

# Übung 11

## Tutorübung zu Grundlagen: Rechnernetze und Verteilte Systeme (Gruppen MI-T7 / DO-T5 SS 2015)

Michael Schwarz

Technische Universität München  
Fakultät für Informatik

08.07.2015 / 09.07.2015

## Domain Name System (DNS) [2, 4, 6, 7]

### Motivation:

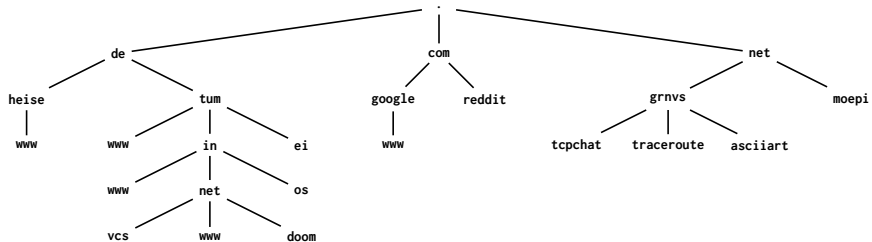
- ▶ Möchte ein Nutzer (Mensch) einen Computer adressieren, z. B. beim Aufruf einer Webseite, will er sich gewöhnlich nicht dessen IP-Adresse merken müssen.
- ▶ Stattdessen adressiert man das Ziel üblicherweise mittels eines hierarchisch aufgebauten Namens, z. B. [www.google.com](http://www.google.com).

Das **Domain Name System (DNS)** besteht aus drei wesentlichen Komponenten:

1. Der **Domain Namespace** ist
  - ▶ ein hierarchisch aufgebauter Namensraum mit
  - ▶ baumartiger Struktur.
2. **Nameservers** speichern
  - ▶ Informationen über den Namensraum,
  - ▶ wobei jeder Server nur kleine Ausschnitte des Namensraums kennt.
3. **Resolver** sind Programme,
  - ▶ die durch Anfragen an Nameserver Informationen aus dem Namespace extrahieren und
  - ▶ anfragenden Clients bzw. Anwendungen zur Verfügung stellen.

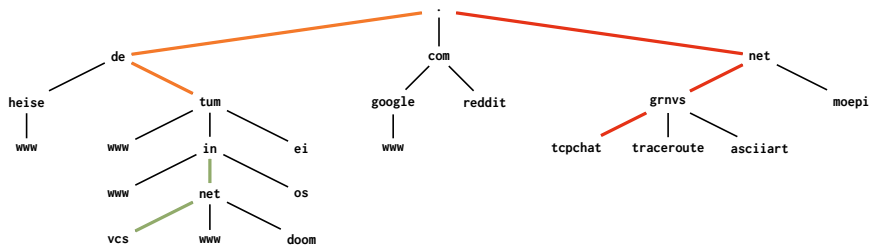
## Domain Namespace

Ein kleiner Auszug aus dem Namespace:



## Domain Namespace

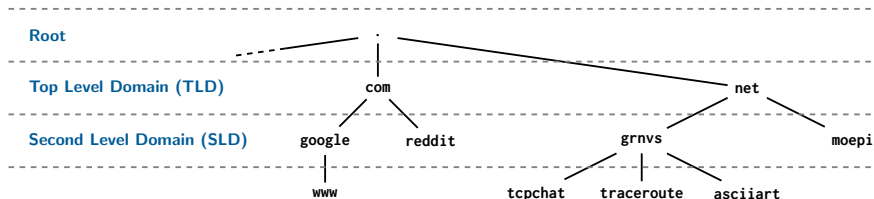
Ein kleiner Auszug aus dem Namespace:



- ▶ Ein **Label** ist ein beliebiger Knoten im Namespace.
- ▶ Ein **Domain Name** ist eine Sequenz von Labels:
  - ▶ Ein **Fully Qualified Domain Name (FQDN)** besteht aus der vollständigen Sequenz von Labels ausgehend von einem Knoten bis zur Wurzel und endet mit einem Punkt, z. B. **tum.de.** oder **tcpchat.grnvs.net..**
  - ▶ Endet er nicht mit einem Punkt, handelt es sich zwar ebenfalls um einen Domain Name, allerdings ist dessen Angabe relativ ausgehend von einem anderen Knoten als der Wurzel, z. B. **vcs.net.in.**
  - ▶ Ein FQDN kann als **Suffix** für einen nicht-qualifizierten Namen verwendet werden, z. B. ergibt **vcs.net.in** zusammen mit dem FQDN **tum.de.** einen neuen FQDN **vcs.net.in.tum.de.**
  - ▶ Ob ein FQDN existiert (z. B. in eine Adresse aufgelöst werden kann), bleibt zunächst offen.

