Domain Name System

Johannes Garimort

Institut für Informatik
Albert-Ludwigs-Universität Freiburg

5. Februar 2008

Wozu ein Domain Name System?

- IP-Adressen sind unanschaulich und schlecht merkbar

Wozu ein Domain Name System?

- IP-Adressen sind unanschaulich und schlecht merkbar
- Beispiel:
  www.spiegel.de. lässt sich besser handhaben als *195.71.11.67*

Wozu ein Domain Name System?

- IP-Adressen sind unanschaulich und schlecht merkbar
- Beispiel:
  www.spiegel.de. lässt sich besser handhaben als *195.71.11.67*
- benötigen System, dass lesbare Namen in IP-Adressen umwandelt

Entstehung

- HOST-Dateien

Entstehung

- HOST-Dateien
- *Network Information Center (NIC)* verwaltet Domain Namen

Entstehung

- HOST-Dateien
- *Network Information Center (NIC)* verwaltet Domain Namen
- durch wachsende Anzahl von Internetteilnehmern

Entstehung

- HOST-Dateien
- *Network Information Center (NIC)* verwaltet Domain Namen
- durch wachsende Anzahl von Internetteilnehmern
  - zentraler Verwaltungsstelle überfordert

Entstehung

- HOST-Dateien
- *Network Information Center (NIC)* verwaltet Domain Namen
- durch wachsende Anzahl von Internetteilnehmern
  - zentraler Verwaltungsstelle überfordert
  - administrativer und technischer Aufwand zu groß

Domain Name System

- 1983 von Paul Mockapetris entworfen

Domain Name System

- 1983 von Paul Mockapetris entworfen
- beschrieben in den RFCs 882 und 883

Domain Name System

- 1983 von Paul Mockapetris entworfen
- beschrieben in den RFCs 882 und 883
- abgelöst und erweitert durch die RFCs 1034 und 1035

Eigenschaften

- Übersetzung von (Domain-)Namen in IP-Adressen: *"forward lookup"*

Eigenschaften

- Übersetzung von (Domain-)Namen in IP-Adressen: *"forward lookup"*
- umgekehrte Auflösung: *"reverse lookup"*

Eigenschaften

- Übersetzung von (Domain-)Namen in IP-Adressen: *"forward lookup"*
- umgekehrte Auflösung: *"reverse lookup"*
- Vorteile:

Eigenschaften

- Übersetzung von (Domain-)Namen in IP-Adressen: *"forward lookup"*
- umgekehrte Auflösung: *"reverse lookup"*
- Vorteile:
  - zuverlässig und flexibel

Eigenschaften

- Übersetzung von (Domain-)Namen in IP-Adressen: *"forward lookup"*
- umgekehrte Auflösung: *"reverse lookup"*
- Vorteile:
  - zuverlässig und flexibel
  - Änderung interner Netzwerkstrukturen leichter durchführbar

Eigenschaften

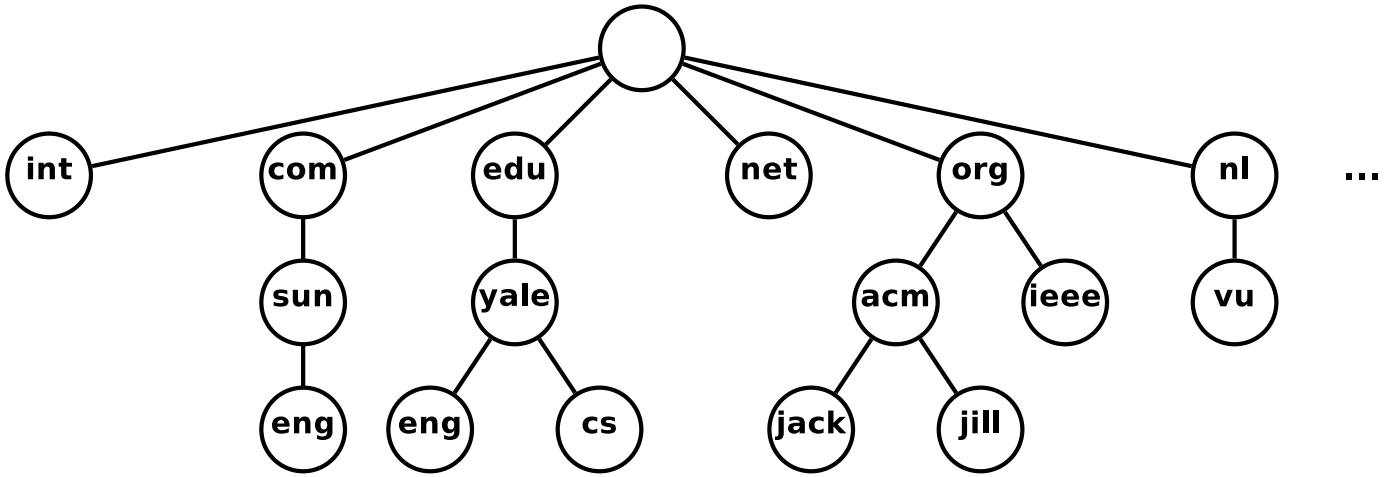
- Übersetzung von (Domain-)Namen in IP-Adressen: *"forward lookup"*
- umgekehrte Auflösung: *"reverse lookup"*
- Vorteile:
  - zuverlässig und flexibel
  - Änderung interner Netzwerkstrukturen leichter durchführbar
  - kann in verschiedenen Strukturen verwendet werden (z.B. Firmen)

