bereits die Root-Server unsere Anfrage beantworten können. Doch dazu später mehr.

Gehen wir zuerst einmal davon aus, das die `hubermensch.ch` Nameserver den A-RR für `a.mx.hubermensch.ch` nicht kennen, wohl aber die für `mx.hubermensch.ch` zuständigen Nameserver. Die Antwort dürfte dann ungefähr so aussehen:

```

; <<>> DiG 9.2.2 <<>> a.mx.hubermensch.ch @ns.apnic.net
;; global options:  printcmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 22594
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;a.mx.hubermensch.ch.                IN      A

;; AUTHORITY SECTION:
mx.hubermensch.ch.                  43200   IN      NS      a.ns.mx.hubermensch.ch
mx.hubermensch.ch.                  43200   IN      NS      b.ns.mx.hubermensch.ch

;; ADDITIONAL SECTION:
a.ns.mx.hubermensch.ch.             43200   IN      A       157.24.23.12
b.ns.mx.hubermensch.ch.             43200   IN      A       157.123.232.123

;; Query time: 1152 msec
;; SERVER: 203.37.255.97#53(ns.apnic.net)
;; WHEN: Mon Apr 28 11:53:44 2003
;; MSG SIZE  rcvd: 109

```

Nun wissen wir, wer für `mx.hubermensch.ch` verantwortlich ist, und könnten nun diese Server nach `a.mx.hubermensch.ch` befragen. Dies wäre das selbe Spielchen wie oben, deswegen lassen wir hier weg.

### 3 Theoretisches

#### 3.1 Resource Record Typen

##### 3.1.1 SOA

SOA bedeutet Start of Authority. Also der Punkt einer Zone, ab welchem ein autoritativer DNS Server für eine Zone zuständig ist. Ein SOA ist folgendermassen aufgebaut:

```
hubermensch.ch. 86400 IN SOA ns1.v-1.ch. ; Prim"arer Nameserver
                        webmaster.v-1.ch. ; Adresse des Hostmasters
                                                (erster Punkt wird zu @)
                        2002022716 ; Serial
                        10800 ; refresh
                        3600 ; retry
                        604800 ; expire
                        86400 ; minimum TTL
```

##### 3.1.2 NS

NS bedeutet Nameserver. Er stellt dar, welche Nameserver für eine Zone zuständig sind. Eine Zone kann, wie ihr oben sicher schon gesehen habt, mehrere NS RR's haben. Ein NS ist folgendermassen aufgebaut:

```
hubermensch.ch. 86400 IN NS ns1.v-1.ch. ; Name eines Nameservers
```

##### 3.1.3 A

A bedeutet Address. Er weist einem Namen, eine IPv4-Adresse zu. Dies ist wohl die Funktion, die man meistens benützt. Der Web-Browser benutzt z.B. den A RR, um herauszufinden, welche Adresse sein Webserver hat. Ein A ist folgendermassen aufgebaut:

```
www.hubermensch.ch. 86400 IN A 157.161.114.120 ; IPv4 Adresse
```

##### 3.1.4 PTR

PTR bedeutet Pointer. Er ist das direkte Gegenstück zu einem A RR. Er wird benutzt, um von einer IP-Adresse auf einen Namen zu zeigen. Ein PTR ist folgendermassen aufgebaut:

```
113.135.144.213.in-addr.arpa.86400 IN PTR mana.projectdream.org. ; Name der IP
```

Achtung: Um von einer IP Adresse auf einen Namen aufzulösen, wird ein spezielles Verfahren angewandt, um eine IP Adresse in einen Namen umzuwandeln. Dies geschieht folgendermassen:

```
213.144.135.113 ; Umkehrung der Reihenfolge
113.135.144.213 ; Anhängen von in-addr.arpa.
113.135.144.213.in-addr.arpa.
```

113.135.144.213.in-addr.arpa Ist nun wieder ein ganz regulärer DNS Namen, der nach dem gleichen Prinzip wie a.mx.hubermensch.ch. aufgelöst wird.

### 3.1.5 MX

MX bedeutet Mail Exchanger. Er ist, wie der Name schon sagt, für den Austausch von Mails verantwortlich. Normalerweise treten MX RR's immer zu mehr auf, nur selten sind sie alleine.

```

; Priorität
projectdream.org. 86400 IN MX 0 a.mx.projectdream.org. ; Name des MX's
projectdream.org. 86400 IN MX 5 b.mx.projectdream.org.
projectdream.org. 86400 IN MX 10 c.mx.projectdream.org.

