

Gruppen Di-T14 / Mi-T25

Tutorübung zu Grundlagen: Rechnernetze und Verteilte Systeme (SS 16)

Michael Schwarz

Institut für Informatik
Technische Universität München

05.07 / 06.07.2016

Anwendungsschicht

Die **Anwendungsschicht (Application Layer)** ist die oberste Schicht der 7 Schichten des ISO/OSI-Referenzmodells.¹ Protokolle der Anwendungsschicht stellen spezifische Dienste bereit.

Beispiele:

- **Domain Name System (DNS)**
Auflösung sog. vollqualifizierter Domännennamen² in IP-Adressen und umgekehrt.
- **Hyper Text Transfer Protocol (HTTP)**
Protokoll zum Transfer von Webseiten und Daten.
- **File Transfer Protocol (FTP)**
Protokoll zum Transfer von Dateien von und zu Servern.
- **Simple Mail Transfer Protocol (SMTP)**
Dient dem Versand von Emails sowie der Kommunikation zwischen Mailservern.
- **Post Office Protocol (POP)** und **Internet Message Access Protocol (IMAP)**
Protokolle, mit denen ein Mail User Agent auf zugestellte Emails zugreifen kann.
- **Telnet**
Einfaches Protokoll zur interaktiven Kommunikation mit einem anderen Host (vgl. TCP-Chat Client).
- **Secure Shell (SSH)**
Verschlüsselte entfernte Anmeldung an einem Host.
- **Simple Network Management Protocol (SNMP)**
Dient dem Monitoring und dem Management von Netzkomponenten.

¹ In vielen Fällen sind Protokolle der Anwendungsschicht Bestandteil der Anwendungen selbst, also innerhalb einer Anwendung implementiert und nicht Bestandteil des Betriebssystems oder einer Middleware.

² Umgangssprachlich als „Webadressen“ bezeichnet.

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 überlicherweise mittels eines hierarchisch aufgebauten Namens, z. B. www.google.com.

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

1. Der **Domain Namespace**

- ist ein hierarchisch aufgebauter Namensraum, und
- hat eine baumartige Struktur.

2. **Nameserver**

- speichern Informationen über den Namensraum,
- jeder Server kennt nur kleine Ausschnitte des Namensraums.

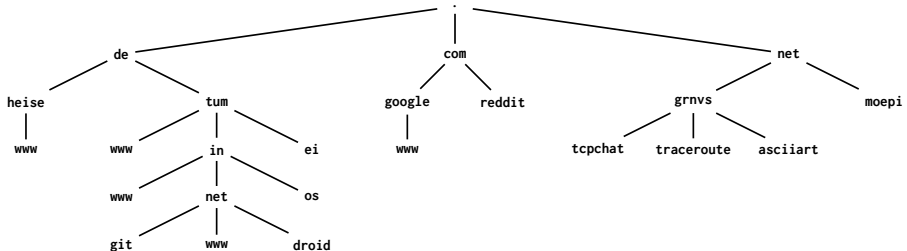
3. **Resolver** sind Programme,

- die durch Anfragen an Nameserver Informationen aus dem Namespace extrahieren, und
- anfragenden Clients bzw. Anwendungen zur Verfügung stellen.

Details: siehe entsprechende RFCs [6, 7, 2, 4]

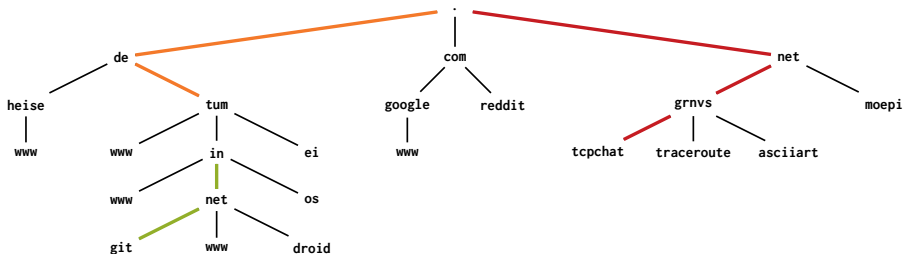
Domain Namespace

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. **git.net.in.**
 - Ein FQDN kann als **Suffix** für einen nicht-qualifizierten Namen verwendet werden, z. B. ergibt **git.net.in** zusammen mit dem FQDN **tum.de.** einen neuen FQDN **git.net.in.tum.de.**
 - Ob ein FQDN existiert (z. B. in eine Adresse aufgelöst werden kann), bleibt zunächst offen.

Nameserver

Der Namespace wird

- in Form einer verteilten Datenbank
- von einer 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 Port UDP/TCP 53:

- Anfragen werden meist an UDP 53 gestellt.
- Anfragen, die Größer als 512 B sind, werden typischerweise 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). Kann für unterschiedliche Zwecke verwendet werden.
- **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.

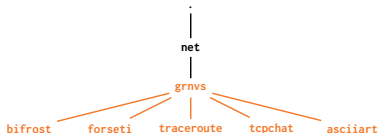
Die Resource Records einer Zone werden auf Nameservern in Form von **Zone Files** gespeichert:

```
$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    2a01:4f8:190:60a3::2

$TTL 3600 ; 1 hour
traceroute A       89.163.225.145
AAAA    2001:4ba0:ffec:0193::0
tcpchat A       89.163.225.145
asciiart CNAME    svm999.net.in.tum.de.
```

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



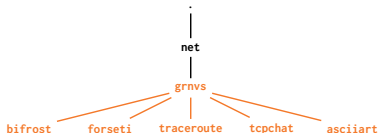
Die Resource Records einer Zone werden auf Nameservern in Form von **Zone Files** gespeichert:

```
$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  2a01:4f8:190:60a3::2

$TTL 3600 ; 1 hour
traceroute A 89.163.225.145
AAAA  2001:4ba0:ffec:0193::0
tcpchat A 89.163.225.145
asciiart CNAME svm999.net.in.tum.de.
```

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



- Setzt die **Time to Live**¹ der nachfolgenden Resource Records auf 1 d.