Eigenschaften

- Übersetzung von (Domain-)Namen in IP-Adressen: *"forward lookup"*
- umgekehrte Auflösung: *"reverse lookup"*
- Vorteile:
  - zuverlässig und flexibel
  - Änderung interner Netzwerkstrukturen leichter durchführbar
  - kann in verschiedenen Strukturen verwendet werden (z.B. Firmen)
  - ermöglicht eine rudimentäre Lastenverteilung

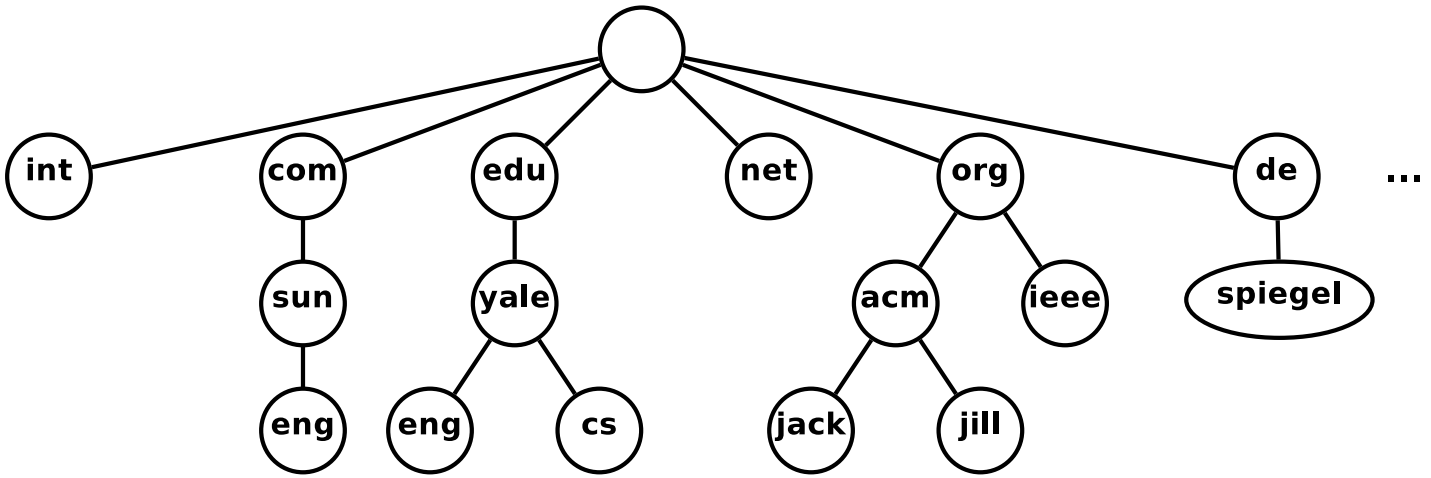
Internet Domain Name Tree

- hierarchisch strukturierter Namensraum
- oberster Knoten: *root*
- erste Ebene: *Top Level Domains*
- Knoten und Blätter heißen Labels



Internet Domain Name Tree

- maximal 255 Zeichen pro Domainname
- maximal 63 Zeichen pro Label
- Domain setzt sich aus Labels zusammen
- Labels stehen weiter rechts umso höher sie im Baum stehen
  Beispiel: *spiegel.de.* (inklusive Punkt: *Fully Qualified Domain-Name (FQDN)*)
- (Sub-)domains werden von darüberliegenden Domains kontrolliert



Name Server

- Name Server sind Programme

Name Server

- Name Server sind Programme
  - die Informationen über Teile des Namensraums speichern