## Übliche Bezeichnungen

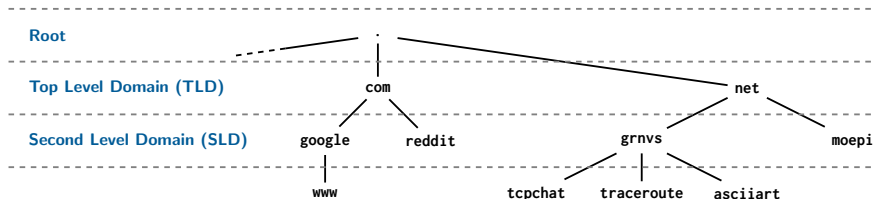
Für die ersten drei Hierarchieebenen im Name Space eigene Bezeichnungen üblich:



Darunter liegende Ebenen werden gelegentlich als **Subdomain** bezeichnet.

## Übliche Bezeichnungen

Für die ersten drei Hierarchieebenen im Name Space eigene Bezeichnungen üblich:



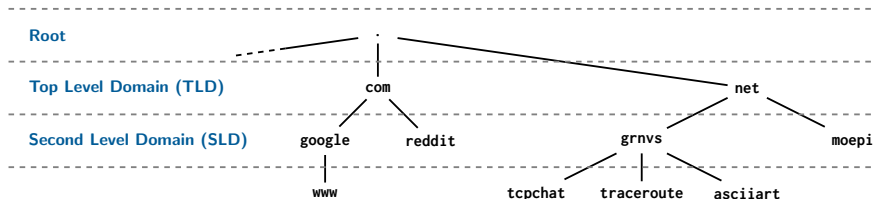
Darunter liegende Ebenen werden gelegentlich als **Subdomain** bezeichnet.

## Vergabe von TLDs und SLDs

- ▶ Top Level Domains werden von der **Internet Corporation for Assigned Names and Numbers (ICANN)** vergeben.
- ▶ Second Level Domains werden von verschiedenen Registraren vergeben.

## Übliche Bezeichnungen

Für die ersten drei Hierarchieebenen im Name Space eigene Bezeichnungen üblich:



Darunter liegende Ebenen werden gelegentlich als [Subdomain](#) bezeichnet.

## Vergabe von TLDs und SLDs

- ▶ Top Level Domains werden von der [Internet Corporation for Assigned Names and Numbers \(ICANN\)](#) vergeben.
- ▶ Second Level Domains werden von verschiedenen Registraren vergeben.

## Zeichensatz

- ▶ Erlaubt sind nur Buchstaben (A–Z) und Zahlen sowie -, wobei letzterer nicht das erste oder letzte Zeichen sein darf.<sup>1</sup>
- ▶ Zwischen Groß- und Kleinschreibung wird nicht unterschieden.

<sup>1</sup> Technisch gesehen könnte ein Eintrag im DNS alle möglichen Oktette enthalten.

## Nameserver

Der Namespace wird

- ▶ in Form einer verteilten Datenbank
- ▶ von ein großen Anzahl von Servern gespeichert,
- ▶ wobei jeder Server nur einen kleinen Teil des gesamten Namespaces kennt.

Zu diesem Zweck ist der Namespace in **Zonen** unterteilt:

- ▶ Zonen sind **zusammenhängende Teilbäume** des Namespaces.
- ▶ Eine Zone kann daher mehrere Ebenen des Namespaces umfassen, aber keine Teilbäume ohne gemeinsame Wurzel.
- ▶ Nameserver bezeichnet man als **autoritativ** für die jeweiligen Zonen, die sie speichern.
- ▶ Dieselbe Zone kann auf mehreren Nameserver gespeichert sein.
- ▶ DNS sieht Mechanismen zum Transfer von Zonen zwischen autoritativen Nameservern vor.
- ▶ Dabei gibt es einen primären Nameserver, auf dem Änderungen an einer Zone vorgenommen werden können, sowie beliebig viele sekundäre Nameserver, welche lediglich über Kopien der Zone verfügen.

Nameserver erwarten eingehende Verbindungen auf UDP/TCP 53:

- ▶ Anfragen werden meist an UDP 53 gestellt.
- ▶ Anfragen, die Größer als 512 B sind, werden an TCP 53 gestellt.
- ▶ Zone Transfers finden immer über TCP 53 statt.



## Resource Records

Die Informationen, die in einer Zone gespeichert sind, bezeichnet man als **Resource Records**:

- ▶ **SOA Record (Start of Authority)** ist ein spezieller Record, der die Wurzel der Zone angibt, für die ein Nameserver autoritativ ist.
- ▶ **NS Records** geben den FQDN eines Nameservers an. Dieser kann auch auf FQDNs in anderen Zonen verweisen.
- ▶ **A Records** assoziieren einen FQDN mit einer IPv4-Adresse.
- ▶ **AAAA Records** assoziieren einen FQDN mit einer IPv6-Adresse.
- ▶ **CNAME Records** sind Aliase, d. h. ein FQDN verweist auf einen "Canonical Name", der selbst wiederum ein FQDN ist.
- ▶ **MX Records** geben den FQDN eines Mailservers für eine bestimmte Domain an, welcher sich nicht notwendigerweise in derselben Zone befinden muss.
- ▶ **TXT Records** assoziieren einen FQDN mit einem String (Text). Wird für unterschiedliche Zwecke verwendet und missbraucht.
- ▶ **PTR Records** assoziieren eine IPv4- oder IPv6-Adresse mit einem FQDN (Gegenstück zu A bzw. AAAA Records).