¹Bitte nicht mit der TTL von IPv4-Paketen verwechseln.

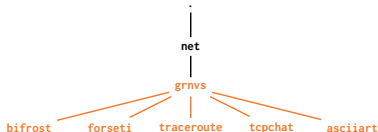
Die Resource Records einer Zone werden auf Nameservern in Form von **Zone Files** gespeichert:

```
$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 2a01:4f8:190:60a3::2

$TTL 3600 ; 1 hour
traceroute A 89.163.225.145
AAAA 2001:4ba0:ffec:0193::0
tcpchat A 89.163.225.145
asciiart CNAME svm999.net.in.tum.de.
```

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



- SOA Record für die Domain `grnvs.net.`²

²Im Zone File ist `grnvs.net` als relativer Domain Name angegeben (also kein FQDN), da der Ausgangspunkt zuvor via `$ORIGIN` festgelegt wurde.

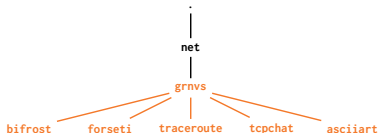
Die Resource Records einer Zone werden auf Nameservern in Form von **Zone Files** gespeichert:

```
$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  2a01:4f8:190:60a3::2

$TTL 3600 ; 1 hour
traceroute A 89.163.225.145
AAAA  2001:4ba0:ffec:0193::0
tcpchat A 89.163.225.145
asciiart CNAME svm999.net.in.tum.de.
```

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



- SOA Record für die Domain `grnvs.net.`²
 - `serial` gibt die Version der Zonendatei an.
 - Mit jedem Update der Zonendatei, z.B. beim Hinzufügen, Entfernen oder Modifizieren von Resource Records, muss diese inkrementiert werden.
 - Dient insbesondere bei Zonentransfers dazu festzustellen, ob ein Secondary veraltete Informationen besitzt.

²Im Zone File ist `grnvs.net` als relativer Domain Name angegeben (also kein FQDN), da der Ausgangspunkt zuvor via `$ORIGIN` festgelegt wurde.

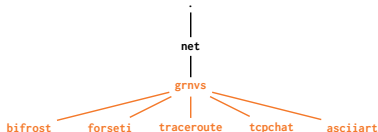
Die Resource Records einer Zone werden auf Nameservern in Form von **Zone Files** gespeichert:

```
$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    2a01:4f8:190:60a3::2

$TTL 3600 ; 1 hour
traceroute A     89.163.225.145
AAAA      2001:4ba0:ffec:0193::0
tcpchat   A     89.163.225.145
asciart    CNAME svm999.net.in.tum.de.
```

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



- SOA Record für die Domain `grnvs.net.`²
 - refresh gibt das Zeitintervall an, in dem Secondaries versuchen, ihre lokale Kopie der Zonendatei mit dem Primary abzugleichen.

²Im Zone File ist `grnvs.net` als relativer Domain Name angegeben (also kein FQDN), da der Ausgangspunkt zuvor via `$ORIGIN` festgelegt wurde.

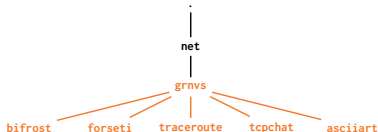
Die Resource Records einer Zone werden auf Nameservern in Form von **Zone Files** gespeichert:

```
$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  2a01:4f8:190:60a3::2

$TTL 3600 ; 1 hour
traceroute A 89.163.225.145
AAAA  2001:4ba0:ffec:0193::0
tcpchat A 89.163.225.145
asciart  CNAME svm999.net.in.tum.de.
```

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



- SOA Record für die Domain `grnvs.net.`²
 - `retry` gibt das Zeitintervall an, in dem Secondaries versuchen, ihren Primary zu kontaktieren, falls der vorherige Versuch fehlgeschlagen ist.

²Im Zone File ist `grnvs.net` als relativer Domain Name angegeben (also kein FQDN), da der Ausgangspunkt zuvor via `$ORIGIN` festgelegt wurde.

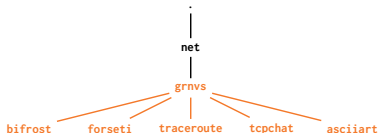
Die Resource Records einer Zone werden auf Nameservern in Form von **Zone Files** gespeichert:

```
$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  2a01:4f8:190:60a3::2

$TTL 3600 ; 1 hour
traceroute A 89.163.225.145
AAAA  2001:4ba0:ffec:0193::0
tcpchat A 89.163.225.145
asciart  CNAME svm999.net.in.tum.de.
```

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



- SOA Record für die Domain `grnvs.net.`²

- `expire` gibt für Secondaries das maximale Zeitintervall seit dem letzten erfolgreichen Abgleich mit seinem Primary an, während dessen die Information des Secondaries als autoritativ gelten.
- Sollte kein Kontakt mit dem Primary innerhalb dieser Zeit möglich sein, stellt der Secondary seinen Betrieb ein.