```

Die höchste Priorität hat der Record mit der niedrigsten Zahl, in diesem Falle wäre das `a.mx.projectdream.org.`. Ein Mailserver wird also versuchen seine Mail für `projectdream.org.` zuerst bei `a.mx.projectdream.org.` abzuliefern. (Frage: Wie findet der Mailserver die IP Adresse von `a.mx.projectdream.org.` heraus?). Schlägt dies fehl, so versucht er es bei `b.mx.projectdream.org.` etc.

### 3.1.6 SRV

SRV steht für Services. In diesem RR werden Services beschrieben, die unter einer bestimmten Domain angeboten wurden. Microsoft nutzt den SRV RR für ihr Active Directory. Mittels DNS und SRV RR's werden z.B. die Rechner bekanntgegeben welche die Kerberos Authentication, und auch den LDAP Part von Active-Directory bereitstellen.

```

; Dienst ; Protokoll ; Hierarchie ;Balance ;Port ; Maschine
_ldap._tcp.pdc._msdcs.hubermensch.ch. 600 IN SRV 0 100 389 server

```

In einem SRV RR werden sowohl Art der Datenübertragung (tcp), Dienst (ldap), Port (389), als auch die Maschine die für diesen Dienst zuständig ist festgehalten. Ebenfalls existiert ein Distribution-Feld, mit zwei Werten. Dieses dient der Lastverteilung, falls für diesen Service mehrere SRV RR's existieren. Das komplette Verständnis des SRV RR's ist für einen versierten Windows-Administrator unumgänglich. Ich habe jedoch keine praktische Erfahrungen mit dem SRV RR, aus diesem Grund möchte ich an dieser Stelle auf RFC2782 verweisen.

## 3.2 Noch mehr Resource Records

### 3.2.1 CNAME

Ein CNAME ist ein sehr spezieller, und sehr gefährlicher Recordtyp. Mit diesem RR kann man sich nicht nur den Prima in den eigenen Fuss schiessen, sondern auch gleich das ganze Bein wegballern.

Ein CNAME RR muss immer alleine sein. D.h. es dürfen keine anderen RR existieren. Aus diesem Grund ist es z.B. nicht möglich für `hubermensch.ch.` einen CNAME RR zu definieren (es gibt schon NS und SOA).

Was macht denn nun ein CNAME genau? Er 'linkt' auf einen anderen Namen. Wenn ich für `www.hubermensch.ch.` einen CNAME auf `www.grossenbacher.ch.` mache, wird der Resolver versuchen `www.grossenbacher.ch.` aufzulösen, und dies dem Client als Antwort zurückgeben.

```

www.hubermensch.ch. 86400 IN CNAME www.grossenbacher.ch. ; Link-Ziel

```

### 3.2.2 AAAA

Mal wieder etwas simples: ein AAAA RR enthaelt eine IPv6 Adresse, und ist ansonsten gleich wie ein A RR

```

may.projectdream.org. 86400 IN AAAA 3ffe:202c:ffff:32:230:4fff:fe08:358d
; IPv6 Adresse

```

### 3.2.3 TXT

TXT ist ein Text Record, und enthaelt menschenlesbaren Text, d.h. Infos. Wird meistens für Scherze unter Techies gebraucht. Aber auch für Blacklists, welche DNS als DB Backend verwenden sind sie sehr nützlich.

```
may.projectdream.org. 86400 IN TXT see http://projectdream.org/nfo/may.html"  
; Freeform Text
```

### 3.2.4 LOC

LOC steht für Location, und zeigt ein wo sich ein Host physikalisch befindet (in Längen und Breitengraden).

```
projectdream.org. 86400 IN LOC 47 0 0.000 N 9 0 0.000 E 779.00m 5m 2m 2m  
; GPS-Koordinaten
```

## 4 Query-Typen

### 4.1 Iterativ

Eine Iterative Anfrage wird lediglich von DNS-Debugging Tools, oder von resolvenden DNS-Servern (hierzu später mehr) gestellt. Auf eine Iterative Anfrage muss immer eine Authoritative Antwort kommen, das heisst ein DNS Server muss für die jeweilige Zone zuständig sein.

Das heisst, eine iterative Anfrage nach `a.mx.hubermensch.ch.` bei den Root-Nameservers (Die als SOA `.'` haben), könnte bereits mit einem A-RR beantwortet werden. Dies ist jedoch üblicherweise nicht der Fall.

