KV ZÜRICH BUSINESS SCHOOL

DNS – Aus der Sicht des Webmasters

Diese Dokumentation wurde verfasst von:

Bernhard Kreinz

Systemspezialist / Webmaster

(Version 1.0 - Juni 1999)

Themenübersicht:

1.0 Vorwort 3
2.0 DNS – Entwicklung und Geschichte 5
3.0 DNS – Der Aufbau 7
3.1 Der Domainnamensraum und die Resource Records 8
3.2 Domains 9
3.2.1 Top Level Domain 10
3.2.2 Die IN-ADDR.APRA-Domains 10
3.2.3 Root Domain 13
3.3 Zonen 13
3.4 Nameserver 14
3.4.1 Rootserver 14
3.4.2 Top Level Domain Server 14
3.4.3 Master Name Servers 15
3.4.4 Caching Server 15
3.4.5 Forwarders 15
3.4.6 Slaves 15
3.5 Name Resolution 16
3.5.1 Recursive Queries and Iterative Queries 16
4.0 Mailrouting und das DNS 17
4.1 MX Records ☺☺☺ 17
4.2 „Offizielle“ bzw. „bereinigte“ Mailadressen 20
4.3 Mail-Gateway-Routing 20
4.4 Spezifikation eines „Fallback“-Mailhosts 21
4.5 Fehlerquellen 22
4.5.1 Alias-Namen in MX-Records 22
4.5.2 Verkettete MX Records 22
4.5.3 Mail Exchanger mit Default-Mailhost 23
5.0 Implementation eines DNS Servers als Workshop 23
5.1 Konzeptionelle Überlegungen 24
5.1.1 Wie sieht unsere Infrastruktur aus ? 24
5.1.2 Wie sehen die Anforderungen aus ? 24
5.1.3 Konsequenzen aus oben genannten Punkten 25
5.2 Konfiguration und Installation eines DNS Servers 25
5.3 Konfiguration eines Client 25
6.0 Der URL 25
6.1 Uniform Resource Locator (URL) ☺☺☺ 25
6.2 URL-Schema bei »http« 26
6.3 URL-Schema bei »mailto« 27
6.4 URL-Schema bei »news« 28
7.0 Tools rund um DNS 28
7.1 NSLOOKUP 29

1.0 Vorwort

Dieser kleine Workshop soll dir Gelegenheit geben, dein Verständnis für das Namenskonzept im Internet zu vertiefen. Was steckt den hinter einer Internetadresse oder URL? Wie findet mein Mail den Absender teacher@barnes.ch ?

Spätestens nach diesem Workshop musst du die Funktionsweise dieses Vorganges kennen. Damit dir alles besser im Hinterkopf hängen bleibt, werden wir im „produktespezifischen" Teil des Workshops den DNS Server von Microsoft installieren.

Vielleicht noch ein paar Worte zu meiner Person ... Nein, die spar ich mir. Das kann man unter http://www.barnes.ch nachlesen. Was aber sicher noch ansteht ist eine Erläuterung zur Taxonomie in Bezug auf das SIZ Zertifikat, falls Du ein solches anstrebst:

Legende:

☺ Jeder Smiley steht für für eine Anforderungsstufe[1] .
💣 Oh! ... Oh ! Jetzt wird s aber spannend. Ich glaube, da steckt ein Tip dahinter!
(Dieser Tip ist fakultativ, muß nicht offiziell gewusst werden. Er kann die Arbeit aber wesentlich erleichtern .... Muß ich noch mehr sagen ?)
© Hier handelt es sich um Originaltexte, welche jeweils in einer Fusszeile referenziert werden. Es kann sein, daß diese in Originalform wiedergegeben werden. Computerenglisch wird als Voraussetzung in diesem Lehrgang angesehen.

Zusammenfassend lässt sich vorwegnehmen, daß der Stoff, den du hier vermittelt bekommst, zu deinen Kernkompetenzen zählt.

Du wirst sehen, daß der Workshop sich in zwei grundsätzliche Kategorien aufteilen lässt:

a) Produktespezifischer Workshop
b) Produkteübergreifendes Verständnis und Know-how

Dieses Dokument wird daher sehr allgemein gehalten und soll als Leitfaden dienen für deine weitere Tätigkeit . Wir wollen die Aspekte in diesem Dokument vorhanden wissen, welche für alle Produkte (in diesem Themenbereich) gelten. Wie soll ich denn wissen auf welchem DNS Server du arbeiten willst oder musst? Handelt es sich um eine BSD Implementation oder arbeitest du mit dem DNS Dienst von Windows NT? So werden wir uns an den Standards orientieren, welche von verschiedenen Gremien IEEE, W3C, IAB usw. zu den jeweiligen Themengebieten vorgegeben werden, dies meint deren RFC's.

[1] 1= Wissen, 2=Verständnis, 3=Anwendung, 4=Analyse, 5=Synthese, 6=Beurteilung

Näheres dazu aber im nächsten Kapitel.

Jetzt beginnen wir aber ....
Und zwar mit einem kleinen historischen Rückblick auf die Entstehung des DNS. Wenn man die Anforderungen benennt, welche zu diesem System geführt haben, dann wird man auch die Funktionsweise einfacher begreifen. Ob im WWW oder beim Mail, beide stellen Anforderungen an ein Namenskonzept bzw. an ein Adressierungskonzept und DNS hat sich als ein Standard durchgesetzt.

Bernie Kreinz

2.0 DNS – Entwicklung und Geschichte

Nomenklatur:

☺☺☺ DNS = Domain Name System

Kleine Geschichte:

Am Anfang war das hosts – File [2] ...
Tatsächlich könnte so die Entstehung des Namensauflösungskonzeptes beschrieben werden. Schon von Beginn an wurde ein zentrales Dokument verwaltet[3], welche auf andere Hosts per ftp periodisch kopiert wurde. Dies wurde im ursprünglichen Verbund von 14 US-Universitäten (Der Geburt des Internet) so gehandhabt und war (in dieser Grösse) noch administrierbar.

☺☺☺ Die Struktur des Hosts File sieht grundsätzlich folgendermaßen aus:

********** Beginn hosts **********

Mailserver # Kommentar zum Eintrag
Webserver- kvz 10.0.1.27

********** ende hosts **********

☺☺☺ Mit anderen Worten .... was hier geschieht ist eine reine Zuordnung von Namen zu bestimmten Adressen. Es liegt in der Natur (Kultur) des Menschen, daß er eine auf Namen basierende Adressierung einer reinen Numerischen Adressierung bevorzugt. Aus den Telematikgrundlagen aber wissen wir, daß die effektive Kommunikation (IP basierend) über die „gemappte“[4] IP-Adresse vollzogen wird und nicht über den verwendeten Namen !

Wenn man sich einerseits die Struktur dieser hosts Datei ansieht, sowie noch Überlegungen anstellt in Bezug auf das Wachstum der Infrastruktur im Internet, so wird einem schnell klar, daß folgende Punkte ein neues, flexibleres System notwendig gemacht haben ☺☺☺ :

- Schwer und träge verwaltbares System
- Mangelnde Transparenz
- Namenskonflikt
- Last auf NIC Computern
- usw.

[2] In Windows NT befindet sich die Datei standardmässig in %system32%\drivers\etc\ . Bei einem Unix ist sie standardmässig im /etc Verzeichnis.
[3] Dies war schon damals die Aufgabe der NIC (Network Information Center). Als Grundlage dafür diente die hosts Datei. Der Form nach war es eine ASCII Datei. Entsprechende Diskussionen über Grundfragen der Namensgebung wurden schon im Herbst 1971 heftig diskutiert. (RFC 226)
[4] Zugeordnete, zugewiesene

Dr. Paul Mockapetris, Vater von DNS stellte folgende Anforderungen in das Domain Name System (DNS) ☺☺☺

- Einfache, verteilte Datenbank als Grundlage – eine Art „Directory Service"
- Hierarchischer Namensraum
- Dezentrale Administration muß möglich sein
- Datentypen müssen erweiterbar sein
- Die Datenbankgrösse darf keiner physischen Limitierung ausgesetzt sein
- Das Ganze soll ausserdem eine „vernünftige" Performance an den Tag legen

☺ ☺ ☺ Directory Service:
Die Aufgabe eines Directory Service ist das Verwalten von Informationen über Objekte. Als zentrale Komponente dieses Systems (DNS) wird die hierarchische Strukturierung des Domainnamensraums betrachtet, welche die dezentrale Verwaltung von Informationen ermöglicht.