### Hinweise:

- ▶ Mehrere A oder AAAA Records (auch unterschiedlicher Zonen) können mit derselben IP-Adresse assoziiert sein.
- ▶ Für einen FQDN kann es maximal einen CNAME geben. Wenn ein CNAME existiert, handelt es sich um einen Alias, weswegen es keine weiteren Resource Records für den betreffenden FQDN mehr geben darf.
- ▶ Für eine Zone bzw. Domain gibt es üblicherweise mehrere NS bzw. MX Records.

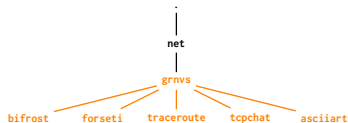
Die Resource Records einer Zone werden auf Nameservern in Form von **Zone Files** gespeichert:

```
$TTL 86400 ; 1 day
grnvs.net. IN SOA  bifrost.grnvs.net. hostmaster.grnvs.net. (
                    164160 ; serial
                    1800  ; refresh (30 minutes)
                    300   ; retry (5 minutes)
                    604800 ; expire (1 week)
                    1800  ; nxdomain (30 minutes)
                )
NS      bifrost.grnvs.net.
NS      forseti.grnvs.net.
A       129.187.145.241

$ORIGIN grnvs.net.
bifrost A      129.187.145.241
forseti A      78.47.25.36
AAAA   AAAA   2a01:4f8:190:60a3::2

$TTL 3600 ; 1 hour
traceroute A      89.163.225.145
AAAA   AAAA   2001:4ba0:ffec:0193::0
tcpchat A      89.163.225.145
asciiart CNAME  svm502.net.in.tum.de.
```

Relevanter Teil des Namespaces (der zur abgebildeten Zone File korrespondierende Teil ist hervorgehoben):



## Resolver

**Resolver** sind Server, die Informationen aus dem DNS extrahieren und das Ergebnis an den anfragenden Client zurückliefern.

- ▶ Da das DNS einer verteilten Datenbank entspricht und kein einzelner Nameserver alle Zonen kennt, sind i. A. mehrere Anfragen notwendig.
- ▶ Resolver fragen dabei schrittweise bei den autoritativen Nameservern der jeweiligen Zonen an.
- ▶ Das Ergebnis wird an den anfragenden Client zurückgegeben und kann (hoffentlich unter Beachtung der TTL im SOA Record der jeweiligen Zone) gecached werden.
- ▶ Stellt ein Client innerhalb dieser Zeit nochmal dieselbe Anfrage, kann diese aus dem Cache beantwortet werden.

(Öffentliche) Resolver sind i. d. R. selbst für keine Zonen selbst autoritativ.

**Problem:** Woher weiß ein Resolver, wo er anfangen soll?

## Resolver

**Resolver** sind Server, die Informationen aus dem DNS extrahieren und das Ergebnis an den anfragenden Client zurückliefern.

- ▶ Da das DNS einer verteilten Datenbank entspricht und kein einzelner Nameserver alle Zonen kennt, sind i. A. mehrere Anfragen notwendig.
- ▶ Resolver fragen dabei schrittweise bei den autoritativen Nameservern der jeweiligen Zonen an.
- ▶ Das Ergebnis wird an den anfragenden Client zurückgegeben und kann (hoffentlich unter Beachtung der TTL im SOA Record der jeweiligen Zone) gecached werden.
- ▶ Stellt ein Client innerhalb dieser Zeit nochmal dieselbe Anfrage, kann diese aus dem Cache beantwortet werden.

(Öffentliche) Resolver sind i. d. R. selbst für keine Zonen selbst autoritativ.

**Problem:** Woher weiß ein Resolver, wo er anfangen soll?

- ▶ Resolver verfügen über eine statische Liste der 13 Root-Server<sup>1</sup>, die für die Root-Zone autoritativ sind.
- ▶ Die Root-Zone wird von der [Internet Corporation for Assigned Names and Numbers](#) verwaltet. Änderungen bedürfen jedoch der Zustimmung durch das [US Department of Commerce](#).
- ▶ Betrieben werden die Root-Server von verschiedenen Organisationen, u. a. ICANN, Versign, U. S. Army, RIPE, NASA, etc.