Name Server

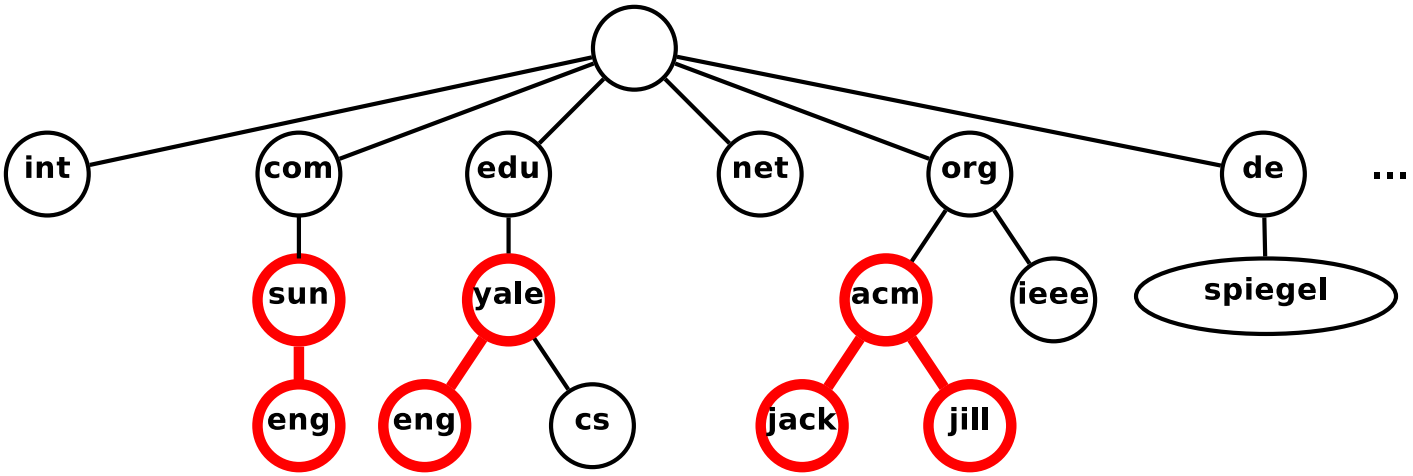
- Name Server sind Programme
  - die Informationen über Teile des Namensraums speichern
  - die Fragen zum Namensraum bearbeiten oder weiterleiten

Name Server

- Name Server sind Programme
  - die Informationen über Teile des Namensraums speichern
  - die Fragen zum Namensraum bearbeiten oder weiterleiten
- Name Server kennen mindestens einen Server aus höherer Ebene

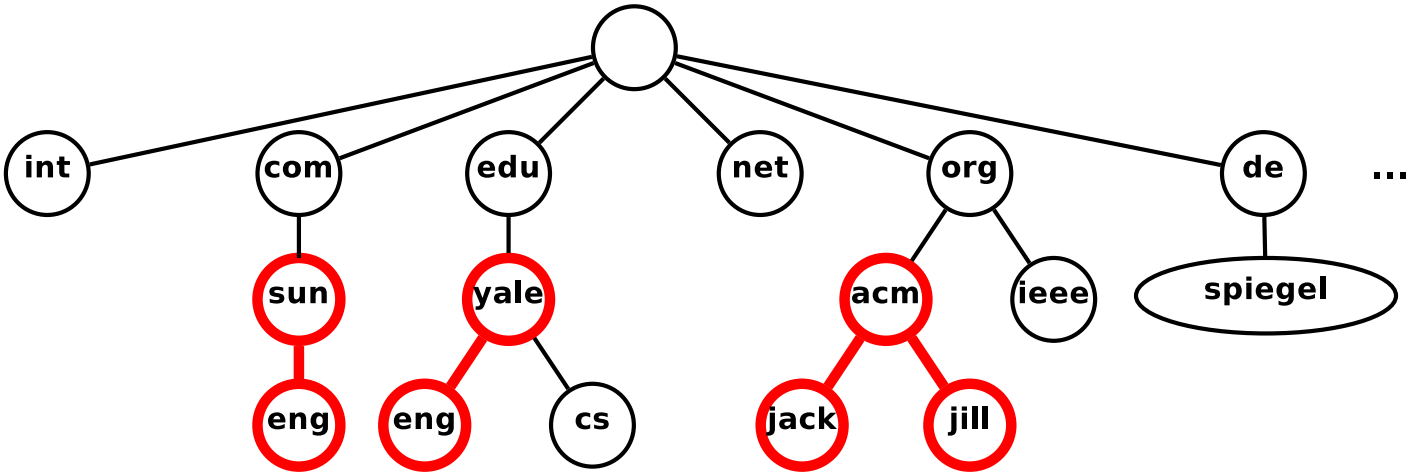
Name Server

- *autoritative* Name Server
  - verantwortlich für eine bestimmte **Zone**
  - kennen Name Server der darunterliegenden Bereiche
- *nicht-autoritative* Name Server
  - nicht dieser Zone zugeordnet sind
  - Informationen über Zone gelten als ungesichert