1983 war es dann soweit und das DNS war geboren. Die immer noch als BIND Implementation bekannte Technologie wurde mittlerweile von anderen Softwareherstellern adaptiert und in zwar kompatibler, aber anders zu administrierenden Form abgeändert. Wie du ja weißt, sind die Userinterfaces der verschiedenen OS unterschiedliche Möglichkeiten offen lassen.

Seit 1993 war die NIC[5] das Mass der Dinge, wenn es um neue Top Level Domains geht. Als Verwalter der Rootdomain war und ist das Nervenzentrum der Adressierung in der Hand der Rootdomain.

Und wie sieht s jetzt aus ? (Was ist los mit www.internic.net ?)

☺ ☺ ☺ ICANN
The Internet Corporation for Assigned Names and Numbers

About ICANN

The Internet Corporation for Assigned Names and Numbers (ICANN) is the new non-profit corporation that was formed to take over responsibility for the IP address space allocation, protocol parameter assignment, domain name system management, and root server system management functions now performed under U.S. Government contract by IANA and other entities.
The Board of ICANN will be composed of nineteen Directors, nine At-Large Directors, nine to be nominated by Supporting Organizations, and the President/CEO (ex officio). The nine At-Large Directors of the Initial Board are serving one-year terms and will be succeeded by At-Large Directors elected by an at-large membership organization.

☺☺☺ Wie du siehst ist auch auf diesem Gebiet einiges los. Darum spare ich mir die vielen Worte – das Wesentliche ist gesagt und Literatur zur Geschichte des DNS gibt es zu Hauf ... auch im Internet. So sei an dieser Stelle schon einmal vorgemerkt, dass das Internet auch hier das beste und aktuellste Informationspotential hat !

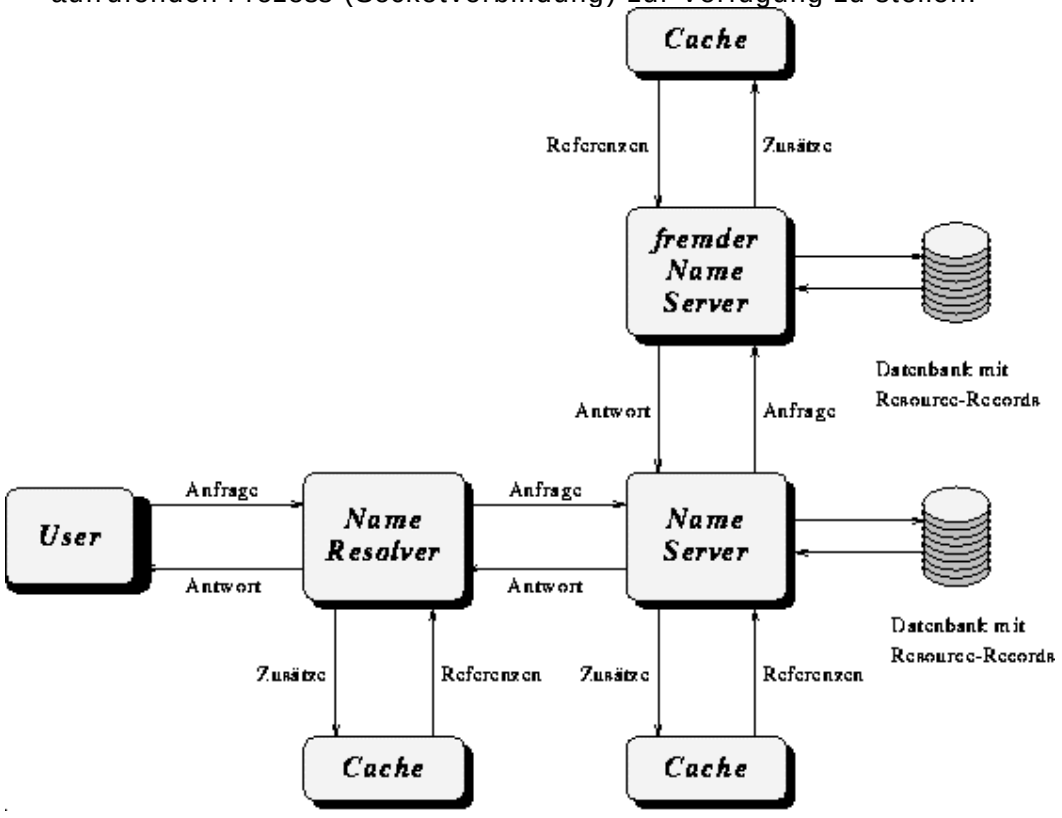
[5] Network Solutions, Inc. Verwalter der Root Domain nach Absprache mit der US Regierung.

3.0 DNS – Der Aufbau

In diesem dritten Kapitel werden wir uns mit den für das Verständnis wichtigen Begriffen auseinandersetzten. Darum wird das Kapitel als Nachschlagewerk betrachtet. Für eine effektives Verständnis ist das Wissen um die Definition der Begriffe unumgänglich. Wir werden die Begriffe kurz beschreiben, damit wir im Anschluss wissen (oder wenigstens ahnen) worum es konzeptionell und technisch geht.

☺☺☺ Das DNS besteht aus drei Hauptkomponenten /RFC 1034/:

- Dem **Domainnamensraum** (Domain Name Space) mit den entsprechenden **Resource Records** (RR) Die RR sind Datensätze, die die Objekte des Namensraums beschreiben.
- Den **Nameservern** (NS): Diese sind Programme, die über Informationen eines Teils des Namensraums verfügen. Wenn ein NS komplette Informationen über einen Teil des Namensraums besitzt, kann er als Authority für diesen Namensraum bezeichnet werden (im Gegensatz zum Caching-Only- Server, der über keine "authoritative" Informationen verfügt, siehe 2.3 auf Seite 7).
- Den **Resolvern**: Sie stellen die aufrufbare Schnittstelle für die Kommunikation zwischen einem Benutzerprozess und einem NS dar. Sie sind in der Lage, Informationen aus Nameservern zu extrahieren und dem aufrufenden Prozess (Socketverbindung) zur Verfügung zu stellen.



3.1 Der Domainnamensraum und die Resource Records

☺☺☺ Das Domain Name System (DNS) ermöglicht die hierarchische Strukturierung eines Namensraums. Der Namensraum wird als Baum mit unterschiedlicher "Tiefe" dargestellt; die Blätter und Knoten des Baums werden als "Label" bezeichnet. Der volle Domainnamen eines Objekts des Raums besteht aus der Verkettung aller Label auf dem Pfad vom Objekt zu der Wurzel des Baums. Die einzelnen Label werden dabei durch Punkte getrennt. Label sind Zeichenketten mit einer Länge von 0--63 ASCII-Zeichen. Sie müssen mit einem alphabetischen Zeichen anfangen. (Diese Vorschrift wurde bereits gelockert; Label dürfen mit numerischen Zeichen beginnen, aber nicht nur aus numerischen Zeichen bestehen, um mögliche Konflikte mit IP-Adressen zu vermeiden. /RFC1123/,/RFC1101/) Der Label mit der Länge 0 ist für die Wurzel des Baums (root) reserviert, d.h. der komplette Name eines Objekts muss mit einem Punkt abgeschlossen werden. Solche Namen werden als absolute Domainnamen bezeichnet. Die gesamte Länge eines Namens sollte 255-n nicht überschreiten (n ist die Anzahl der Label des vollen Domainnamens).

Die Daten, die ein Objekt im DNS beschreiben, werden als ein Satz von Resource Records (RR) dargestellt. Es gibt mehrere RR-Typen, die unterschiedliche Informationen über ein Objekt enthalten.
So wird z.B. im RR vom Typ A die Zuordnung Objektname - Adresse festgehalten. RR des Typs HINFO enthalten hostspezifische Informationen wie Hardware und Betriebssystem.