²Im Zone File ist `grnvs.net` als relativer Domain Name angegeben (also kein FQDN), da der Ausgangspunkt zuvor via `$ORIGIN` festgelegt wurde.

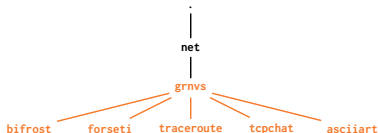
Die Resource Records einer Zone werden auf Nameservern in Form von **Zone Files** gespeichert:

```
$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  2a01:4f8:190:60a3::2

$TTL 3600 ; 1 hour
traceroute A 89.163.225.145
AAAA  2001:4ba0:ffec:0193::0
tcpchat A 89.163.225.145
asciiart CNAME svm999.net.in.tum.de.
```

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



- SOA Record für die Domain `grnvs.net.`²
 - `nxdomain`³ gibt an, wie lange anfragende Resolver in ihrem Cache halten dürfen, dass ein angefragter Resource Record nicht existiert.

²Im Zone File ist `grnvs.net` als relativer Domain Name angegeben (also kein FQDN), da der Ausgangspunkt zuvor via `$ORIGIN` festgelegt wurde.

³Dieses Feld hatte früher eine andere Bedeutung.

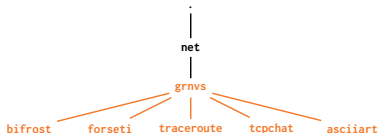
Die Resource Records einer Zone werden auf Nameservern in Form von **Zone Files** gespeichert:

```
$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 2a01:4f8:190:60a3::2

$TTL 3600 ; 1 hour
traceroute A 89.163.225.145
AAAA 2001:4ba0:ffec:0193::0
tcpchat A 89.163.225.145
asciiart CNAME svm999.net.in.tum.de.
```

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



- Die NS Records geben die FQDNs der autoritativen Nameserver für diese Zone an.
- Deren FQDNs müssen nicht notwendigerweise dieselbe Endung wie die aktuelle Zone haben.

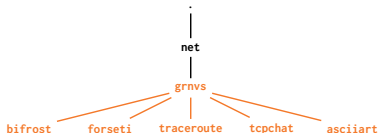
Die Resource Records einer Zone werden auf Nameservern in Form von **Zone Files** gespeichert:

```
$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 2a01:4f8:190:60a3::2

$TTL 3600 ; 1 hour
traceroute A 89.163.225.145
AAAA 2001:4ba0:ffec:0193::0
tcpchat A 89.163.225.145
asciiart CNAME svm999.net.in.tum.de.
```

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



- A Record für den FQDN **grnvs.net**.
- Was geschieht, wenn dieser fehlt, sieht man beim Versuch den FQDN **ei.tum.de** im Browser aufzurufen.

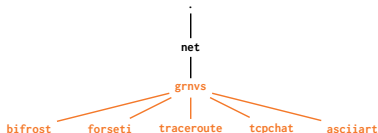
Die Resource Records einer Zone werden auf Nameservern in Form von **Zone Files** gespeichert:

```
$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 2a01:4f8:190:60a3::2
```

```
$TTL 3600 ; 1 hour
traceroute A 89.163.225.145
AAAA 2001:4ba0:ffec:0193::0
tcpchat A 89.163.225.145
asciiart CNAME svm999.net.in.tum.de.
```

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



- Gibt an, dass alle nachfolgenden Domains in Resource Records, für die dieser Server autoritativ ist, relativ zu **\$ORIGIN** zu sehen sind.
- Spart Schreibarbeit bei den nachfolgenden Resource Records, die alle auf **grnvs.net.** enden.

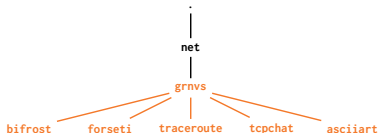
Die Resource Records einer Zone werden auf Nameservern in Form von **Zone Files** gespeichert:

```
$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 2a01:4f8:190:60a3::2

$TTL 3600 ; 1 hour
traceroute A 89.163.225.145
AAAA 2001:4ba0:ffec:0193::0
tcpchat A 89.163.225.145
asciiart CNAME svm999.net.in.tum.de.
```

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



- A Records für insgesamt vier Hosts.
 - Die ersten beiden sind für die beiden autoritativen Nameserver der Zone.
 - Die anderen beiden verweisen auf dieselbe IP-Adresse, d. h. `traceroute.grnvs.net.` und `tcpchat.grnvs.net.` sind in Wirklichkeit derselbe Host.

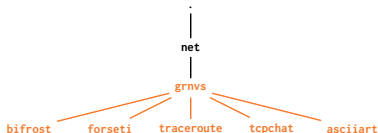
Die Resource Records einer Zone werden auf Nameservern in Form von **Zone Files** gespeichert:

```
$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   2a01:4f8:190:60a3::2

$TTL 3600 ; 1 hour
traceroute A      89.163.225.145
AAAA   2001:4ba0:ffec:0193::0
tcpchat A      89.163.225.145
asciiart CNAME  svm999.net.in.tum.de.
```

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



- AAAA Records für zwei Hosts (IPv6).