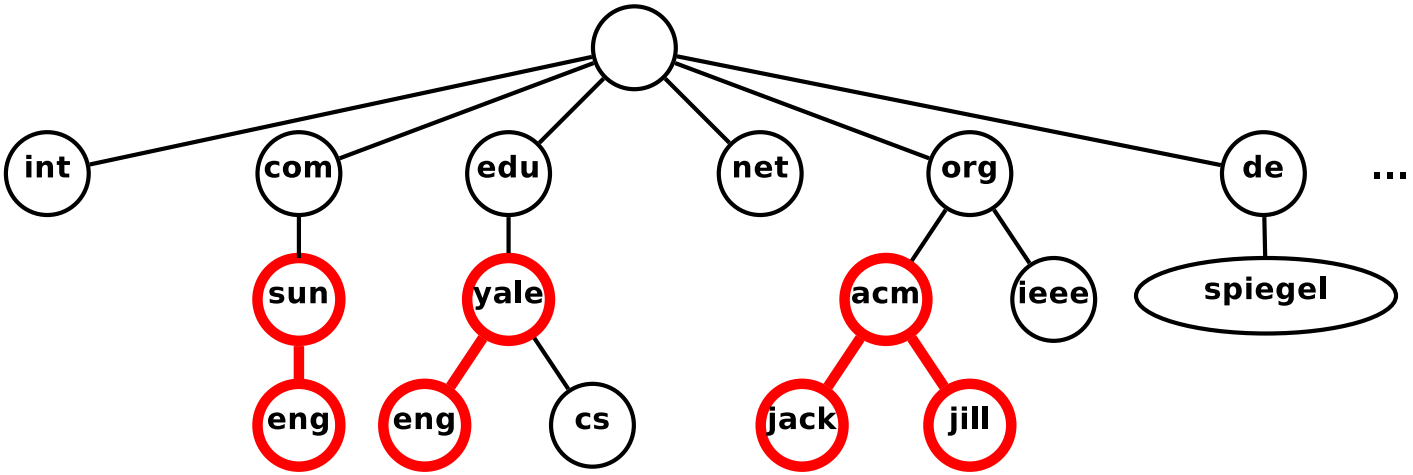
Zone

- Teile des Domain-Baumes die durch gleiche Name Server verwaltet werden, nennt man **Zone**
- für jede Zone *Primary Name Server*
- Informationen werden in der **Zonendatei** gespeichert



Zone

- Name Server werden oft als Server-Cluster angelegt
- weitere Server: *Secondary Name Server*
- Zonendateien identisch auf allen autoritativen Servern
- Synchronisation per Zonentransfer



Resolver

- Schnittstelle zwischen Name Servern und Anwendungen

Resolver

- Schnittstelle zwischen Name Servern und Anwendungen
- einfach aufgebaute Software-Module

Resolver

- Schnittstelle zwischen Name Servern und Anwendungen
- einfach aufgebaute Software-Module
- können Informationen von DNS-Servern abfragen

Resolver

- Schnittstelle zwischen Name Servern und Anwendungen
- einfach aufgebaute Software-Module
- können Informationen von DNS-Servern abfragen
- *rekursive* oder *iterative* Anfragen

Format

- **Resource Record**:

```
<NAME> [<TTL>] [<CLASS>] <TYPE> <RDATA>
```

Format

- **Resource Record**:

  `<NAME> [<TTL>] [<CLASS>] <TYPE> <RDATA>`

- **NAME** - Domainname zu dem der Resource Records gehört

Format

- **Resource Record**:

  `<NAME> [<TTL>] [<CLASS>] <TYPE> <RDATA>`

- **NAME** - Domainname zu dem der Resource Records gehört
- **TYPE** - Typ des Resource Records

Format

- **Resource Record**:

  `<NAME> [<TTL>] [<CLASS>] <TYPE> <RDATA>`

- **NAME** - Domainname zu dem der Resource Records gehört
- **TYPE** - Typ des Resource Records
- **CLASS** - Protokollgruppe des Resource Records (optional)

Format

- **Resource Record**:

  `<NAME> [<TTL>] [<CLASS>] <TYPE> <RDATA>`

- **NAME** - Domainname zu dem der Resource Records gehört
- **TYPE** - Typ des Resource Records
- **CLASS** - Protokollgruppe des Resource Records (optional)
- **TTL** - time to live (in Sekunden): Gültigkeit des Resource Records (optional)

Format

- **Resource Record**:

  `<NAME> [<TTL>] [<CLASS>] <TYPE> <RDATA>`

- **NAME** - Domainname zu dem der Resource Records gehört
- **TYPE** - Typ des Resource Records
- **CLASS** - Protokollgruppe des Resource Records (optional)
- **TTL** - time to live (in Sekunden): Gültigkeit des Resource Records (optional)
- **LENGTH** - Länge des **RDATA**-Feldes

Format

- **Resource Record**:

  `<NAME> [<TTL>] [<CLASS>] <TYPE> <RDATA>`

- **NAME** - Domainname zu dem der Resource Records gehört
- **TYPE** - Typ des Resource Records
- **CLASS** - Protokollgruppe des Resource Records (optional)
- **TTL** - time to live (in Sekunden): Gültigkeit des Resource Records (optional)
- **LENGTH** - Länge des **RDATA**-Feldes
- **RDATA** - Daten entsprechend dem Typ und der Klasse des Resource Records

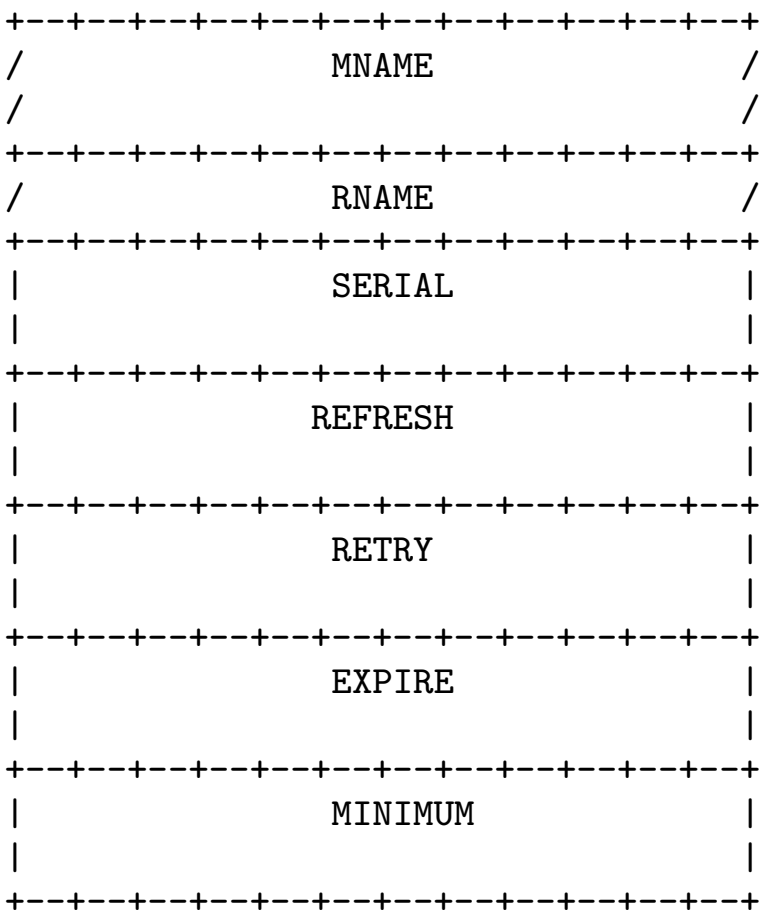
Resource Record Typen

Typ	Bedeutung	Inhalt
A	IPv4 Host Address	32-bit IP-Adresse
NS	Name Server	Name des autoritativen Servers der Domain
SOA	Start of Authority	Bestandteil der Zonendatei; enthält Angaben zur Verwaltung und Zonentransfer
CNAME	Canonical Name	Alias zu einem vorhandenen DNS-Namen
MX	Mail Exchanger	Mailserver der Domain
PTR	Pointer	Domain Name Pointer (vergleichbar mit symbolischem Links)
TXT	Text	frei definierbarer Text
AAAA	IPv6 Host Address	128-bit IP-Adresse

- Beispiele für Resource Records

NAME	TTL	CLASS	TYPE	RDATA
uni-freiburg.de.	42250	IN	A	132.230.2.100
www.spiegel.de.	36383	IN	A	195.71.11.67
whitehouse.gov.	7172	IN	MX	100 mailhub-wh2.whitehouse.gov
www.uni-freiburg.de.	1	IN	CNAME	www.ruf.uni-freiburg.de
www.ruf.uni-freiburg.de.	1	IN	CNAME	wwwneu.uni-freiburg.de
wwwneu.uni-freiburg.de.	1	IN	A	132.260.6.74
wwwneu.uni-freiburg.de.	1	IN	A	132.260.6.75

SOA RDATA Format



Domain Name des Primary Name Server

Mailadresse

Zeitabstand bis zur nächsten Nachfrage, ob Veränderungen vorliegen

Zeitabstand, in dem ein Slave Nachfrage wiederholt, wenn Master nicht antwortet

Reagiert Master nicht, deaktiviert Slave nach der angegebenen Zeitspanne die Zone

Zonendatei

- Liste von Resource Records

Zonendatei

- Liste von Resource Records
- besteht mindestens aus einem SOA- und einem NS-Resource Record

Format

Identification (16)	**Parameter** (16)
Number of Questions (16)	**Number of Answers** (16)
Number of Authority (16)	**Number of Additional** (16)
Question Section ...	
Answer Section ...	
Authority Section ...	
Additional Information Section ...	