Hier nun die Datenfelder eines RR:
(Eine ausführliche Beschreibung der RR-Formate findet man in /RFC 1033-35/ und den Domain-Administrators Guides verschiedener Softwarehersteller.)

Inhalt eines Resource Record:

name > Der Domainname des Objekts zu dem der RR gehört.
class > Protokollgruppe (IN = Internet, CH = Chaosnet, HS = Hesiod).
type > RR-Typ.
ttl > time to live (in Sekunden); Zeit wie lang dieser RR gültig ist und gecached werden darf.
rdata > Daten, die das Objekt beschreiben, zu dem dieses RR gehört.

Die Datenformate sind abhängig vom RR-Typ (s.u.).

☺☺☺ Hier einige RR-Typen, die häufigsten in der NS-Konfiguration verwendet werden: (alle anderen RR-Typen, insbesondere WKS, MB, MG, MINFO und MR Resource Records werden in der Regel nicht verwendet.)

RR	Funktionalität	RDATA-Feld
A	Die Adresse eines Hosts	32-bit IP-Adresse
CNAME	Definition eines Aliasnamen zu einem Canonical Name	Domainname(Canonical Name)
HINFO	Host-Info wie Typ und	CPU und Betriebssystem

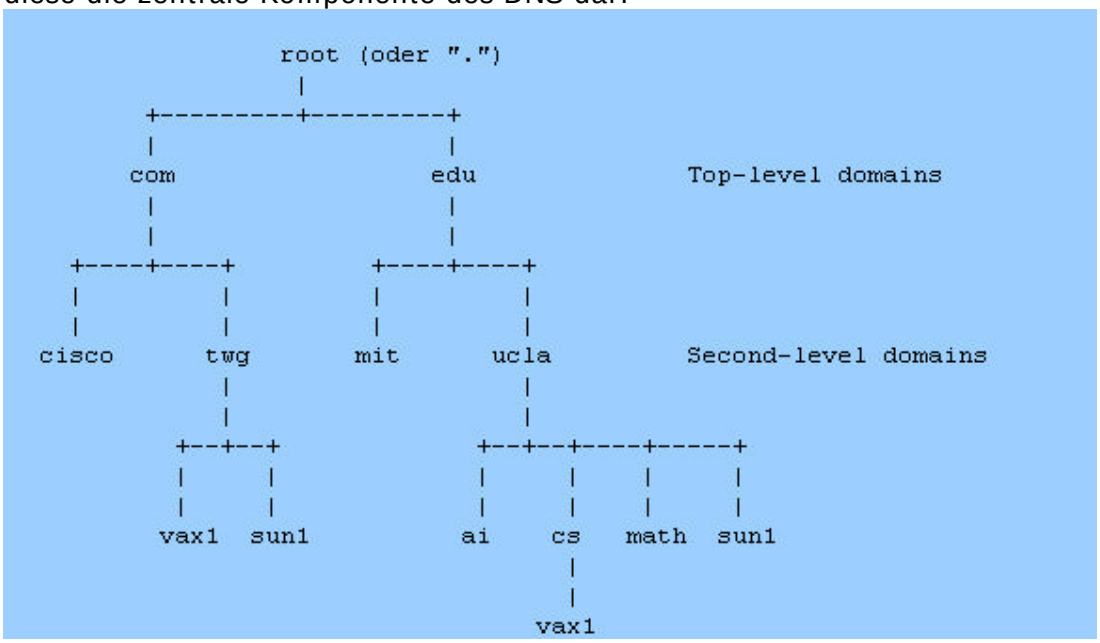
	Betriebssystem	CPU und Betriebssystem
MX	Mail-Exchange	16-bit preference, Name des Mailhost
NS	Ein authoritativer Nameserver	Hostname
PTR	Zeiger (pointer) zu einem Domainnamen	Domainname
SOA	Definiert "Start Of Authority" für eine Zone	mehrere Felder (NS-Name, Fehler-Mailbox, Serial-Nr. der Zonendaten, mehrere Timer)

Beispiele:

rusvx2.rus.uni-stuttgart.de.	**86400**	**IN**	**A**	**129.69.1.2**
rusvx2.rus.uni-stuttgart.de.		**IN**	**HINFO**	**VAX8810 VMS-5.1**
2.1.69.129.in-addr.arpa.	**86400**	**IN**	**PTR**	**rusvx2.rus.uni-stuttgart.de.**
ibmvm.rus.uni-stuttgart.de.		**IN**	**CNAME**	**rusvm1.rus.uni-stuttgart.de.**

3.2 Domains

☺☺☺ Ziel des DNS ist es, die dezentrale Verwaltung von Informationen über Objekte der Datenkommunikation zu ermöglichen. Diese dezentrale Verwaltung der Informationen, sowie die Eindeutigkeit der Objektnamen kann nur durch eine hierarchische Strukturierung des Namensraums erreicht werden; somit stellt diese die zentrale Komponente des DNS dar.



☺ ☺ ☺ Eine Domain ist ein kompletter Ast dieser baumartigen Verzeichnis - Struktur.

Es ist jetzt leicht, Äste des Baumes einzelnen Administratoren zuzuordnen. Der Administrator kann, nach Bedarf, die Kontrolle über Teile seines Astes an eine

andere Person delegieren usw. Man bezeichnet eine Domain abzüglich aller delegierten Äste als Zone. Der Administrator einer Zone ist zur Verwaltung aller Namen in seinem Zonenbereich bevollmächtigt und managet somit eine Subdomain.

Ein Beispiel:
Der Administrator der Domain ucla.edu kann die Subdomain ai.ucla.edu an einen anderen vergeben, dadurch entstehen zwei Zonen für die Domain ucla.edu: die erste enthält ucla.edu mit allen Subdomains ausser ai.ucla.edu, die zweite besteht nur aus der Domain ai.ucla.edu. Durch die Baumstruktur des Namensraums ist die Eindeutigkeit der Knotennamen weltweit gewährleistet. Der Datenadministrator einer Domain muss dafür sorgen, dass in seiner Domain die Namen eindeutig sind. Der Name sun1.twg.com und sun1.ucla.edu sind weltweit eindeutig.

☺☺☺ Domains sind administrative Gebilde, die eine dezentralisierte Verwaltung von Namen erlauben. Die Struktur einer Domain sollte so gestaltet werden, dass sie möglichst die Struktur der kontrollieren- den Organisation reflektiert: die Namen werden mit unterschiedlicher „Tiefe“ gewählt, abhängig von der Grösse und Komplexität einer Organisation.
Beispielsweise kann eine Universität ihren Namensraum so unterteilen, dass dieser der Strukturierung der Universität in Fakultäten bzw. Instituten entspricht.

3.2.1 Top Level Domain

Eine administrative Entscheidung /RFC 920/ hat die Top-Level Domains festgelegt:

☺☺☺

➔ ISO-Länder-Codes (z.B. CH für die Schweiz)
➔ und die folgenden organisatorischen Kategorien (sogenannte "generic top level domains") unterteilt:

->MIL > US-Militär
->GOV > US-Regierung
->EDU > Bildungswesen
->COM > Kommerzielle Einrichtungen
->NET > Netzwerk- und Netzwerk-Management-Organisationen
->ORG > Andere ``non-profit'' Organisationen
->ARPA > Wird nicht mehr als Teil von Rechnernamen verwendet. INT > Internationale Organisationen. Wurde später eingeführt; wird kaum verwendet.

3.2.2 Die IN-ADDR.APRA-Domains

Das Auffinden der Adresse eines Knotens im Internetnamensraum geschieht durch eine gezielte Durchsuchung des hierarchischen Namens"baums" und ist verhältnismäßig leicht.
Umgekehrt ist das Auffinden des Namens eines Knotens, dessen Adresse bekannt ist, im Prinzip nicht möglich, ohne dass der ganze Baum durchsucht wird.