Iterative Anfragen können auf 6 Arten beantwortet werden:

- Garnicht ('Ich weiss es nicht')
- Mit den Adressen der Root Server ('Ich weiss es nicht')
- Mit einem NXDOMAIN (Diese Domain existiert nicht)
- Mit einem SERVFAIL (Fehlkonfiguration!)
- Mit NS RR's (Delegation)
- Mit den angefragten RR's (Endgültige Antwort)

### 4.2 Rekursiv

Eine rekursiv Anfrage wird von einem DNS-Client gestellt. Dieser erwartet nun einfach eine Antwort, egal ob autoritativ oder nicht. Diese Anfragen werden üblicherweise von resolvenden DNS-Servern entgegengenommen. Rekursive Anfragen können auf 3 Arten beantwortet werden:

- Garnicht ('Ich mach das das nicht')
- Mit einem SERVFAIL (Fehlkonfiguration!)
- Mit den angefragten RR's (Endgültige Antwort)

## 5 DNS-Server Typen

So weit, so verwirrend. Es gibt nämlich mehrere Typen DNS Server: Resolver und Authoritive.

### 5.1 Resolver

Ein Resolver löst Namen für DNS-Clients auf. Dies läuft so ab:

- DNS-Client zu Resolver: A RR von a.mx.hubermensch.ch? (rekursiv)
  - Resolver zu Root: A RR von a.mx.hubermensch.ch.? (iterativ)
  - Root zu Resolver: NS RR von ch.! (authoritiv)
  - Resolver zu Switch: A RR von a.mx.hubermensch.ch.? (iterativ)
  - Switch zu Resolver: NS RR von hubermensch.ch.! (authoritiv)
  - Resolver zu Vision 1: A RR von a.mx.hubermensch.ch.? (iterativ)
  - Vision 1 zu Resolver: A RR von a.mx.hubermensch.ch.! (authoritiv)
- Resolver zu DNS-Client: A RR von a.mx.hubermensch.ch.! (non-authoritiv)

Ein Resolver steht z.B. bei einem Provider, damit seine Kunden Namen wie `www.brigitte.com.` auflösen können.

### 5.2 Authoritiver Nameserver

Authoritive Nameserver sind z.B. solche von Switch, die Root-Server, und natürlich auch solche von Webhostern. Ablaufen tut das so:

- Resolver zu Vision 1: A RR von a.mx.hubermensch.ch.? (iterativ)
- Vision 1 zu Resolver: A RR von a.mx.hubermensch.ch.! (authoritiv)

oder so:

- Resolver zu Switch: A RR von a.mx.hubermensch.ch.? (iterativ)
- Switch zu Resolver: NS RR von hubermensch.ch.! (authoritiv)

### 5.3 Die Realität

Die meiste Software, welche DNS implementiert, unterscheidet nicht zwischen Authoritiven Nameservern und Resolvem. Als Beispiel zu dieser Kategorie gelten der M\$ DNS Server, und BIND. Hier sind beide Funktionalitäten in die selbe Software integriert, was die Administration ebendieser komplex und undurchsichtig macht. Andere DNS Software, wie z.B. die djbdns Suite trennt zwischen authoritive nameserver (tinydns) und resolver (dnscache) auf.

## 6 Glue, oder Leimschnüffeln für Anfänger

Das ist der Moment, wo ihr schreien dürft: 'Noch komplizierter?'

Jap, jetzt wirds nämlich richtig eklig, und wir kommen zu einem Thema das selbst DNS-Admins ins grübeln bringen kann. Am besten kann man dieses Thema mit Beispielen erklären:

### 6.1 Delegation mit Glue

```

; <<>> DiG 9.2.2 <<>> hubermensch.ch @NS.APNIC.NET.
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47051
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
hubermensch.ch.                IN      A

;; AUTHORITY SECTION:
hubermensch.ch.                43200   IN      NS      ns1.v-1.ch.
hubermensch.ch.                43200   IN      NS      ns2.v-1.ch.

;; ADDITIONAL SECTION:
ns1.v-1.ch.                    43200   IN      A       157.161.114.35
ns2.v-1.ch.                    43200   IN      A       157.161.114.36

;; Query time: 432 msec
;; SERVER: 203.37.255.97#53(NS.APNIC.NET.)
;; WHEN: Tue Apr 29 17:39:47 2003
;; MSG SIZE rcvd: 104

```

Dieses Query dürfte jedem von euch bekannt vorkommen. Wie ihr seht, liefert uns Switch die IP Adressen der Vision One Nameserver gleich mit. Um also mehr über hubermensch.ch müssen wir lediglich eine Anfrage an diese IP Adressen schicken. Dies ist eine Delegation mit Glue und damit gut.