Tabelle: Douglas E. Comer. Internetworking with TCP/IP, Seite 432

Parameter

Bit of PARAMETER field	Meaning
0	Operation: 0 Query 1 Response
1-4	Query Type: 0 Standard 1 Inverse 2 Server status request 4 Notify 5 Update
5	Set if answer authoritative
6	Set if message truncated
7	Set if recursion desired
8	Set if recursion available
9	Set if Data is authenticated
10	Set if checking is disabled
11	Reserved

Tabelle: Douglas E. Comer. Internetworking with TCP/IP, Seite 433

Parameter

Bit des PARAMETER Felds	Bedeutung
12-15	Response Type 0 No error 1 Format error in query 2 Server failure 3 Name does not exist 5 Refused 6 Name exists when it should not 7 RR set exsist 8 RR set that should exist does not 9 Server not authoritative for the zone 10 Name not contained in zone

Tabelle: Douglas E. Comer. Internetworking with TCP/IP, Seite 433

Query

- Format eines *Queries*

Query Domain Name (32)	
Query Type (16)	**Query Class** (16)

Tabelle: Douglas E. Comer. Internetworking with TCP/IP, Seite 434

Reply

- Format der Felder *Answer Section*, *Authoriy Section* und *Additional Information Section*

Resource Domain Name ...	
Type (16)	**Class** (16)
TTL (16)	**Resource Data Length** (16)
Resource Data ...	

Tabelle: Douglas E. Comer. Internetworking with TCP/IP, Seite 435

Reply

- Format der Felder *Answer Section*, *Authoriy Section* und *Additional Information Section*

Resource Domain Name ...	
Type (16)	**Class** (16)
TTL (16)	**Resource Data Length** (16)
Resource Data ...	

Tabelle: Douglas E. Comer. Internetworking with TCP/IP, Seite 435

- Queries der Anfrage in jeder Antwortnachricht mit enhalten

Namensauflösung

- Resolver schickt Query und fordert Auflösung an

Namensauflösung

- Resolver schickt Query und fordert Auflösung an
- Name Server prüft, ob angeforderte Domain in seiner Zone liegt

Namensauflösung

- Resolver schickt Query und fordert Auflösung an
- Name Server prüft, ob angeforderte Domain in seiner Zone liegt
- falls nicht-autoritativ, gibt es zwei Möglichkeiten zur Auflösung:

Namensauflösung

- Resolver schickt Query und fordert Auflösung an
- Name Server prüft, ob angeforderte Domain in seiner Zone liegt
- falls nicht-autoritativ, gibt es zwei Möglichkeiten zur Auflösung:
  - man hangelt sich von Name Server zu Name Server bis zur gewünschten Ressource (iterative Anfrage)

Namensauflösung

- Resolver schickt Query und fordert Auflösung an
- Name Server prüft, ob angeforderte Domain in seiner Zone liegt
- falls nicht-autoritativ, gibt es zwei Möglichkeiten zur Auflösung:
  - man hangelt sich von Name Server zu Name Server bis zur gewünschten Ressource (iterative Anfrage)
  - man lässt DNS-Server die Namensübersetzung machen (rekursive Anfrage)

Namensauflösung

- Resolver sendet Query und fordert entweder iterative oder rekursive Auflösung

Namensauflösung

- Resolver sendet Query und fordert entweder iterative oder rekursive Auflösung
- *iterative* Auflösung

Namensauflösung

- Resolver sendet Query und fordert entweder iterative oder rekursive Auflösung
- *iterative* Auflösung
  - Name Server übermittelt Adressen, die der Client als nächstes kontaktieren kann

Namensauflösung

- Resolver sendet Query und fordert entweder iterative oder rekursive Auflösung
- *iterative* Auflösung
  - Name Server übermittelt Adressen, die der Client als nächstes kontaktieren kann
- *rekursive* Auflösung

Namensauflösung

- Resolver sendet Query und fordert entweder iterative oder rekursive Auflösung
- *iterative* Auflösung
  - Name Server übermittelt Adressen, die der Client als nächstes kontaktieren kann
- *rekursive* Auflösung
  - Name Server übernimmt die Rolle des Resolvers und kontaktiert andere Name Server

Namensauflösung

- Resolver sendet Query und fordert entweder iterative oder rekursive Auflösung
- *iterative* Auflösung
  - Name Server übermittelt Adressen, die der Client als nächstes kontaktieren kann
- *rekursive* Auflösung
  - Name Server übernimmt die Rolle des Resolvers und kontaktiert andere Name Server
  - Name wird entweder aufgelöst oder Error zurückgegeben