Um Letzteres zu erleichtern, wurde eine Domain eingeführt, welche Adressen als Teil eines Namens benutzt, der wiederum zu dem Domainnamen des gesuchten Rechner zeigt. Die Domain, mit der das "Adresse zum Namen''- Mapping

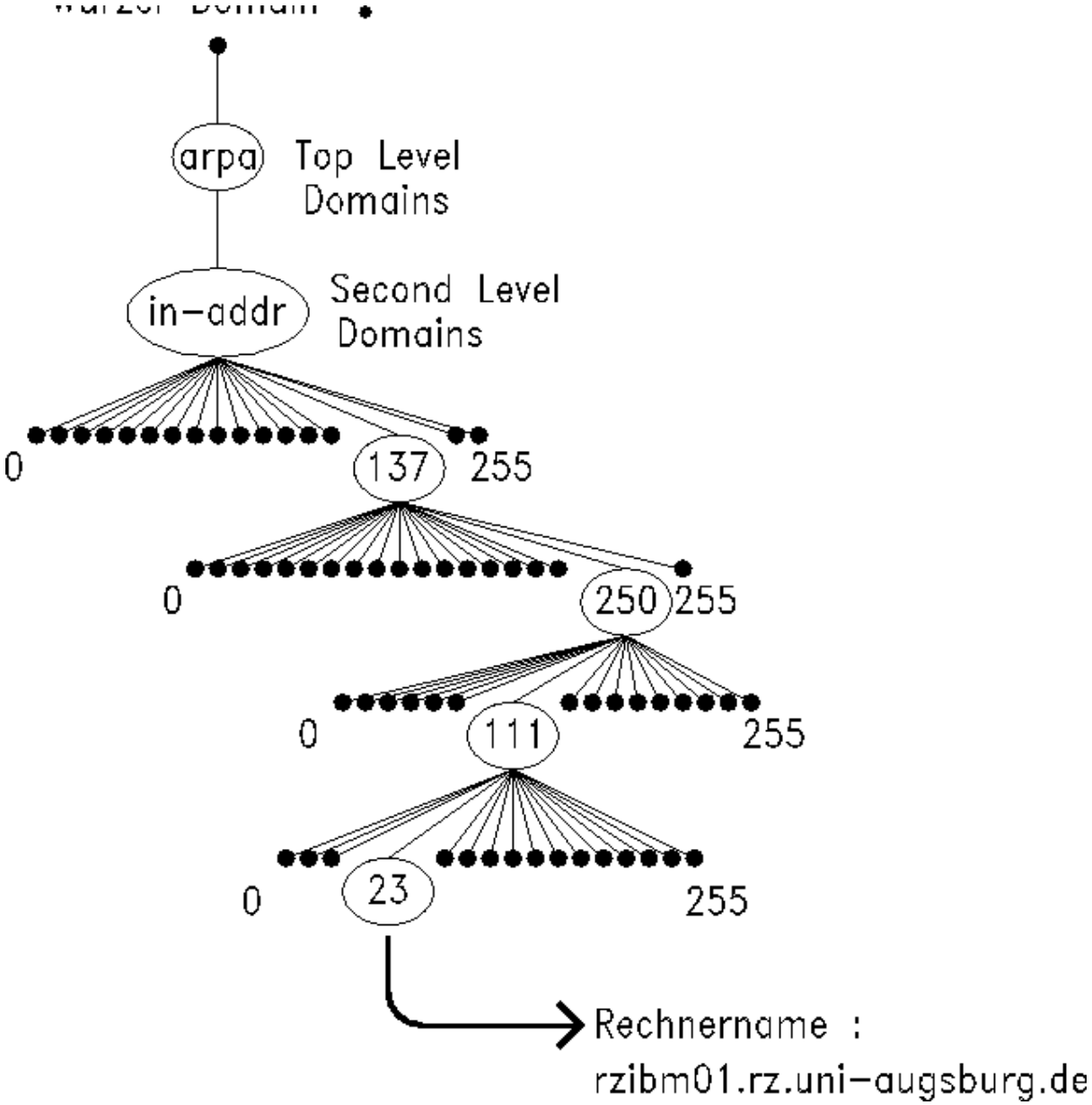
verwirklicht wird, heisst IN-ADDR.ARPA. Namen in dieser Domain dienen als eine Art invertiertes Register, das die Zuordnung der Adresse zum Namen ermöglicht. Diese Namen werden so dargestellt:

```
*.IN-ADDR.ARPA
```

wobei * eine IP-Adresse repräsentiert, in der die Reihenfolge der Bytes umgedreht wurde. Solche Namen werden als Zeiger nach gültigen Namen des Internetnamensraums verwendet. Letzteres wird mit Hilfe der sogenannten Pointer(PTR)-RRs realisiert. z.B. im folgenden RR zeigt der IN-ADDR.ARPA-Name eines Rechners mit der IP-Adresse 129.69.1.31 zu seinem gültigen Internet-Namen (canonical name):

```
31.1.69.129.IN-ADDR.ARPA.   IN      PTR rusvm1.rus.uni-stuttgart.de.
```

Schematische Darstellung des Konzeptes:



In diesem Beispiel wird der Rechnername zur IP-Adresse 137.250.111.23 gesucht. Die Antwort muß also in der 'in-addr.arpa' Domäne stehen. Jetzt wird klar, warum die Adresse in umgekehrter Reihenfolge in der Zone gespeichert steht. Man beginnt mit der Suche am Domänenteil der Adresse, damit auch hier wie bei den Namen die Daten auf Zonen verteilt werden können.

Im allgemeinen sind die NS für die obersten IN-ADDR.ARPA-Domains den Root-Servern bekannt.

Kontaktadresse:
IN-ADDR.ARPA Government Systems, Inc. Attn: Network Information Center 14200 Park Meadow Drive Suite 200 Chantilly, VA 22021 Tel.: +1-800-365-3642 u. +1-703-802-4535 Fax: +1-703-802-8376 hostmaster@nic.ddn.mil oder registrar@nic.ddn.mil

3.2.3 Root Domain

☺☺☺ Wird zur Zeit von der www.icann.org verwaltet.
(Zentrale Registraturstelle für IP Adressierung und Top Level DNS)
Für die Topleveldomains stehen insgesamt 12 Root DNS Server in Betrieb:

Auszug mittels dem Tool: nslookup

A.ROOT-SERVERS.NET	internet address = 198.41.0.4
H.ROOT-SERVERS.NET	internet address = 128.63.2.53
B.ROOT-SERVERS.NET	internet address = 128.9.0.107
C.ROOT-SERVERS.NET	internet address = 192.33.4.12
D.ROOT-SERVERS.NET	internet address = 128.8.10.90
E.ROOT-SERVERS.NET	internet address = 192.203.230.10
I.ROOT-SERVERS.NET	internet address = 192.36.148.17
F.ROOT-SERVERS.NET	internet address = 192.5.5.241
G.ROOT-SERVERS.NET	internet address = 192.112.36.4
J.GTLD-SERVERS.NET	internet address = 198.41.0.21
K.GTLD-SERVERS.NET	internet address = 195.8.99.11
F.GTLD-SERVERS.NET	internet address = 207.159.77.18

3.3 Zonen

☺ ☺ ☺ Ein Zone ist, wie bereits vorweggenommen, ein in sich geschlossener Namensraum und ist somit auch eine Teilmenge einer Domain.

Damit ist es möglich, trotz dezentraler Verwaltung, weltweit eindeutige Namen zu definieren und nicht zuletzt auch zu administrieren.

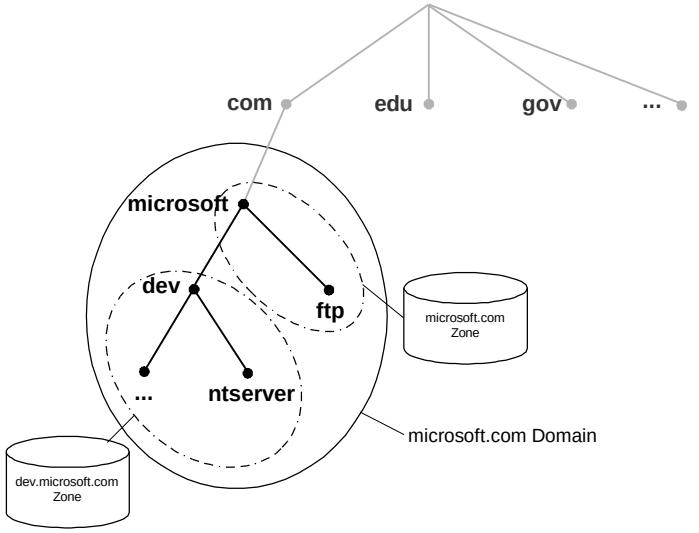
Beispiel:

Als Administrator der Zone „klasse1.kvz-schule.ch“ ist es mir möglich (innerhalb des mir zugeteilten IP Address Ranges) Namen wie folgt zu generieren.