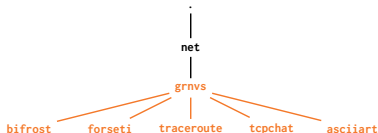
Die Resource Records einer Zone werden auf Nameservern in Form von **Zone Files** gespeichert:

```
$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    2a01:4f8:190:60a3::2

$TTL 3600 ; 1 hour
traceroute A       89.163.225.145
AAAA      2001:4ba0:ffec:0193::0
tcpchat  A       89.163.225.145
asciiart CNAME  svm999.net.in.tum.de.
```

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



- CNAME Record, der auf eine unserer virtuellen Maschinen verweist.
- Das Ziel `svm999.net.in.tum.de.` liegt außerhalb dieser Zone.

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. für keine Zone selbst autoritativ.

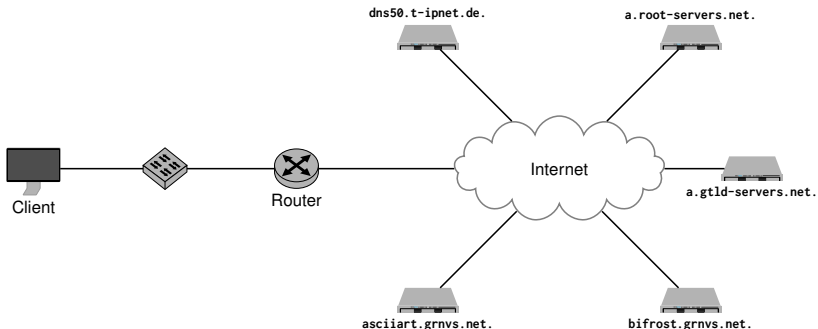
Problem: Woher weiß ein Resolver, wo er anfangen soll?

- Resolver verfügen über eine statische Liste der 13 Root-Server¹, 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.

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

Beispiel: Gewöhnlicher privater Internetanschluss eines motivierten Studenten, der zur Bearbeitung der 5. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

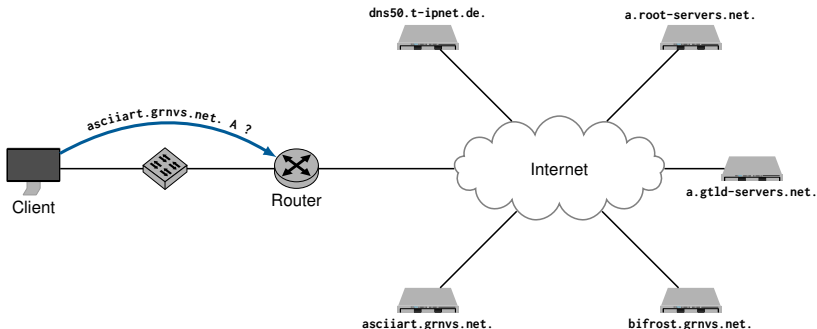
- Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.¹
- Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- DNS-Anfragen des Clients an den Router sind **rekursiv (recursive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



¹ 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 5. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

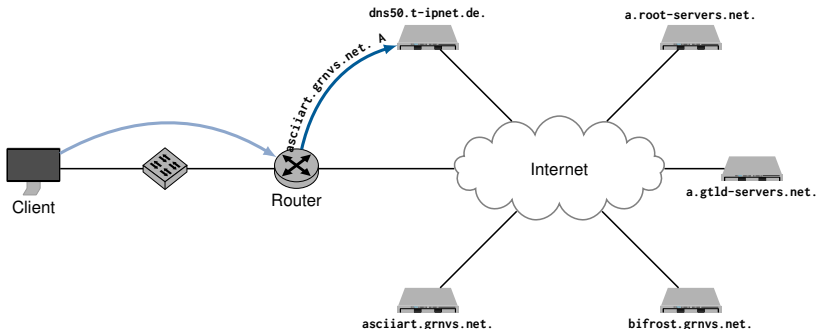
- Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.¹
- Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- DNS-Anfragen des Clients an den Router sind **rekursiv (recursive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



¹ 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 5. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

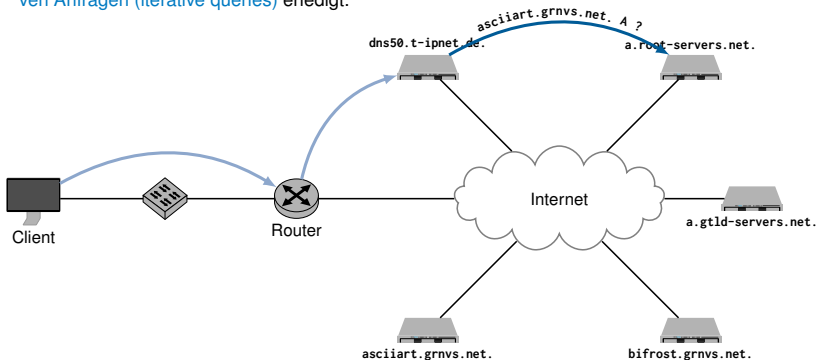
- Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.¹
- Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- DNS-Anfragen des Clients an den Router sind **rekursiv (recursive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



¹ 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 5. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

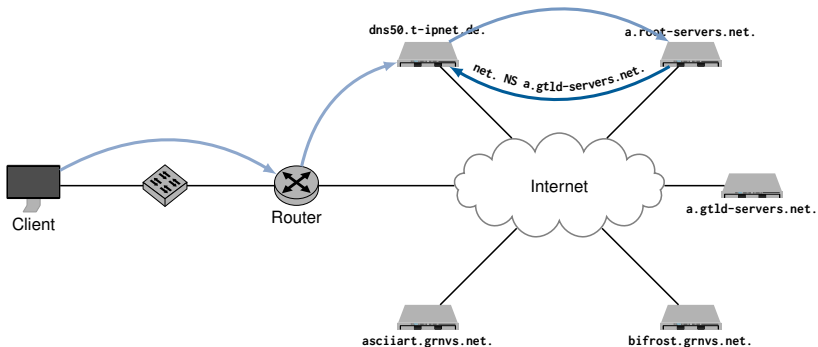
- Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.¹
- Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- DNS-Anfragen des Clients an den Router sind **rekursiv (recursive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