Namensauflösung

- Resolver sendet Query und fordert entweder iterative oder rekursive Auflösung
- *iterative* Auflösung
  - Name Server übermittelt Adressen, die der Client als nächstes kontaktieren kann
- *rekursive* Auflösung
  - Name Server übernimmt die Rolle des Resolvers und kontaktiert andere Name Server
  - Name wird entweder aufgelöst oder Error zurückgegeben
  - rekursive Auflösung ist optional und wird nicht von allen Servern unterstützt

Namensauflösung

- Resolver sendet Query und fordert entweder iterative oder rekursive Auflösung
- *iterative* Auflösung
  - Name Server übermittelt Adressen, die der Client als nächstes kontaktieren kann
- *rekursive* Auflösung
  - Name Server übernimmt die Rolle des Resolvers und kontaktiert andere Name Server
  - Name wird entweder aufgelöst oder Error zurückgegeben
  - rekursive Auflösung ist optional und wird nicht von allen Servern unterstützt
  - meistens rekursive Auflösung

Namensauflösung

- Resolver sendet Query und fordert entweder iterative oder rekursive Auflösung
- *iterative* Auflösung
  - Name Server übermittelt Adressen, die der Client als nächstes kontaktieren kann
- *rekursive* Auflösung
  - Name Server übernimmt die Rolle des Resolvers und kontaktiert andere Name Server
  - Name wird entweder aufgelöst oder Error zurückgegeben
  - rekursive Auflösung ist optional und wird nicht von allen Servern unterstützt
  - meistens rekursive Auflösung
  - DNS-Root-Server unterstützen nur iterative Anfragen

Problem

- nicht-autoritative Name Server können Namen nur über den Namensbaum auflösen

Problem

- nicht-autoritative Name Server können Namen nur über den Namensbaum auflösen
- ständiges ablaufen des Baumes ist ineffizient:

Problem

- nicht-autoritative Name Server können Namen nur über den Namensbaum auflösen
- ständiges ablaufen des Baumes ist ineffizient:
  - Überbelastung der DNS Root Server

Problem

- nicht-autoritative Name Server können Namen nur über den Namensbaum auflösen
- ständiges ablaufen des Baumes ist ineffizient:
  - Überbelastung der DNS Root Server
  - viele Anfragen beziehen sich auf Name Server, die sich im gleichen Unterbaum befinden (Lokalität)

Lösung

- nicht-autoritative Name Server speichern daher Informationen zu Anfragen

Caching

Lösung

- nicht-autoritative Name Server speichern daher Informationen zu Anfragen

Caching

- *TTL* bestimmt Gültigkeitsdauer

Lösung

- nicht-autoritative Name Server speichern daher Informationen zu Anfragen

Caching

- *TTL* bestimmt Gültigkeitsdauer
- *TTL* wird von autoritativen Servern gesetzt

Lösung

- nicht-autoritative Name Server speichern daher Informationen zu Anfragen

Caching

- *TTL* bestimmt Gültigkeitsdauer
- *TTL* wird von autoritativen Servern gesetzt
- Caching funktioniert, da sich IP-Adressen von Server i.d.R. selten ändern

Lösung

- nicht-autoritative Name Server speichern daher Informationen zu Anfragen

Caching

- *TTL* bestimmt Gültigkeitsdauer
- *TTL* wird von autoritativen Servern gesetzt
- Caching funktioniert, da sich IP-Adressen von Server i.d.R. selten ändern
- Spezialfall: *Caching only Name Server* sind für keine Zone verantwortlich und müssen Anfragen über gespeicherte Informationen oder Namensauflösung beantworten

Negative Caching

- Name Server merkt sich Namen, die er nicht auflösen konnte

Negative Caching

- Name Server merkt sich Namen, die er nicht auflösen konnte
- kann auf Anfragen schneller antworten

- ein *"reverse lookup"* liefert zu einer IP-Adresse wenn möglich die entsprechende Domain

- ein *"reverse lookup"* liefert zu einer IP-Adresse wenn möglich die entsprechende Domain
- Suche nach der IP-Adresse im Serverbaum wäre ineffizient

- ein *"reverse lookup"* liefert zu einer IP-Adresse wenn möglich die entsprechende Domain
- Suche nach der IP-Adresse im Serverbaum wäre ineffizient
- spezielle Domain *IN-ADDR.ARPA*

- ein *"reverse lookup"* liefert zu einer IP-Adresse wenn möglich die entsprechende Domain
- Suche nach der IP-Adresse im Serverbaum wäre ineffizient
- spezielle Domain *IN-ADDR.ARPA*
  - unterhalb existieren lediglich drei Subdomain-Ebenen

- ein *"reverse lookup"* liefert zu einer IP-Adresse wenn möglich die entsprechende Domain
- Suche nach der IP-Adresse im Serverbaum wäre ineffizient
- spezielle Domain *IN-ADDR.ARPA*
  - unterhalb existieren lediglich drei Subdomain-Ebenen
  - Labels bestehen aus Zahlen zwischen 0 und 255