- Schüler1.klasse1.kvz-schule.ch
- Schüler2.klasse1.kvz-schule.ch
- Und Lehrer.klasse1.kvz-schule.ch

... wobei hinter jedem Namen eine IP Adresse definiert wird. Nie vergessen ! Kommunikation über TCP/IP wird immer mittels der IP Adresse vollzogen und nicht mit den DNS Namen – dieser muss zuerst aufgelöst werden ! Dazu aber später mehr !

Als kleiner Anhang vielleicht noch ein kleines Beispiel:[6]



... wollen wir es mal kurz miteinander diskutieren ?

3.4 Nameserver

Ein Nameserver ist ein Programm, das über Informationen eines Teils des Namensraums verfügt und in der Lage ist, Fragen (Queries) über diese Informationen zu beantworten. (Das Format dieser Anfragen und der dazugehörenden Antworten sind in /RFC 1035/ beschrieben.)

☺☺☺ Es gibt verschiedene Kategorien von NS:

3.4.1 Rootserver

Sie sind die obersten Authorities. Sie verfügen über komplette Informationen für alle Internet Top-Level Domains und sie sind in der Lage, den zuständigen NS für eine beliebige Subdomain zu ermitteln. (Das Leben ist aber etwas komplizierter; mindestens bis August 1990 enthielten die Root- und Top-Level Server nur Informationen über NS, die zuständig für "connected"- Netze waren, d.h. Informationen von Objekten in nicht "connected"- Netzen waren/sind entweder gar nicht oder über Umwege erreichbar,

3.4.2 Top Level Domain Server

Sie kennen alle NS für die Second-Level Domains in ihrer Zone. Technisch wird eine Zone als ein zusammenhängender Teil des Domainnamensraums betrachtet, welches durch eine Datei mit den RRs der Objekte der Zone realisiert wird. Diese Datei muss dann mit einem SOA-Record beginnen.)

[6] Schema ist aus dem Whitepaper zum DNS Server von Microsoft entnommen.

3.4.3 Master Name Servers

Sie sind immer ``authoritative'' für eine oder mehrere Zonen.
Es gibt:

Primary Nameserver
Und
Secondary Nameserver

Die Primary-NS laden ihre Zonendaten von einer Datei, während die Secondary-NS vom Primary-NS ihre Daten erhalten, welche (optional) in eigenen Backup-Dateien abgespeichert werden.
Jedesmal, wenn ein Secondary-NS bootet, lädt er seine Daten aus der Backup-Datei (falls vorhanden) und überprüft durch den Vergleich der Serial-Nr. des SOA-RRs des Primary mit seiner eigenen die Aktualität der Daten. Um die Probleme zu vermeiden, die mit Ausfall oder Überlastung eines Nameservers verbunden sind, ist erforderlich, dass jede Zone des Domainnamensraums neben dem zuständigen Primary Nameserver einen oder besser mehrere Secondary Nameserver betreibt.

3.4.4 Caching Server

Alle Nameserver sind Caching-Server, d.h. sie speichern die aus anderen NS erworbenen Kenntnisse über den Domainnamensraum. Diese Daten bleiben solange erhalten, wie das ttl-Feld (time to live) in dem entsprechenden RR angibt /RFC 1034/.
Ein Caching-Only-Server ist ein Nameserver ohne Authority über eine Domain und somit ohne eigenen Daten. Er beantwortet Anfragen mit Informationen aus seinem Cache oder durch Fragen von ``authoritative'' Nameservern. Seine Antworten sind dann als ``non-authoritative'' bezeichnet. Caching-Server bieten somit einen Mechanismus zur Entlastung anderen zentralen NS an.

3.4.5 Forwarders

Sie sind NS mit voller Internet-Konnektivität, d.h. sie können problemlos Informationen von den Root-Servern oder anderen Master-Servern holen. Sie werden von anderen Nameservern zur ``Erledigung'' von Abfragen benutzt, die diese nicht beantworten können oder ``wollen''. Dadurch kann der Forwarder in kurzer Zeit ein grosses Cache bilden, was die Antwortzeit in vielen Fällen erheblich verkürzt. Ein Nameserver sollte einen Forwarder-Server benutzen, wenn die Verbindung zu den Root-Nameservern fehlt oder wenn eine überlastete USA-Leitung von zusätzlichem NS-Verkehr entlastet werden soll.

3.4.6 Slaves

Ein Slave-Server hat feste Vorgaben, welche anderen NS er für Queries, die er nicht selber beantworten kann, befragen darf. Diese sind in seiner Konfiguration als Forwarder eingetragen. Ein Slave wird niemals die Root-Server oder andere ihm übergeordnete NS direkt fragen, sondern nur die als Forwarder spezifizierten

NS. Ein Slave-Server kann durchaus Master-Server für eine oder mehrere Zonen sein.

3.5 Name Resolution

☺☺☺ Resolver sind Programme, die in der Lage sind, Informationen aus Nameservern abzufragen. Sie stellen eine aufrufbare Schnittstelle zum NS-Dienst dar. Der Resolver befindet sich auf dem gleichen Rechner wie das Programm, welches seine Dienste benötigt, muss aber meist Name- server fragen, die sich auf entfernten Rechnern befinden. Resolver können eigenständige Programme oder sie können an die Applikation angebunden sein. Sie ersetzen den Applikationen daher den Zugriff auf die Hostdateien.

3.5.1 Recursive Queries and Iterative Queries

Es gibt prinzipiell zwei Abfragestrategien im DNS: Man kann die NS Top-Down fragen bis die gesuchte Antwort gefunden ist, oder man kann einen Nameserver beauftragen, diese Aufgabe zu erledigen. Resolver verwenden eben diese zwei Methoden um Nameresolution zu erreichen. Dafür brauchen sie die IP-Adresse von mindestens einem NS, der in der Lage ist, auch nicht-lokale Fragen zu beantworten. Der Resolver kann nun seine Frage stellen mit gleichzeitigem Verlangen von "recursion" oder "no recursion". Im ersten Fall muss der gefragte NS andere NS kontaktieren, bis er die Frage des Resolvers vollständig beantwortet hat. Dann schickt er die Antwort an den Resolver zurück. Im zweiten Fall gibt der gefragte NS eine Liste von Nameservern zurück, die diese Fragen beantworten könnten. Der Resolver ist dann für die weitere Namensresolution verantwortlich.
Die üblichen Resolver Implementierungen reflektieren die oben geschilderten Abfragestrategien. Es gibt sogenannte "Stub"-Resolver, die nur die rekursive Methode verwenden. Die "Full"-Resolver können auf beide Methoden zurückgreifen. Die rekursive Methode wird nur dann verwendet, wenn sowohl der Resolver wie auch der NS sich auf ihre Verwendung geeinigt haben. Resolver, die Teil einer Applikation sind, sind in der Regel "Stub"-Resolver. Dies vereinfacht die Implementierung (alles erledigt der Nameserver).
In manchen Implementierungen ist noch eine wichtige Funktion des Resolvers vorhanden: Die erworbenen Informationen werden durch Caching ttl-Sekunde lang aufgehoben, was die Netzwerk- und Nameserverbelastung erheblich reduzieren kann. Resolver, die nicht eigenständige Prozesse sind, sondern Teil einer Applikation, cachen die erworbene Kenntnisse nicht, sodass die Informationen bei Bedarf neu geholt werden müssen.
Ein bekanntes Programm, welches interaktiv aufrufbare Resolver ohne Caching-Funktion benötigt ist das Programm nslookup.