---

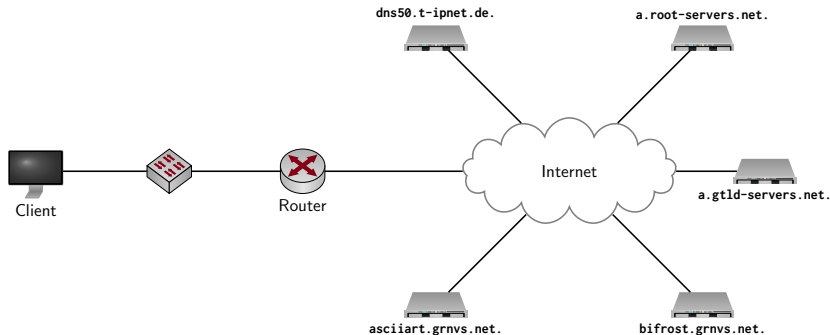
<sup>1</sup> In Wirklichkeit handelt es sich dabei um hunderte Server, welche über die 13 IP-Adressen via Anycast erreichbar sind.

## Ausschnitt der Root-Hints

```
; This file holds the information on root name servers needed to
; initialize cache of Internet domain name servers
; (e.g. reference this file in the "cache . <file>"
; configuration file of BIND domain name servers).
;
; This file is made available by InterNIC
; under anonymous FTP as
;   file           /domain/named.cache
;   on server      FTP.INTERNIC.NET
; -OR-            RS.INTERNIC.NET
;
; last update:    May 23, 2015
; related version of root zone:  2015052300
;
; formerly NS.INTERNIC.NET
;
.
.
A.ROOT-SERVERS.NET. 3600000 NS A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET. 3600000 A 198.41.0.4
A.ROOT-SERVERS.NET. 3600000 AAAA 2001:503:ba3e::2:30
;
; FORMERLY NS1.ISI.EDU
;
.
.
B.ROOT-SERVERS.NET. 3600000 NS B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET. 3600000 A 192.228.79.201
B.ROOT-SERVERS.NET. 3600000 AAAA 2001:500:84::b
;
; FORMERLY C.PSI.NET
;
.
.
.
```

**Beispiel:** Gewöhnlicher privater Internetanschluss eines motivierten Studenten, der zur Bearbeitung der 4. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

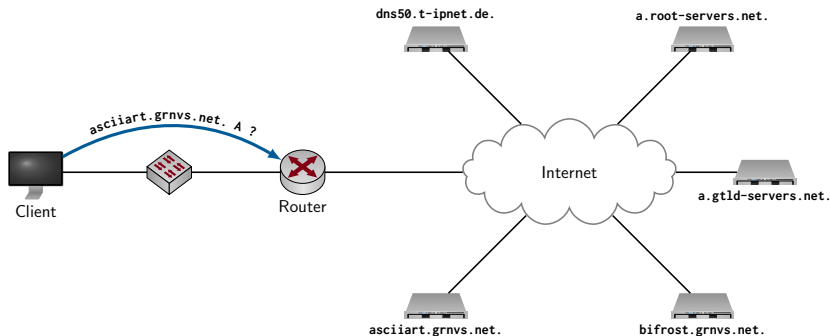
- ▶ Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.<sup>1</sup>
- ▶ Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- ▶ DNS-Anfragen des Clients an den Router sind *rekursiv (recursive queries)*.
- ▶ Die eigentliche Namensauflösung wird vom Resolver `dns50.t-ipnet.de.` mittels einer Reihe von *iterativen Anfragen (iterative queries)* erledigt.



<sup>1</sup> Diesen Vorgang bezeichnet man als [Forwarding](#). Weswegen könnte das sowohl aus Providersicht als auch aus Nutzersicht vorteilhaft sein?

**Beispiel:** Gewöhnlicher privater Internetanschluss eines motivierten Studenten, der zur Bearbeitung der 4. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

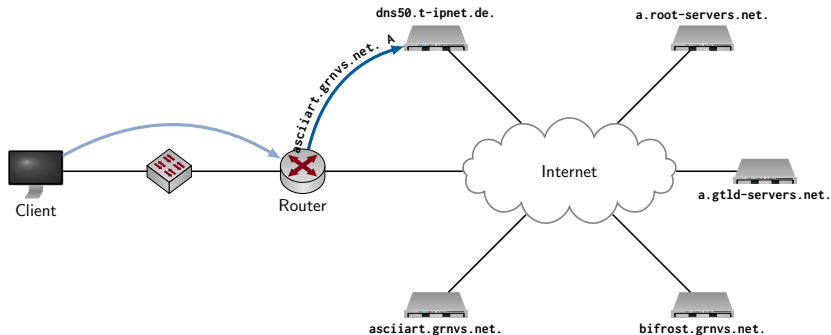
- ▶ Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.<sup>1</sup>
- ▶ Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- ▶ DNS-Anfragen des Clients an den Router sind *rekursiv (recursive queries)*.
- ▶ Die eigentliche Namensauflösung wird vom Resolver `dns50.t-ipnet.de.` mittels einer Reihe von *iterativen Anfragen (iterative queries)* erledigt.