Frage: Wieso darf Switch uns die A RR's für ns1.v-1.ch. mitteilen?

## 6.2 Delegation ohne Glue

```

; <<>> DiG 9.2.2 <<>> aol.ch @NS.APNIC.NET.
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55301
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 0

;; QUESTION SECTION:
;aol.ch.                IN      A

;; AUTHORITY SECTION:
aol.ch.                43200  IN      NS      ns2.ascio.net.
aol.ch.                43200  IN      NS      ns1.ascio.net.

;; Query time: 417 msec
;; SERVER: 203.37.255.97#53(NS.APNIC.NET.)
;; WHEN: Tue Apr 29 17:44:49 2003
;; MSG SIZE  rcvd: 69

```

Dies ist eine eine Delegation ohne Glue (glueless), und das ist nicht gut. Was fehlt in dieser Antwort? Genau, die Additional Section. Wieso? Weil die `ch.` Nameserver einen SOA von `ch.` haben, und damit keine Antworten für `irgendwas.net.` herausgeben dürfen. Wo ist nun das Problem mit einer Delegation ohne Glue? Um mehr über `aol.ch.` müssen wir erst die A RR's von `ns2.ascio.net` herausfinden. Und dafür darf man erstmal die Root-Server, und dann die `net.` TLD Server anfragen. Sprich, eine Glueless Delegation braucht einiges mehr an Zeit. Frage: Was passiert, wenn die NS RR's für `ascio.net.` auf `ns1.aol.ch` zeigen?

## 6.3 Wieso ist Glueless schlecht?

- Langsam
- Loopbildung möglich

## 6.4 Wie mache ich eine Zone mit Glue?

Das ist relativ simpel: Die NS Records müssen einfach unterhalb der selben TLD wie die Zone liegen. d.h. für `hubermonsch.ch.` `irgendwas.ch.`, für `cnn.com.` `irgendwas.com.`

## 6.5 Zusätzliches

Beachten sie, das alle RR's, die auf einen Namen zeigen (z.B. MX) glueless sein können. Auch hier sollte darauf geachtet werden, das solche Namen immer mit glue gemacht werden.

## 7 Delegationen

Achtung: Dieser Abschnitt ist nicht theoretisch

### 7.1 .

Ein normaler Resolver fragt ja stets zuerst bei den Root-Servern an. Wenn man also eine neue Top-Level Domain (wie z.B. `com.` oder `ch.`) will, muss man dies in den Root-Servern eintragen. Die Root-Server werden von der ICANN verwaltet. Ohne ganz viel Papierkram ist es also unmöglich seine eigene TLD zu bekommen.

### 7.2 ch.

Eine 2nd Level Domain, wie z.B. `hubermonsch.ch.` ist dagegen schon einiges leichter zu bekommen. Man muss sich lediglich bei `nic.ch` eine Domain registrieren, angeben welche Nameserver zuständig sind, und schon trägt Switch die nötigen Angaben auf ihren Nameservern ein. Natürlich will die Switch dafür Geld sehen.

### 7.3 CNO

CNO steht für `.com` `.net` `.org`. Auch für diese TLD's kann man leicht eine Delegation bekommen, indem man sich an `networksolutions.com` wendet.

### 7.4 de.

Um eine Delegation unter `de.` zu bekommen, muss man entweder in Deutschland wohnen oder genug Geld haben (wie immer). Hier wendet man sich an einen Registrar der `denic.de`, welcher die Domain für Sie bei der `denic` registriert.

## 8 Spezielles

### 8.1 Load Balancing (Webserver)

Load Balancing, z.B. für Webserver lassen sich sehr leicht durch das erstellen von mehreren A RR's erreichen. Der Webbrowser nimmt dann zufällig eine der beiden Adressen.

```
hubermensch.ch.      86400   IN      A       157.161.114.120
hubermensch.ch.      86400   IN      A       157.161.114.140
```

### 8.2 Load Balancing (Mail)

Load Balancing für Email ist ebenfalls recht simpel, es kann über mehrere MX RR's mit der selben Priorität realisiert werden.

```
hotmail.com.        3600    IN      MX      5 mx1.hotmail.com.
hotmail.com.        3600    IN      MX      5 mx2.hotmail.com.
hotmail.com.        3600    IN      MX      5 mx3.hotmail.com.
hotmail.com.        3600    IN      MX      5 mx4.hotmail.com.
```