4.0 Mailrouting und das DNS [7]

Das Internet Simple Mail Transfer Protocol (SMTP) stellt eine Ende-zu- Ende-Verbindung auf der Basis von TCP/IP zwischen sendendem umd empfangendem Mailsystem her. Eine solche Verbindung garantiert, dass die zu verschickende Nachricht solange auf dem Absender-Rechner verweilt, bis sie erfolgreich auf den Empfänger-Rechner übertragen werden kann.

☺☺☺ Zur Adressierung der Empfänger-Rechner werden DNS-konforme Domainnamen verwendet. Eine Mailadresse besteht generell aus einem lokalen Teil (Mailbox, User-ID) und einem Domain-Teil (Domainname des Rechners):

Zum Beispiel: postmaster@noc.belwue.de

Die Aufgabe eines Mailsystems ist es, anhand der syntaktischen Analyse einer Zieladresse zu entscheiden, auf welche Weise die dazugehörige Nachricht zu verschicken ist. Hier kann man grob zwischen drei Fällen unterscheiden:

- 💣💣 ist identisch mit dem Domainnamen des sendenden Rechners. Die Nachricht wird lokal ausgeliefert. Es wird das Netzwerk überhaupt nicht in Anspruch genommen.
- 💣💣 ist ein Domainname eines Rechners, dessen IP-Adresse aus der eigenen Host-Tabelle entnommen werden kann. Danach wird eine Verbindung zum SMTP-Port des Zielrechners hergestellt.
- ☺☺☺ ist dem sendenden Rechner zunächst unbekannt und das Mailsystem muss die IP-Adresse des Zielrechners über eine Nameserverabfrage auflösen. Danach wird eine Verbindung zum SMTP-Port des Zielrechners hergestellt.

Im letzten Fall wird deutlich, wie stark erfolgreiches Mail-Routing von einem zuverlässig betriebenen DNS abhängt. Ein Nameserver, der unvollständige bzw. falsche Informationen verbreitet oder erst gar nicht in Betrieb ist, kann die Ursache dafür sein, wenn E-Mail nicht an der Zieladresse ankommt.

4.1 MX Records ☺☺☺

Die Spezifikation des DNS definiert für besondere Zwecke des Mail- Routings Mail Exchanger (MX) Resource Records.

(MB, MG, MINFO und MR Resource Records sind alternative Konzepte, die sich jedoch nicht durchgesetzt haben. -> X.400 wie z.B Lotus Notes Adressierung: Felix-Muster/Informatik/Zurich/CH/ZURICH)

Ein MX-Record spezifiziert einen Domainnamen und einen zugehörigen Rechner - den Mail Exchanger -, der in der Lage ist, Mail an eine E-Mailadresse mit diesem Domainnamen auszuliefern. (Der Domain-Teil einer E-Mailadresse braucht kein Domainname eines existierenden Rechners zu sein. Siehe auch Abschnitt 4.2). Es

[7] Edith Petermann, e-mail: Edith.Petermann@rz.uni-mannheim.de

können mehrere Mail Exchanger für den gleichen Domainnamen angegeben werden.

Mailsysteme können so konfiguriert werden, dass sie einen Nameserver nach MX Einträgen für den Domain-Teil einer gegebenden Zieladresse abfragen. (WKS Records werden von allen bekannten SMTP- Systemen bei der MX-Verarbeitung nicht berücksichtigt. Hierzu /RFC-974/ und /RFC-1123/.) Die Syntax von MX-Records lautet:

```
belwue.ch.      IN      MX      50      noc.belwue.de.
                IN      MX      100     ncc.belwue.de.
*.belwue.de.    IN      MX      50      noc.belwue.de.
                IN      MX      100     ncc.belwue.de.
```

Die ersten beiden MX-Records definieren noc.belwue.de und ncc.belwue.de als Mail Exchanger für den Domainnamen belwue.de . Eine an postmaster@belwue.de adressierte E-Mail wird an den Rechner noc.belwue.de geschickt, der dann für das weitere Ausliefern zuständig ist. Der Term gibt die Reihenfolge an, die von einem Mailer befolgt werden muss, wenn mehrere Mail Exchanger für einen Domainnamen existieren. Mail Exchanger mit kleineren Präferenzwerten müssen bei der Verbindungsaufnahme bevorzugt werden. Falls z.B. der Rechner noc.belwue.de nicht erreichbar sein sollte, wird die E-Mail an ncc.belwue.de geschickt, da dieser Rechner den nächst höheren Präferenzwert aufweist. Sind mehrere Mail Exchanger mit gleichen Präferenzwerten angegeben, kann ein Mailer den Mail Exchanger, der die E-Mail vorrangig geliefert bekommt, zufallsgesteuert auswählen .

- der Auswahlalgorithmus ist hierbei nicht mehr festgelegt. Ist kein Mail Exchanger definiert, d.h. es gibt dazu keinen MX-Record, versuchen Mailsysteme, die Nachricht direkt an den mit bezeichneten Rechner auszuliefern, falls für diesen ein A-Record definiert wurde. Diese Tatsache soll den/die Domainadministrator/in jedoch nicht veranlassen, MX-Records für Rechner zu unterschlagen, wenn A-Records definiert sind. Der geringere Arbeitsaufwand für die Pflege der DNS-Datenbasis wird durch eine höhere Belastung des Nameservers erkauft, da bei fehlendem MX-Record die Auflösung der Rechneradresse eine zusätzliche DNS-Abfrage erfordert.

Der zweite Satz vom MX-Records im obigen Beispiel verwendet Namen mit Wildcards (*). In diesem Fall kann für die Menge von Domainnamen, die in hinteren Teil die Zeichenkette belwue.de aufweisen (z.B. nic.belwue.de), ein gemeinsamer Mail Exchanger spezifiziert werden. Es ist darauf hinzuweisen, dass die Wildcard * auch für mehrere Labels Platzhalter sein kann (z.B. für den Rechner bingo.foo.belwue.de in der Subdomain foo von belwue.de). Desweiteren schliesst *.belwue.de nicht (!) den Namen belwue.de ein.

Wildcards verlieren jedoch ihre Wirkung, wenn die gefragte Zieldomain zu einer delegierten Zone gehört. Z.B. gilt das Muster *.belwue.de in einem MX RR nicht für den Domainnamen info.stgt.belwue.de , falls stgt.belwue.de eine delegierte Subdomain von belwue.de ist. Oder wenn die gefragte Zieldomain oder ein Name

zwischen der Wildcard Domain und der gefragten Zieldomain in einem Resource Record gleich welcher Art definiert wurde.

Der Algorithmus für die Erweiterung von Wildcards lautet somit: [8]

```
IF                      there is ANY sort of RR for the machine
THEN        IF          there is an non-wildcard MX for it
            THEN        use those in order
            ELIF        there is a A record (or records)
            THEN        use it (/them)
            ELSE        fail
            FI
ELIF                    there is a wildcard MX record
THEN                    use it
ELSE                    fail
FI
```

Hierzu ein Beispiel:

```
*.belwue.de.      IN    MX    50    noc.belwue.de.
noc.belwue.de.    IN    A           129.143.2.1
                  IN    MX    50    noc.belwue.de.
foo.belwue.de.    IN    MX    50    bar.belwue.de.
bar.belwue.de.    IN    A           129.143.2.14
                  IN    MX    50    bar.belwue.de.
```

Der Mail Exchanger noc.belwue.de darf keine E-Mail für die Rechner foo.belwue.de oder bingo.foo.belwue.de abnehmen, auch nicht für den Fall, dass bar.belwue.de nicht betriebsbereit wäre. Er akzeptiert darüberhinaus auch nicht E-Mail an die Adressen bar.belwue.de oder smurf.bar.belwue.de, da für den Namen bar.belwue.de ein A-Record existiert. Wird jedoch gewünscht, dass alle E-Mail an Adressen innerhalb der Domain noc.belwue.de - mit Ausnahme von foo.belwue.de und bar.belwue.de -- von dem MXer noc.belwue.de in Empfang genommen wird, sind weitere MX Records notwendig:

```
*.belwue.de.      IN    MX    50    noc.belwue.de.
noc.belwue.de.    IN    A           129.143.2.1
                  IN    MX    50    noc.belwue.de.
foo.belwue.de.    IN    MX    50    bar.belwue.de.
*.foo.belwue.de.  IN    MX    50    noc.belwue.de.
bar.belwue.de.    IN    A           129.143.2.14
                  IN    MX    50    bar.belwue.de.
*.bar.belwue.de.  IN    MX    50    noc.belwue.de.
```

Das obige Beispiel zeigt, dass Wildcard MX-Records oft nicht die Wirkung zeigen, die man erwartet. Sie sollten nur mit grosser Vorsicht verwendet werden.