<sup>1</sup> Diesen Vorgang bezeichnet man als [Forwarding](#). Weswegen könnte das sowohl aus Providersicht als auch aus Nutzersicht vorteilhaft sein?

**Beispiel:** Gewöhnlicher privater Internetanschluss eines motivierten Studenten, der zur Bearbeitung der 4. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

- ▶ Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.<sup>1</sup>
- ▶ Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- ▶ DNS-Anfragen des Clients an den Router sind *rekursiv (recursive queries)*.
- ▶ Die eigentliche Namensauflösung wird vom Resolver `dns50.t-ipnet.de.` mittels einer Reihe von *iterativen Anfragen (iterative queries)* erledigt.

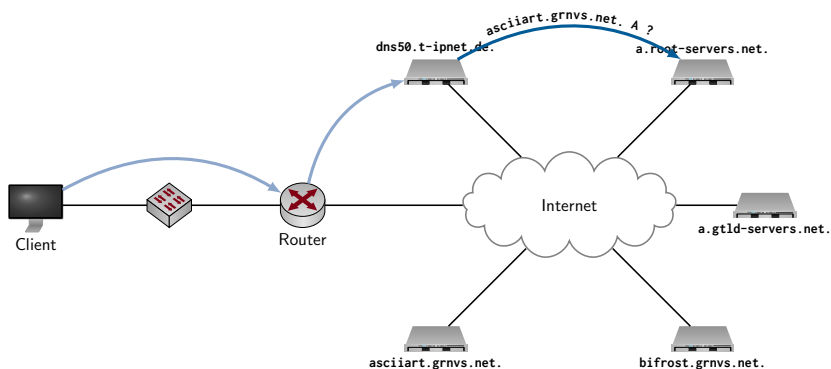


<sup>1</sup> Diesen Vorgang bezeichnet man als *Forwarding*. Weswegen könnte das sowohl aus Providersicht als auch aus Nutzersicht vorteilhaft sein?



**Beispiel:** Gewöhnlicher privater Internetanschluss eines motivierten Studenten, der zur Bearbeitung der 4. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

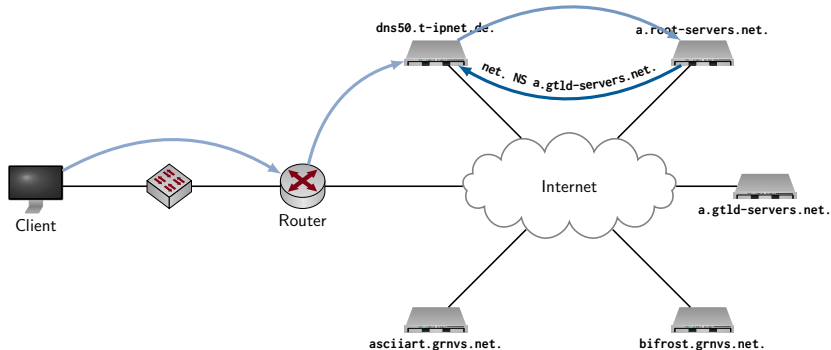
- ▶ Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.<sup>1</sup>
- ▶ Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- ▶ DNS-Anfragen des Clients an den Router sind **rekursiv (recursive queries)**.
- ▶ Die eigentliche Namensauflösung wird vom Resolver `dns50.t-ipnet.de.` mittels einer Reihe von **iterativen Anfragen (iterative queries)** erledigt.



<sup>1</sup> Diesen Vorgang bezeichnet man als **Forwarding**. Weswegen könnte das sowohl aus Providersicht als auch aus Nutzersicht vorteilhaft sein?

**Beispiel:** Gewöhnlicher privater Internetanschluss eines motivierten Studenten, der zur Bearbeitung der 4. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

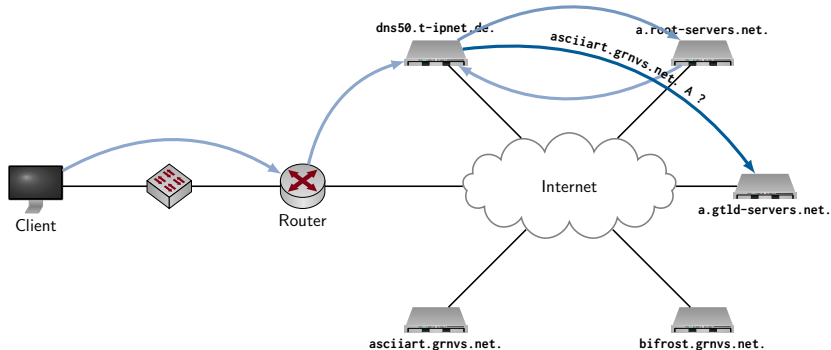
- ▶ Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.<sup>1</sup>
- ▶ Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- ▶ DNS-Anfragen des Clients an den Router sind **rekursiv (recursive queries)**.
- ▶ Die eigentliche Namensauflösung wird vom Resolver `dns50.t-ipnet.de.` mittels einer Reihe von **iterativen Anfragen (iterative queries)** erledigt.