¹ 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 5. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

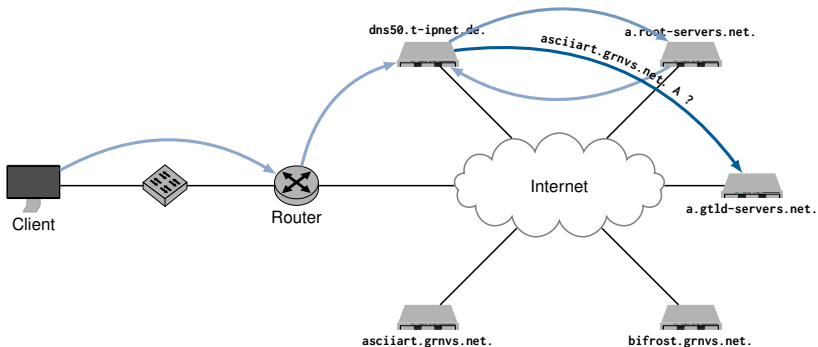
- Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.¹
- Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- DNS-Anfragen des Clients an den Router sind **rekursiv (recursive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



¹ 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 5. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

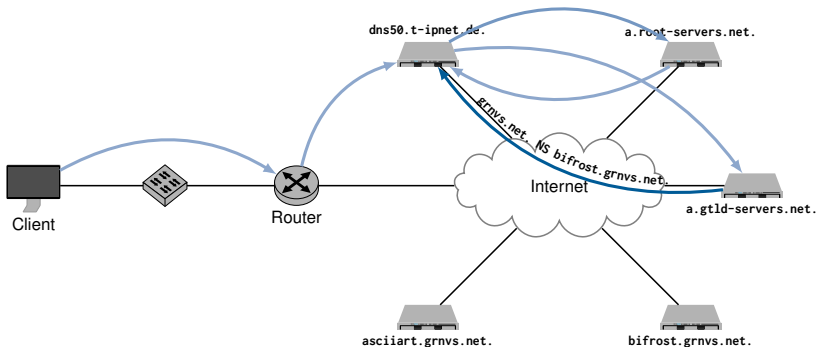
- Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.¹
- Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- DNS-Anfragen des Clients an den Router sind **rekursiv (recursive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



¹ 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 5. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

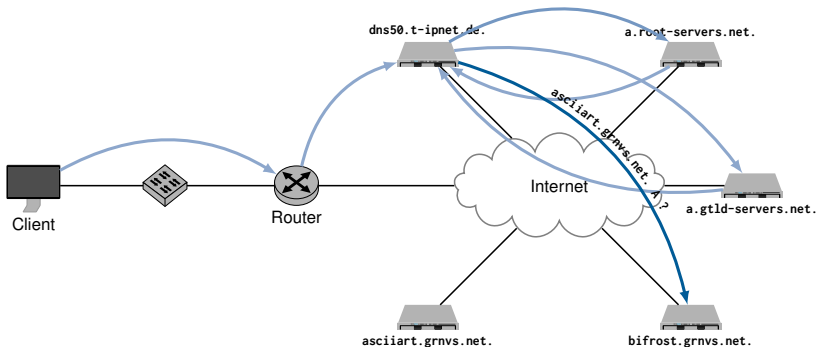
- Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.¹
- Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- DNS-Anfragen des Clients an den Router sind **rekursiv (recursive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



¹ 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 5. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

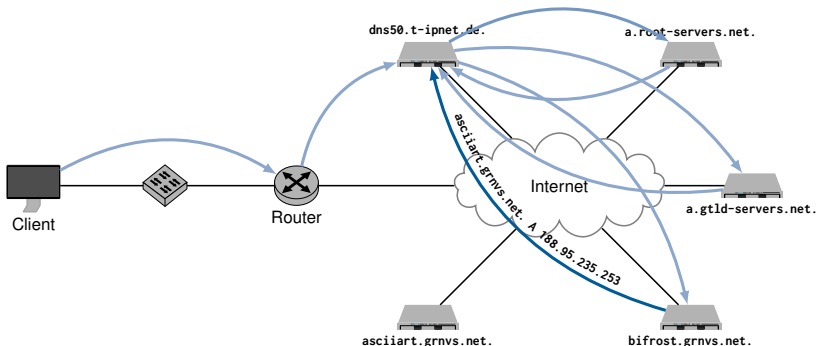
- Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.¹
- Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- DNS-Anfragen des Clients an den Router sind **rekursiv (recursive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



¹ 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 5. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

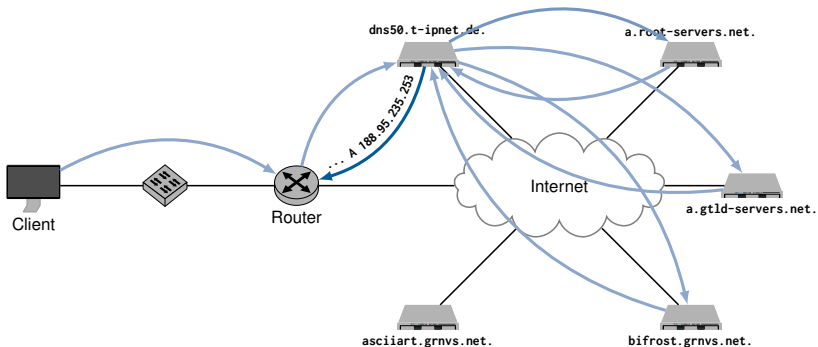
- Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.¹
- Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- DNS-Anfragen des Clients an den Router sind **rekursiv (recursive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



¹ 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 5. Programmieraufgabe mittels Webbrowser auf den Server mit FQDN `asciiart.grnvs.net.` zugreift.