Wie oben schon erwähnt, kann trotz fehlendem MX-Record für eine gegebene Zieladresse eine SMTP-Verbindung zustande kommen. Voraussetzung dafür ist, dass für die Zieladresse im DNS ein A-Record definiert wurde, was bedeutet, dass sie einen IP-adressierbaren Rechner repräsentiert. Zusätzlich muss der Zielrechner eine spontane SMTP-Verbindung akzeptieren können, d.h. ein

[8] (Der Algorithmus wurde von Piete.Brooks@cl.cam.ac.uk auf der sun-nets Mailing Liste veröffentlicht. Weitere Einzelheiten zur Interpretation von Wildcard MX-Records sind in /RFC1034/ S. 25--26 zu finden.)

staendig "empfangsbereites" Mailsystem betreiben. (Genauer gesagt muss das Mailsystem des Zielrechners einen "Server"-Prozess betreiben, der ständig auf eintreffende SMTP-Verbindungen wartet und die zu übertragenden Daten auch unmittelbar abnehmen kann - selbst bei mehreren simultan aufgebauten Verbindungen. Hierzu sind insbesondere die Mailsysteme von Personal Computern meist nicht in der Lage.) In allen anderen Fällen werden MX-Records notwendig. Einige typische Beispiele für den Einsatz von MX-Records werden im folgenden vorgestellt.

4.2 „Offizielle“ bzw. „bereinigte“ Mailadressen

Häufig werden in Organisationen Mailadressen der Mitarbeiter so gewählt, dass keine Rechnerinformationen darin vorkommen. (Im obigen Beispiel wird die Mailadresse postmaster@belwue.de verwendet, wobei der Domainname belwue.de kein Name eines Rechners ist.) Hierzu muss ein Mail Exchanger bereitgestellt werden, der alle Nachrichten an "bereinigte" Mailadressen innerhalb der Organisation weiterverteilen kann. Ebenso kann dafür gesorgt werden, dass alle aus der Organisation ausgehende Mail mit einer bereinigten Absenderadresse versehen wird. (Dies kann entweder durch das Mailsystem an jedem lokalen Rechner oder an einem zentralen "Mailhost" durchgeführt werden. Im letzteren Fall bietet sich die Integration von Mail Exchanger/Mailhost an.)
Offizielle Mailadressen erlauben die Definition von Adressierungsregeln, mit denen aus Vor- und Nachnamen von Benutzern zusammen mit dem Domainnamen der Organisation Mailadressen

(z.B. donald.duck@disneyland.com)

generiert werden können, die den Vorteil aufweisen, nach aussen hin auch dann gültig zu bleiben, wenn der Benutzer seinen Rechner wechselt.

MX-Records können auch zum Schutz von Rechnern eingesetzt werden, für die aufgrund von Sicherheitrisiken im Nameserver keine Einträge (A-Records, PTR-Records) gemacht werden, und deren IP-Adresse geheim gehalten werden muss. Ein solcher "verborgener" Rechner ist im Internet nur durch E-Mail erreichbar, nicht aber durch telnet oder ftp.

Die Adressierung erfolgt wie oben über eine vom Rechnernamen bereinigte Mailadresse, die auf einen Mail Exchanger zeigt.

4.3 Mail-Gateway-Routing

Im weltweiten Netzwerkverbund ist E-Mail zur Zeit die einzige Kommunikationsform, die auch über die Grenzen der verschiedenen existierenden Protokollwelten hinweg funktioniert. So kann man z.B. von einem SMTP-Host aus Nachrichten an BITNET-, X.400- oder UUCP- Adressen verschicken. Die Vermittlung zwischen den unterschiedlichen Mailprotokollen erledigen Mail-Gateways.
Die lange Zeit übliche Praxis, Angaben zum Gateway-Routing in Mailadressen unterzubringen

(Z.B. user%final-host@gateway-host oder hop1!hop2!hop3!user@gateway-host.)

, überfordert viele E-Mail-Benutzer und führt zudem zu Zieladressen, die vom jeweiligen Standpunkt des Absenders abhängig sind. Wesentlich eleganter ist es, die Routing- Information im DNS in Form von MX-Records abzulegen: Für eine Adresse in der fremden Protokollwelt existiert ein MX-Record, der auf ein Mail-Gateway zeigt, dass für die Konvertierung und Weiterleitung der Nachricht verantwortlich ist. Ein Benutzer muss nur noch wissen, in welcher Weise eine Mailadresse der fremden Protokollwelt in der RFC822-Notation notiert wird.

Beispiel:

*.uni-stuttgart.dbp.de. IN MX 10 noc.belwue.de

Hinter dem Namensraum uni-stuttgart.dbp.de stehen alle X.400-Adressen innerhalb der Universitaet Stuttgart (C=de, ADMD=dbp und PRMD=uni-stuttgart). (Die Abbildung zwischen der O/R- und RFC822-Notation von X.400-Adressen spezifiziert /RFC987/.) Der Mail Exchanger noc.belwue.de betreibt ein SMTP/X.400 Gateway, welches alle Nachrichten an diese Adressen zu den zugeständigen X.400 MTAs weiterleitet. Leider lässt sich die Routing-Information zu Mail- Gateways nicht immer so einfach im DNS unterbringen. Insbesondere bei den flachen Namensräumen von BITNET/EARN und UUCP sind Mailgateways nicht über weltweit gültige MX-Records ansprechbar. (Es ist im Prinzip möglich, mit einem Wildcard MX-Record ein lokales SMTP/BITNET- oder SMTP/UUCP-Gateway zu spezifizieren. Dann aber muss gewährleistet sein, dass diese Information ausschliesslich von solchen Rechnern abfragbar ist, die sich im „Einzugsbereich“ des Gateways befinden. Wir raten jedoch davon ab.)

4.4 Spezifikation eines „Fallback“-Mailhosts

Wenn ein Rechner am Internet nicht in Betrieb ist, kann mit ihm keine SMPT-Verbindung aufgenommen werden. Eine Nachricht, die an solch einen Rechner adressiert ist, wird vom sendenden Rechner in der Regel ca. drei Tage in einer Warteschlage gehalten. Während dieser Zeit wird periodisch versucht, die Mail an den Zielrechner auszuliefern. Gelingt dies nicht, erhält der Absender seine Nachricht zusammen mit einer Fehlermeldung zurück. Mit einem MX-Record kann für einen nur sporadisch betriebenen Rechner ein Mailhost spezifiziert werden, der während der Stillstandzeiten die Nachrichten stellvertretend in Empfang nimmt. Beispiel:

small-pc.belwue.de.	**IN**	**MX**	**50**	**small-pc.belwue.de.**
	IN	**MX**	**100**	**mailmaster.belwue.de.**

Ist der Rechner small-pc.belwue.de nicht erreichbar, springt für ihn der Rechner mailmaster.belwue.de ein, und bewahrt die Nachrichten für small-pc solange auf, bis dieser empfangsbereit ist. Auf welche Weise small-pc von mailmaster die aufbewahrten Nachrichten empfängt, kann von Fall zu Fall verschieden sein. (z.B. könnte mailmaster eine besonders lange timeout-Zeit für die Weiterleitung der Mail verwenden, oder er könnte sie in einer lokalen Mailbox ablegen, auf small-pc bei gegebener Zeit über das "Post Office Protocol" /RFC 1081/ zugreifen kann.) Einzelheiten darüber gehen über den Rahmen dieses Handbuchs hinaus.

Das Fallback-Konzept bietet sich auch zur Absicherung von Rechnern an, die in einer grosser Organisation zentrale Kommunikationsdienste wie z.B. Mailhost, Mailgateway übernehmen:

noc.belwue.de.	IN	MX	50	noc.belwue.de.
	IN	MX	100	ncc.belwue.de.