<sup>1</sup> Diesen Vorgang bezeichnet man als **Forwarding**. Weswegen könnte das sowohl aus Providersicht als auch aus Nutzersicht vorteilhaft sein?

**Beispiel:** Gewöhnlicher privater Internetanschluss eines motivierten Studenten, der zur Bearbeitung der 4. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

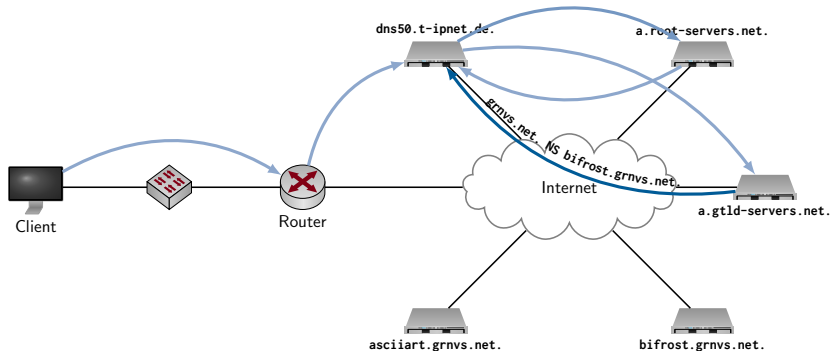
- ▶ Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.<sup>1</sup>
- ▶ Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- ▶ DNS-Anfragen des Clients an den Router sind *rekursiv (recursive queries)*.
- ▶ Die eigentliche Namensauflösung wird vom Resolver `dns50.t-ipnet.de.` mittels einer Reihe von *iterativen Anfragen (iterative queries)* erledigt.



<sup>1</sup> Diesen Vorgang bezeichnet man als **Forwarding**. Weswegen könnte das sowohl aus Providersicht als auch aus Nutzersicht vorteilhaft sein?

**Beispiel:** Gewöhnlicher privater Internetanschluss eines motivierten Studenten, der zur Bearbeitung der 4. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

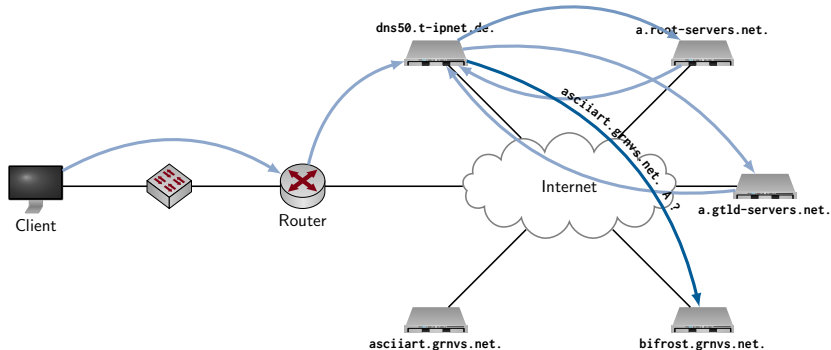
- ▶ Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.<sup>1</sup>
- ▶ Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- ▶ DNS-Anfragen des Clients an den Router sind *rekursiv (recursive queries)*.
- ▶ Die eigentliche Namensauflösung wird vom Resolver `dns50.t-ipnet.de.` mittels einer Reihe von *iterativen Anfragen (iterative queries)* erledigt.



<sup>1</sup> Diesen Vorgang bezeichnet man als **Forwarding**. Weswegen könnte das sowohl aus Providersicht als auch aus Nutzersicht vorteilhaft sein?

**Beispiel:** Gewöhnlicher privater Internetanschluss eines motivierten Studenten, der zur Bearbeitung der 4. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

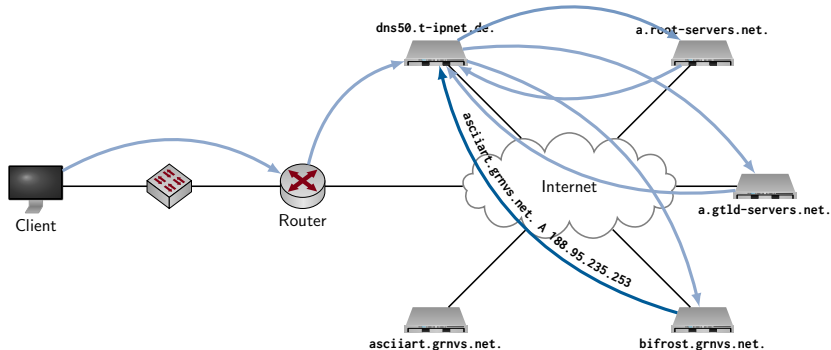
- ▶ Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.<sup>1</sup>
- ▶ Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- ▶ DNS-Anfragen des Clients an den Router sind *rekursiv (recursive queries)*.
- ▶ Die eigentliche Namensauflösung wird vom Resolver `dns50.t-ipnet.de.` mittels einer Reihe von *iterativen Anfragen (iterative queries)* erledigt.