### 8.3 TTL

TTL steht für Time to Live. Es gibt mehrere TTL's, die wichtigste dürfte wohl die eines einzelnen RR's sein:

```
                ; TTL dieses RR's
hubermensch.ch.  86400   IN      A       157.161.114.120
```

Diese TTL gibt an, wie lange der RR in Sekunden gecacht werden darf. Dieses Caching geschieht durch Resolver. In diesem Falle darf der Resolver das Ergebnis seiner Anfrage einen Tag lang cachen, bevor er erneut Nachfragen muss. Wird diese Zone nun geändert, kann es schlimmstenfalls einen Tag dauern, bis Clients von Resolver xy wieder aktuelle Informationen haben. Doch ein herabsetzen der TTL erhöht den Traffic um einiges, weswegen hier meist ein Kompromiss geschlossen wird. Die sogenannten DynDNS Dienste nutzen genau dies aus. Die TTL eines A RR's wird extrem herabgesetzt (60s), und schon hat man DynDNS. Es existieren jedoch noch mehr TTL's, diese sind jedoch alle im SOA angegeben, und nur für die AXFR Transfer-Methode nötig (mit einer Ausnahme):

```
hubermensch.ch.  86400 IN SOA  ns1.v-1.ch.      ; Prim"arer Nameserver
                webmaster.v-1.ch. ; Adresse des Hostmasters
                2002022716 ; Serial
                10800     ; refresh
                3600      ; retry
                604800    ; expire
                86400     ; minimum TTL
```

- Refresh: Gibt an, wie oft der Master gepollt werden soll
- Retry: Gibt an, wie oft ein Refresh erneut versucht werden soll, falls der Master unerreichbar war
- Expire: Gibt an, nach wieviel Zeit der Slave aufhört auf Anfragen zu Antworten, wenn er den Master nicht mehr erreichen konnte
- Minimum TTL: Gibt die minimale TTL in einer Zone an. Die minimum TTL gilt für NXDOMAIN Antworten, die ebenfalls gecached werden.

## 9 Praktisches

### 9.1 Zonentransfer

Natürlich müssen die Zonen, so denn mehrere authoritative Server für eine Domain zuständig sind, unter den Server synchronisiert werden. Für diese Synchronisation gibt es mehrere Methoden

#### 9.1.1 AXFR

AXFR ist die klassische Zonen-Transfer Methode. Sie wurde mit dem ersten DNS-Server (BIND) in Betrieb genommen, und erfreut sich unter BIND Benutzern immer noch grosser Beliebtheit, obwohl sie eigentlich die schlechteste aller zur Verfügung stehenden Methoden ist. AXFR wurde um ein paar Mechanismen erweitert, die seine Leistungsfähigkeit etwas steigern. Dies sind IXFR (Incremental), der die zu übertragende Datenmenge reduziert, und Notify, der die Zeit zwischen Zonen-Änderung und dem Zonentransfer enorm verkürzt.

#### 9.1.2 Ablaufschema

- Zone auf Master wird geändert, Serial wird angepasst
- Slaves pollen beim Master periodisch den SOA (Serial-Änderung)  
Erweiterung: Notify, Master schickt Nachricht an alle Slaves, welche daraufhin die Zone mittels AXFR/IXFR holen
- Slaves laden die Zone mittels AXFR vom Master herunter  
Erweiterung: IXFR, es werden nur Änderungen übertragen
- Fertig, die Zone ist synchronisiert

#### 9.1.3 LDAP

LDAP ist eine etwas neuere Methode, um Zonendaten zu speichern. Sie wird z.B. vom M\$ DNS Server benutzt, wenn die Zone im sogenannten 'Active-Directory integriert' Modus ist. LDAP hat den Vorteil, das eine Änderung der Zone, und deren Replikation leicht über LDAP geschehen kann. Dies schafft dann auch das Prinzip von Master und Slave ab, da LDAP Referrals kennt.