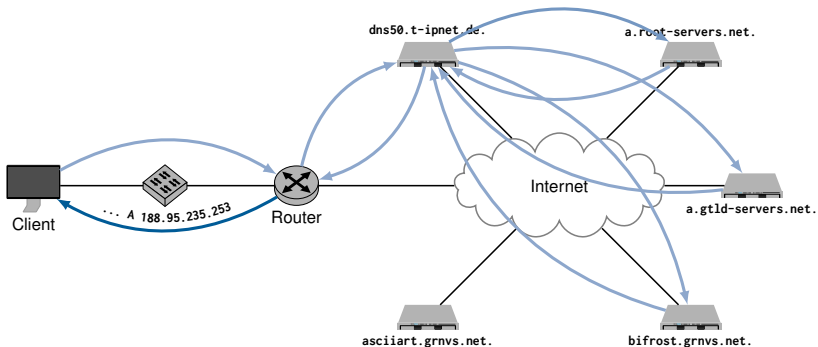
- Der Router arbeitet zwar als Resolver, leitet sämtliche Anfragen aber an einen Resolver des Providers weiter. Dessen IP-Adresse sei dem Router bekannt.¹
- Der Client verwendet den Router als Resolver. Dessen IP-Adresse ist bekannt.
- DNS-Anfragen des Clients an den Router sind **rekursiv (recursive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



¹ 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 5. 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.¹
- 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.



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

Bei Kompressionsverfahren muss unterscheiden werden:

1. Verlustfreie Komprimierung (engl. lossless compression)

- Komprimierte Daten können verlustfrei, d. h. exakt und ohne Informationsverlust, wiederhergestellt werden.
- Verlustfrei komprimierte Dateiformate sind beispielsweise ZIP, PNG² (Bilder), FLAC³ (Musik), ...

2. Verlustbehaftete Komprimierung (engl. lossy compression):

- Komprimierte Daten können im Allgemeinen nicht wieder exakt rekonstruiert werden.
- Es tritt also ein Verlust von Information bei der Komprimierung auf.
- Dafür ermöglichen diese Verfahren meist höhere und in Abhängigkeit des Verlustfaktors variable Kompressionsraten.
- Verlustbehaftet komprimierte Dateiformate sind beispielsweise MP3, MPEG, JPEG, ...

²Portable Network Graphics

³Free Lossless Audio Codec

Beispiel 1: Huffman-Code

- Viele Protokolle komprimieren Daten vor dem Senden (Quellenkodierung).

Grundlegende Idee der Huffman-Kodierung:

- Nicht alle Textzeichen treten mit derselben Häufigkeit auf, z. B. tritt der Buchstabe „E“ in der deutschen Sprache mit einer Häufigkeit von 17,4 % gefolgt von „N“ mit 9,8 % auf.
- Anstelle Zeichen mit uniformer Codewortlänge zu kodieren (z. B. ASCII-Code), werden **häufigen Zeichen kürzere Codewörter** zugewiesen.
- Die Abbildung zwischen Zeichen und Codewörtern bleibt dabei eindeutig und umkehrbar, weswegen es sich um ein verlustloses Kompressionsverfahren handelt.

Konstruktion eines Huffman-Codes

- Gegeben Sei das Alphabet $\mathcal{A} = \{A, B, C, D, E, F, G, H\}$ sowie Auftretswahrscheinlichkeiten $\Pr[X = z]$ für alle Zeichen $z \in \mathcal{A}$.
- Es sei außerdem vorausgesetzt, dass die einzelnen Zeichen unabhängig voneinander auftreten.¹

z	$\Pr[X = z]$
a	0,02
b	0,03
C	0,05
D	0,08
E	0,12
F	0,15
G	0,25
H	0,30

¹ Andernfalls würden die Aussagen zur Optimalität des Huffman-Codes im Allgemeinen nicht mehr zutreffen.

Konstruktion eines Huffman-Codes

- Gegeben Sei das Alphabet $\mathcal{A} = \{A, B, C, D, E, F, G, H\}$ sowie Auftretswahrscheinlichkeiten $\Pr[X = z]$ für alle Zeichen $z \in \mathcal{A}$.
- Es sei außerdem vorausgesetzt, dass die einzelnen Zeichen unabhängig voneinander auftreten.

z	$\Pr[X = z]$
a	0,02
b	0,03
C	0,05
D	0,08
E	0,12
F	0,15
G	0,25
H	0,30

0,02

A

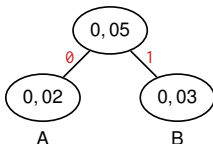
0,03

B

Konstruktion eines Huffman-Codes

- Gegeben Sei das Alphabet $\mathcal{A} = \{A, B, C, D, E, F, G, H\}$ sowie Auftrittswahrscheinlichkeiten $\Pr[X = z]$ für alle Zeichen $z \in \mathcal{A}$.
- Es sei außerdem vorausgesetzt, dass die einzelnen Zeichen unabhängig voneinander auftreten.