<sup>1</sup> Diesen Vorgang bezeichnet man als **Forwarding**. Weswegen könnte das sowohl aus Providersicht als auch aus Nutzersicht vorteilhaft sein?

**Beispiel:** Gewöhnlicher privater Internetanschluss eines motivierten Studenten, der zur Bearbeitung der 4. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

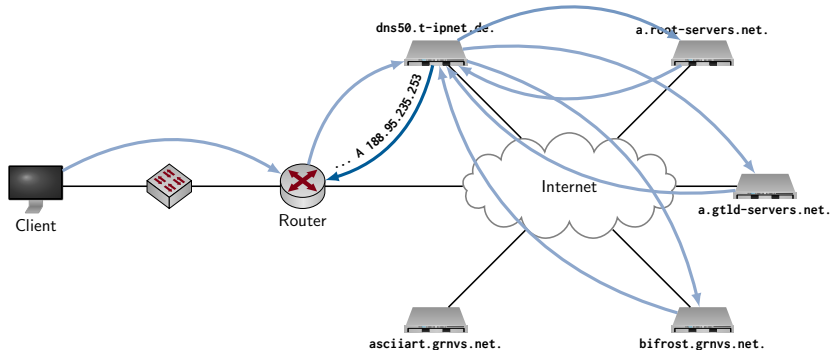
- ▶ Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.<sup>1</sup>
- ▶ Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- ▶ DNS-Anfragen des Clients an den Router sind *rekursiv (recursive queries)*.
- ▶ Die eigentliche Namensauflösung wird vom Resolver `dns50.t-ipnet.de.` mittels einer Reihe von *iterativen Anfragen (iterative queries)* erledigt.



<sup>1</sup> Diesen Vorgang bezeichnet man als **Forwarding**. Weswegen könnte das sowohl aus Providersicht als auch aus Nutzersicht vorteilhaft sein?

**Beispiel:** Gewöhnlicher privater Internetanschluss eines motivierten Studenten, der zur Bearbeitung der 4. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

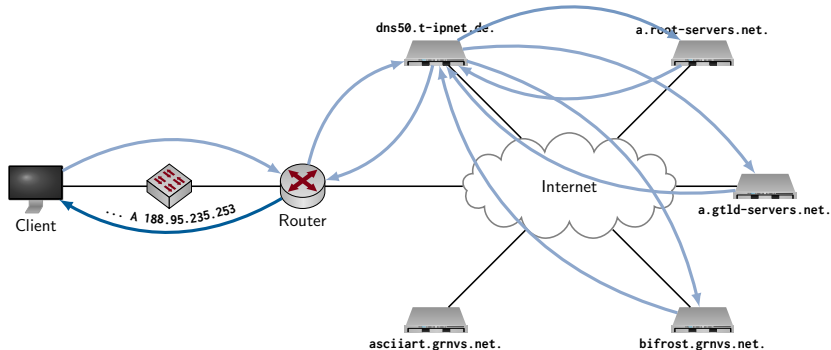
- ▶ Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.<sup>1</sup>
- ▶ Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- ▶ DNS-Anfragen des Clients an den Router sind *rekursiv (recursive queries)*.
- ▶ Die eigentliche Namensauflösung wird vom Resolver `dns50.t-ipnet.de.` mittels einer Reihe von *iterativen Anfragen (iterative queries)* erledigt.



<sup>1</sup> Diesen Vorgang bezeichnet man als **Forwarding**. Weswegen könnte das sowohl aus Providersicht als auch aus Nutzersicht vorteilhaft sein?

**Beispiel:** Gewöhnlicher privater Internetanschluss eines motivierten Studenten, der zur Bearbeitung der 4. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

- ▶ Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.<sup>1</sup>
- ▶ Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- ▶ DNS-Anfragen des Clients an den Router sind *rekursiv (recursive queries)*.
- ▶ Die eigentliche Namensauflösung wird vom Resolver `dns50.t-ipnet.de.` mittels einer Reihe von *iterativen Anfragen (iterative queries)* erledigt.



<sup>1</sup> Diesen Vorgang bezeichnet man als **Forwarding**. Weswegen könnte das sowohl aus Providersicht als auch aus Nutzersicht vorteilhaft sein?