#### 9.1.4 rsync over ssh

Eine weitere generische Methode zur Datenreplikation unter \*nix-Systemen ist die Verwendung von rsync über einen SSH Tunnel. Diese Methode wird z.B. von tinydns favorisiert, und ermöglicht eine hohe Geschwindigkeit beim Abgleich der Zone. Änderungen sind jedoch genauso wie bei AXFR nur auf dem Haupt-Server möglich. Im Gegensatz zu AXFR arbeitet rsync over ssh jedoch nach dem push-Verfahren, und ermöglicht so eine zeitnahe Synchronisation der Zone. rsync over ssh hat gegenüber LDAP den Nachteil, das nur an einem Ort Änderungen möglich sind. Meistens stellt dies jedoch keine Einschränkung dar. rsync over ssh ist einiges simpler als ein authoritiver DNS-Server mit LDAP Anbindung, und aus diesem Grund auch im allgemeinen sicherer implementiert.

## 10 Verständnisfragen

### 10.1 Erster Teil

1. Erklären sie saemtliche Bestandteile von `mana.sg.projectdream.org`.
2. Erklären sie den Ablauf der Auflösung des Namens `www.hubermensch.ch`.
3. Was ist eine iterative Anfrage?
4. Was ist eine authoritative Anfrage?
5. Was ist eine authoritative Antwort?
6. Was ist ein autoritiver Nameserver?
7. Was ist ein Resolver?
8. Was ist eine rekursive Antwort?
9. Was ist eine rekursive Anfrage?
10. Was ist ein Root-Server?
11. Was ist der SOA eines Root-Servers?
12. Was ist der SOA eines Switch Nameservers?
13. Darf ein Root Server einen A RR für `www.goatse.cx` herausgeben?
14. Darf ein Switch Server einen A RR für `www.cnn.com` herausgeben?
15. Womit kann ein Resolver auf eine Iterative Anfrage beantworten?
16. Womit kann ein Autoritiver Nameserver eine Iterative Anfrage antworten?
17. Trennt M\$ DNS Server Resolver und Autoritiven Nameserver?

### 10.2 Zweiter Teil

1. Was versteht man unter Glue?
2. Wieso ist eine Glueless Delegation schlecht?
3. Welche Gefahren gehen von einem CNAME RR aus?
4. Wann ist ein CNAME RR Invalid?
5. Wann kriege ich einen SERVFAIL zurück?
6. Wann kriege ich einen NXDOMAIN zurück?
7. Mit wem redet ein Resolver?
8. Mit wem redet ein autoritiver Nameserver?
9. Mit wem redet ein Client?
10. Wieviele NS RR's braucht eine Domain?

11. Was ist ein AAAA RR?
12. Wofür braucht es den SOA RR?
13. Kann ich durch umsetzen des SOA RR's zu einem Root-Server werden?
14. Wie kriege ich eine Delegation für eine de. Domain?

## 11 Übungen

Folgende Übungen sind auf einem Blatt Papier auszuführen (Alternativ auch direkt ein Zonefile schreiben, oder in nem Texteditor (Das schliesst Word aus) auf dem Laptop). Sie dienen der Verinnerlichung der Theorie.

### 11.1 Kleinfirma

Erzeugen sie eine Zone für eine Kleinfirma, mit folgenden Randdaten:

- Webservers IP: 1.2.3.4
- Nameserver 1: 1.2.3.2
- Nameserver 2: 1.2.2.1
- Mail-Exchanger: 1.2.3.4
- Domain: firma.ch

Beschreiben sie ebenfalls, wer für eine entsprechende Delegation kontaktiert werden muss.

### 11.2 Mittlere Firma, mit drei Filialen

Erzeugen sie die Zonen für eine mittlere Firma, mit drei Filialen

- Hauptsitz St. Gallen
  - Webservers IP: 1.2.3.4
  - Nameserver 1: 1.2.3.2
  - Nameserver 2: 1.2.2.1
  - Zwei Load-Balancing MX's: 1.2.2.2 und 1.2.2.3
  - Zwei Secondary MX's: 1.5.1.1 und 1.6.1.1
  - Domain: firma.ch und sg.firma.ch
  - Delegiert zh.firma.ch und wil.firma.ch
- Filiale Zürich
  - Nameserver: 2.1.1.1
  - Webservers IP: 2.1.1.1
  - Domain: zh.firma.ch
  - Als Mailserver werden jene des Hauptsitzes verwendet
- Filiale Wil
  - Nameserver: 3.1.1.1
  - Webservers IP: 3.1.1.1
  - Domain: wil.firma.ch
  - Als Mailserver wird ein eigener (3.1.1.2) als primärer, und die des Hauptsitzes als Redundanz verwendet.

### **11.3 Switch Nameserver**

Erzeugen sie eine Zone, wie sie auf einem TLD Server, wie jener der Switch gebraucht werden koennte. Die Daten hierfür dürfen sie sich ausdenken.