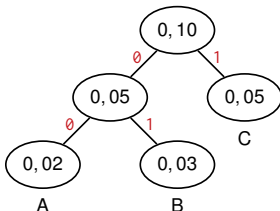
z	$\Pr[X = z]$
a	0,02
b	0,03
C	0,05
D	0,08
E	0,12
F	0,15
G	0,25
H	0,30



Konstruktion eines Huffman-Codes

- Gegeben Sei das Alphabet $\mathcal{A} = \{A, B, C, D, E, F, G, H\}$ sowie Auftretswahrscheinlichkeiten $\Pr[X = z]$ für alle Zeichen $z \in \mathcal{A}$.
- Es sei außerdem vorausgesetzt, dass die einzelnen Zeichen unabhängig voneinander auftreten.

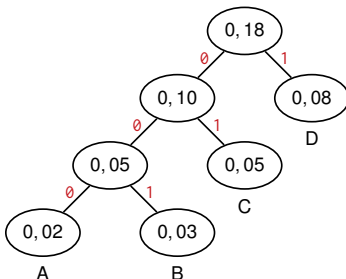
z	$\Pr[X = z]$
a	0,02
b	0,03
C	0,05
D	0,08
E	0,12
F	0,15
G	0,25
H	0,30



Konstruktion eines Huffman-Codes

- Gegeben Sei das Alphabet $\mathcal{A} = \{A, B, C, D, E, F, G, H\}$ sowie Auftretswahrscheinlichkeiten $\Pr[X = z]$ für alle Zeichen $z \in \mathcal{A}$.
- Es sei außerdem vorausgesetzt, dass die einzelnen Zeichen unabhängig voneinander auftreten.

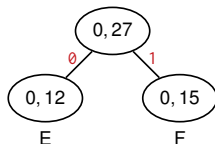
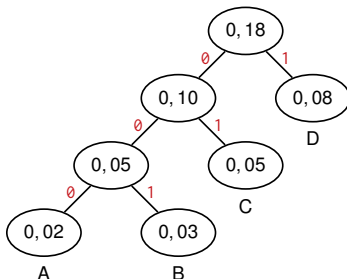
z	$\Pr[X = z]$
a	0,02
b	0,03
C	0,05
D	0,08
E	0,12
F	0,15
G	0,25
H	0,30



Konstruktion eines Huffman-Codes

- Gegeben Sei das Alphabet $\mathcal{A} = \{A, B, C, D, E, F, G, H\}$ sowie Auftretswahrscheinlichkeiten $\Pr[X = z]$ für alle Zeichen $z \in \mathcal{A}$.
- Es sei außerdem vorausgesetzt, dass die einzelnen Zeichen unabhängig voneinander auftreten.

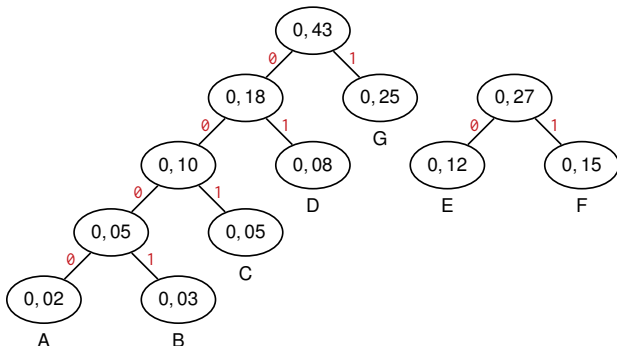
z	$\Pr[X = z]$
a	0,02
b	0,03
C	0,05
D	0,08
E	0,12
F	0,15
G	0,25
H	0,30



Konstruktion eines Huffman-Codes

- Gegeben Sei das Alphabet $\mathcal{A} = \{A, B, C, D, E, F, G, H\}$ sowie Auftretswahrscheinlichkeiten $\Pr[X = z]$ für alle Zeichen $z \in \mathcal{A}$.
- Es sei außerdem vorausgesetzt, dass die einzelnen Zeichen unabhängig voneinander auftreten.

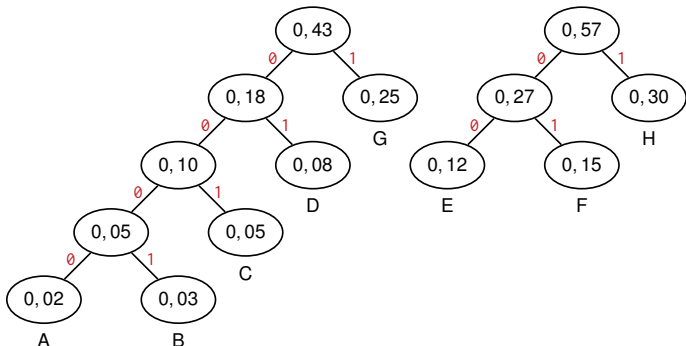
z	$\Pr[X = z]$
a	0,02
b	0,03
C	0,05
D	0,08
E	0,12
F	0,15
G	0,25
H	0,30



Konstruktion eines Huffman-Codes

- Gegeben Sei das Alphabet $\mathcal{A} = \{A, B, C, D, E, F, G, H\}$ sowie Auftretswahrscheinlichkeiten $\Pr[X = z]$ für alle Zeichen $z \in \mathcal{A}$.
- Es sei außerdem vorausgesetzt, dass die einzelnen Zeichen unabhängig voneinander auftreten.