## Reverse DNS

Im DNS können mittels **PTR (Pointer) Records** auch FQDNs zu IP-Adressen hinterlegt werden. Dies bezeichnet man als **Reverse DNS**. Hierzu existiert für IPv4 und IPv6 jeweils eine eigene Zone:

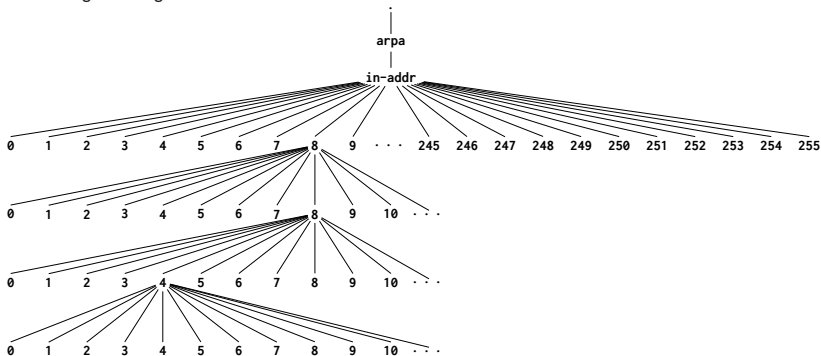
- ▶ **in-addr.arpa.** für IPv4
- ▶ **ip6.arpa.** für IPv6

## Reverse DNS

Im DNS können mittels **PTR (Pointer) Records** auch FQDNs zu IP-Adressen hinterlegt werden. Dies bezeichnet man als **Reverse DNS**. Hierzu existiert für IPv4 und IPv6 jeweils eine eigene Zone:

- ▶ **in-addr.arpa.** für IPv4
- ▶ **ip6.arpa.** für IPv6

Für IPv4 wird der Namespace unterhalb von `in-addr.arpa.` durch die vier Oktette in umgekehrter Reihenfolge erzeugt:

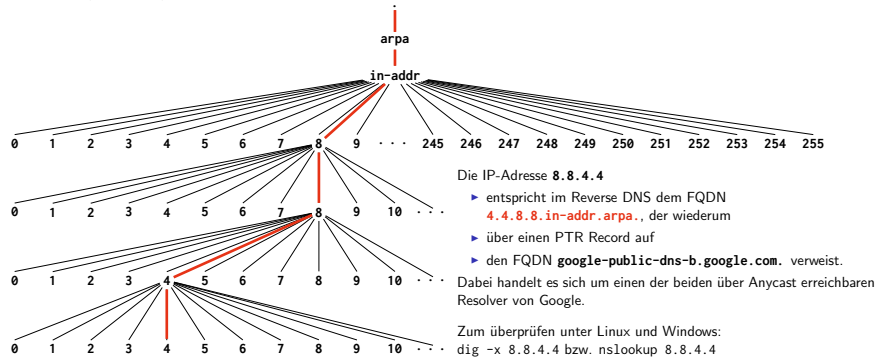


## Reverse DNS

Im DNS können mittels **PTR (Pointer) Records** auch FQDNs zu IP-Adressen hinterlegt werden. Dies bezeichnet man als **Reverse DNS**. Hierzu existiert für IPv4 und IPv6 jeweils eine eigene Zone:

- ▶ **in-addr.arpa.** für IPv4
- ▶ **ip6.arpa.** für IPv6

Für IPv4 wird der Namespace unterhalb von `in-addr.arpa.` durch die vier Oktette in umgekehrter Reihenfolge erzeugt:



Für den Namespace unterhalb von `in-addr.arpa` ergeben sich einige Einschränkungen:

- ▶ Da jede Ebene einem ganzen Oktett entspricht, gibt es maximal vier Ebenen.
- ▶ Subnetze, deren Präfixlänge nicht 8, 16, 24 oder 32 ist, können nicht in getrennten Zonen gespeichert werden (letztere entsprechen einzelnen IP-Adressen).
- ▶ Die Abbildung von Subnetzen anderer Größen ist nur mit Tricks möglich.

Für den Namespace unterhalb von `in-addr.arpa` ergeben sich einige Einschränkungen:

- ▶ Da jede Ebene einem ganzen Oktett entspricht, gibt es maximal vier Ebenen.
- ▶ Subnetze, deren Präfixlänge nicht 8, 16, 24 oder 32 ist, können nicht in getrennten Zonen gespeichert werden (letztere entsprechen einzelnen IP-Adressen).
- ▶ Die Abbildung von Subnetzen anderer Größen ist nur mit Tricks möglich.

Der Namespace für IPv6 (unterhalb von `ip6.arpa`.) ist sehr ähnlich aufgebaut:

- ▶ Die Aufteilung findet an 4 bit-Grenzen anstelle ganzer Oktette statt.
- ▶ Dies erweitert die Möglichkeiten zur Aufteilung in getrennte Zonen.
- ▶ Der Namespace ist infolge der 128 bit langen IPv6-Adressen entsprechend größer.