- ein *"reverse lookup"* liefert zu einer IP-Adresse wenn möglich die entsprechende Domain
- Suche nach der IP-Adresse im Serverbaum wäre ineffizient
- spezielle Domain *IN-ADDR.ARPA*
  - unterhalb existieren lediglich drei Subdomain-Ebenen
  - Labels bestehen aus Zahlen zwischen 0 und 255
  - jede Ebene im *IN-ADDR.ARPA*-Baum repräsentiert eine Komponente einer IP-Adresse in umgedrehter Reihenfolge

- ein *"reverse lookup"* liefert zu einer IP-Adresse wenn möglich die entsprechende Domain
- Suche nach der IP-Adresse im Serverbaum wäre ineffizient
- spezielle Domain *IN-ADDR.ARPA*
  - unterhalb existieren lediglich drei Subdomain-Ebenen
  - Labels bestehen aus Zahlen zwischen 0 und 255
  - jede Ebene im *IN-ADDR.ARPA*-Baum repräsentiert eine Komponente einer IP-Adresse in umgedrehter Reihenfolge
  - Auflösung nach einer IP-Adresse funktioniert wie im restlichen DNS

- ein *"reverse lookup"* liefert zu einer IP-Adresse wenn möglich die entsprechende Domain
- Suche nach der IP-Adresse im Serverbaum wäre ineffizient
- spezielle Domain *IN-ADDR.ARPA*
  - unterhalb existieren lediglich drei Subdomain-Ebenen
  - Labels bestehen aus Zahlen zwischen 0 und 255
  - jede Ebene im *IN-ADDR.ARPA*-Baum repräsentiert eine Komponente einer IP-Adresse in umgedrehter Reihenfolge
  - Auflösung nach einer IP-Adresse funktioniert wie im restlichen DNS
  - Einträge sind vom Resource Record Typ PTR

- ein *"reverse lookup"* liefert zu einer IP-Adresse wenn möglich die entsprechende Domain
- Suche nach der IP-Adresse im Serverbaum wäre ineffizient
- spezielle Domain *IN-ADDR.ARPA*
  - unterhalb existieren lediglich drei Subdomain-Ebenen
  - Labels bestehen aus Zahlen zwischen 0 und 255
  - jede Ebene im *IN-ADDR.ARPA*-Baum repräsentiert eine Komponente einer IP-Adresse in umgedrehter Reihenfolge
  - Auflösung nach einer IP-Adresse funktioniert wie im restlichen DNS
  - Einträge sind vom Resource Record Typ PTR
  - Auflösung nicht immer eindeutig

- Beispiele:
  - Über *100.2.230.132.in-addr.arpa.* lässt sich die Domain *uni-freiburg.de.* mit der IP-Adresse *132.230.2.100* auflösen.
    Die entsprechene Resource Record im *ARPA*-Baum ist demnach:

    ```
    100.2.230.132.in-addr.arpa.  84350  IN  PTR  uni-freiburg.de.
    ```

  - Hingegen lässt sich die Domain *www.spiegel.de.* mit der IP-Adresse *195.71.11.67* nicht auflösen.

Angriffsformen

- *Distributed Denial of Service* Angriffe auf Name Server

Angriffsformen

- *Distributed Denial of Service* Angriffe auf Name Server
  - Überlastung von Name Servern

Angriffsformen

- *Distributed Denial of Service* Angriffe auf Name Server
  - Überlastung von Name Servern
- DNS-Amplification-Angriff

Angriffsformen

- *Distributed Denial of Service* Angriffe auf Name Server
  - Überlastung von Name Servern
- DNS-Amplification-Angriff
- DNS-Spoofing

Angriffsformen

- *Distributed Denial of Service* Angriffe auf Name Server
  - Überlastung von Name Servern
- DNS-Amplification-Angriff
- DNS-Spoofing
- Cache Poising

DNS Security Extensions (DNSSEC)

- Authentizität von DNS-Nachrichten

DNS Security Extensions (DNSSEC)

- Authentizität von DNS-Nachrichten
- Datenintegrität

DNS Security Extensions (DNSSEC)

- Authentizität von DNS-Nachrichten
- Datenintegrität
- keine Verschlüsselung der DNS-Daten

DNS Security Extensions (DNSSEC)

- Authentizität von DNS-Nachrichten
- Datenintegrität
- keine Verschlüsselung der DNS-Daten
- asymmetrische Kryptosystem

DNS Security Extensions (DNSSEC)

- Authentizität von DNS-Nachrichten
- Datenintegrität
- keine Verschlüsselung der DNS-Daten
- asymmetrische Kryptosystem
  - öffentlicher Schlüssel wird in RR-Typ versendet

Ausblick und Zusammenfassung

Ausblick und Zusammenfassung

Vielen Dank für die Aufmerksamkeit!