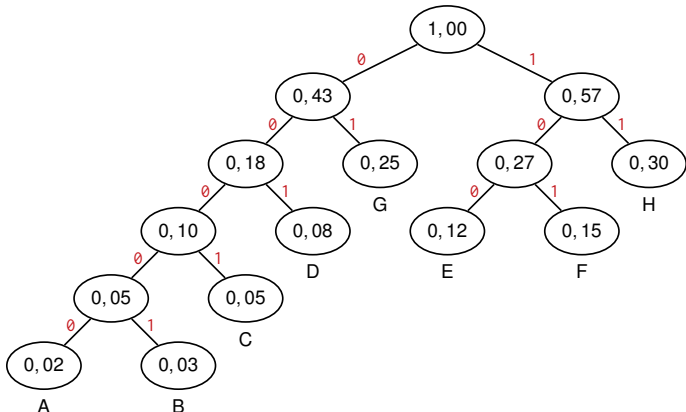
z	$\Pr[X = z]$
a	0,02
b	0,03
C	0,05
D	0,08
E	0,12
F	0,15
G	0,25
H	0,30



Konstruktion eines Huffman-Codes

- Gegeben Sei das Alphabet $\mathcal{A} = \{A, B, C, D, E, F, G, H\}$ sowie Auftretswahrscheinlichkeiten $\Pr[X = z]$ für alle Zeichen $z \in \mathcal{A}$.
- Es sei außerdem vorausgesetzt, dass die einzelnen Zeichen unabhängig voneinander auftreten.

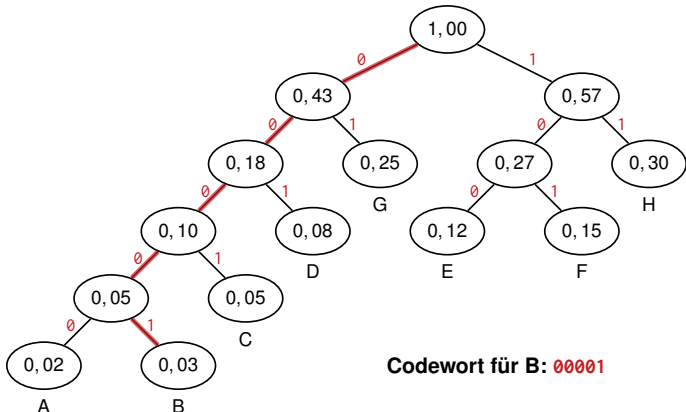
z	$\Pr[X = z]$
a	0,02
b	0,03
C	0,05
D	0,08
E	0,12
F	0,15
G	0,25
H	0,30



Konstruktion eines Huffman-Codes

- Gegeben Sei das Alphabet $\mathcal{A} = \{A, B, C, D, E, F, G, H\}$ sowie Auftretswahrscheinlichkeiten $\Pr[X = z]$ für alle Zeichen $z \in \mathcal{A}$.
- Es sei außerdem vorausgesetzt, dass die einzelnen Zeichen unabhängig voneinander auftreten.

z	$\Pr[X = z]$
a	0,02
b	0,03
C	0,05
D	0,08
E	0,12
F	0,15
G	0,25
H	0,30



Codewort für B: **00001**

Durchschnittliche Codewortlänge

z	$\Pr[X = z]$	Huffman-Code	Länge $l_H(z)$	Uniformer Code
A	0,02	00000	5	000
B	0,03	00001	5	001
C	0,05	0001	4	010
D	0,08	001	3	011
E	0,12	010	3	100
F	0,15	011	3	101
G	0,25	10	2	110
H	0,30	11	2	111

- Uniformer Code:

$E[l_U(z)] = 3,0$, da alle Codewörter gleich lang sind

- Huffman-Code:

$$E[l_H(z)] = \sum_{z \in \mathcal{A}} \Pr[X = z] l_H(z) = 2,6$$

$$\Rightarrow \text{Die Einsparung beträgt } 1 - \frac{E[l_H(z)]}{E[l_U(z)]} \approx 13\%$$

Anmerkungen zum Huffman-Code:

- Statische Huffman-Codes sind darauf angewiesen, dass die Auftrittswahrscheinlichkeit der Zeichen den Erwartungen entspricht.
- Zeichenhäufigkeiten können dynamisch bestimmt werden, allerdings muss dem Empfänger dann das verwendete **Codebuch** mitgeteilt werden.
- Der Huffman-Code ist ein **optimaler** und **präfixfreier** Code.

Definition (Optimaler Präfixcode)

Bei einem **präfixfreien Code** sind gültige Codewörter niemals Präfix eines anderen Codeworts desselben Codes. Ein **optimaler** präfixfreier Code minimiert darüber hinaus die mittlere Codewortlänge

$$\sum_{i \in \mathcal{A}} p(i) \cdot |c(i)|,$$

wobei $p(i)$ die Auftrittswahrscheinlichkeit von $i \in \mathcal{A}$ und $c(i)$ die Abbildung auf ein entsprechendes Codewort bezeichnen.

- Die Huffman-Codierung zählt zu den sogenannten **Entropy-Encoding** Verfahren.
- Huffman-Codierung kann grundsätzlich nicht nur für einzelne Zeichen sondern auch für Zeichenfolgen angewandt werden. Längere Codewörter können infolge der Komplexität zum Bestimmen des Codebuchs aber zu Problemen führen.