Der Rechner noc.belwue.de fungiert als Mailhost für eine Grosszahl von Workstations auf dem Campus der Universität Stuttgart. Eine Mail an eine nicht-lokale, d.h. nicht in der selben Domain des Absenders liegende Adresse wird zuerst an den Mailhost geschickt, der sie entsprechend seiner Routingstrategien an den Zielrechner oder an ein Mailgateway weiterleitet. Beim Ausfall des Mailhosts noc.belwue.de würde für eine grosse Zahl von Rechnern die Mail-Verbindung zur Aussenwelt unterbrochen werden. Für diesen Fall steht der Fallback-Mailhost ncc.belwue.de bereit, der in diesem Fall die Dienste von noc.belwue.de übernehmen kann. Voraussetzung für den dynamischen Ersatz des Mailhosts ist die Fähigkeit des Mailsystems der lokalen Rechner, die IP-Adresse des Mailhosts über DNS-Abfragen aufzulösen.

4.5 Fehlerquellen

Im folgenden werden einige Fehlerquellen bei der Verwendung von MX-Records aufgezeigt, die für das Mail-Routing Überraschungen hervorrufen können.

4.5.1 Alias-Namen in MX-Records

Der Algorithmus zur Elimination von irrelevanten bzw. verbotenen Mail Exchangern /RFC974/ versagt, wenn in MX-Records Alias-Namen verwendet werden. (Alias-Namen dürfen prinzipiell nur in CNAME-Records auftreten.) Dies kann zu Mail-Loops führen. Um derartige Probleme von vornherein auszuschliessen, sollten Alias-Namen in MX-Records prinzipiell vermieden werden.

4.5.2 Verkettete MX Records

Eine Anordnung von MX-Records der Form:

*.foo.de.	IN	MX	0	host-a.foo.de.
host-a.foo.de.	IN	MX	0	host-b.bar.de.

hat die Wirkung, dass alle Nachrichten für Rechner in der Domain foo.de über den Mail Exchanger host-a.foo.de weitergeleitet werden. Der Rechner host-b.bar.de erhält nur Nachrichten, die an host-a.foo.de direkt adressiert sind, nicht jedoch Nachrichten, die an Rechner in der Domain foo.de adressiert sind.

4.5.3 Mail Exchanger mit Default-Mailhost

Es ist zulässig, dass ein Mail Exchanger mit einem Default-Mailhost kooperiert (Ein Default-Mailhost oder auch Mail-Relay ist ein besonderer Rechner, an den andere Rechner solche E-Mail schicken können, die sie selbst nicht ausliefern wollen oder können - z.B. weil ihr Mailsystem keine MX Information auswerten kann. Häufig werden Rechner so konfiguriert, dass sie nur lokale Mail selbst ausliefern und nicht-lokale Mail dem Default-Mailhost überlassen.)
In diesem Fall ist das Mailsystem des Mail Exchangers so konfiguriert, dass es ausgehende Nachrichten an nicht-lokale Adressen - d.h. Adressen ausserhalb der eigenen Domain - sofort an den Default-Mailhost weiter - leitet, der für das weitere Routing zuständig ist. Erhält jedoch ein Mail Exchanger eine Nachricht mit einer Zieladresse, die er laut MX-Record annehmen oder weiterleiten sollte, muss sein Mailsystem diese Adresse auch "erkennen". Ist dies nicht der Fall, wird die Zieladresse als nicht-lokal interpretiert und die Nachricht an den Default-Mailhost zurückgeschickt. Auf diese Weise können Mailschleifen zwischen Mail Exchanger und Mailhost hervorgerufen werden. Es muss deshalb vor dem Eintrag des MX-Records in das DNS sichergestellt werden, dass das Mailsystem des Mail Exchangers alle in den MX-Records spezifizierten Domainnamen akzeptiert.

Unter Unix kann für den SMTP-MTA sendmail kann dazu der Test-Modus (Aufruf sendmail -bt) herangezogen werden.

Für den Zeitraum der Konfiguration und des Tests des Mailsystems eines Mail Exchangers sollte der statische Eintrag eines Default-Mailhosts ausser Kraft gesetzt werden. Es ist anzuraten, Mail Exchanger so „intelligent“ wie möglich zu machen, dass sie keinen Default-Mail- host benötigen.

5.0 Implementation eines DNS Servers als Workshop

Diese Kapitel wird sehr kurz ausfallen, da das Hauptdokument auch in zwei Jahren noch Gültigkeit haben soll. In diesem Sinne wird der Workshop lediglich die Angehensweise beschreiben. Wie das Adressierungskonzept und die Arbeitsweise eines DNS Servers und einer DNS Architekur aussehen ist klar, aber wie Soft- und Hardware in zwei oder 3 Jahren aussehen ist sehr spekulativ.

Von einer Neuadressierung ist weit und breit nichts zu sehen. Es gibt einen Verwandten zum DNS nämlich das Telefon mit seinem Adressierungskonzept ... kommt dir doch bekannt vor. Wie gross wäre wohl der Aufwand einer weltweiten Umstellung zu beziffern, mit der Auflage ständiger Erreichbarkeit ?

Also schreiten wir voran....

Wir werden den DNS Server von Microsoft installieren ... Klingt echt spannend ... Ich habe diesen gewählt, weil wird dann mehr Augenmerk auf Funktionalität und Konfigruation von DNS legen können und uns nicht mit systemspezifischen Eigenheiten, wie z.B. unter Unix herumschlagen müssen.

5.1 Konzeptionelle Überlegungen

Wir dürfen uns nichts vormachen - ohne konkreten Business Case geht hier gar nichts. Um diesen definieren zu können muss zuerst einmal abgecheckt werden, was überhaupt im Rahmen des Möglichen drin liegt.
Arbeite ich als Webmaster bei einem grossen Unternehmen, welches bereits eine eigene Infrastruktur hat oder bin ich bei einem ISP (Service Provider) tätig, welcher ebenfalls eine eigene Infrastruktur hat ? Diese Frage stellt sich. Eine Ausnahme hierzu stellt sich höchstens, wenn du bei einem ISP einsteigst, welcher den Markt von unten her aufbauen aufrollen will. Mit anderen Worten, Du wirst vermutlich auf bestehende Systeme stossen, welche bereits laufen, und somit lediglich Konfigurationsänderungen einspielen.
Eine eigene Infrastruktur nur für DNS wäre wohl zu kostspielig, wobei man nicht vergessen darf, dass es bereits Anbieter (auch ISP's) im Internet gibt, welche diese Dienstleistung (DNS) auch separat anbieten. Man muss halt auch mal nachfragen ....

5.1.1 Wie sieht unsere Infrastruktur aus ?

Falls wir selber loslegen wollen und eine eigene Infrastruktur für die Inbetriebnahme eines DNS Server aufbauen wollen brauchen wir sicher ...

- Eine Standleitung zu unserem Carrier (Swisscom, Diax, Cable & Wireless o.a.) Eine DIAL-Up Kommunikation ist abzulehnen, da der Server ständig erreichbar sein muss. (Denkbar wäre lediglich der Einsatz eines CacheServers mit langem TTL - aber auch dieser Gedanke ist mir nicht sympatisch, da Änderungen in der DNS häufig vorkommen können.
- Einen oder besser zwei PC-Server mit mindestens einem ISDN Adapter (Es kann theoretisch auch ein ganz normaler PC sein. Wir dürfen nicht vergessen, dass das DNS nicht viel Rechenleistung braucht. Schneller Diskzugriff (SCSI 3 oder Fibre Channel) und viel Memory (min 256MB) sind aber ratsam.
- Ein Betriebsystem mit dazu passender DNS Server Software und Hardware. In diesem Fall entscheiden wir und für Windows NT Server und den Microsoft DNS Server. [9]
- Einen funktionierenden TCP/IP Stack mit einer im Internet gültigen IP Adressen sowie die Kontaktadressen von (zur Zeit Switch für .ch und InterNic für .com, .org und .net
- Ein Verständnis von Betriebsystem, Netzwerkkommunikation und vor allem von der Funktionsweise des DNS.

5.1.2 Wie sehen die Anforderungen aus ?

Die Anforderungen ergeben sich aus ... ist vermutlich der falsche Ansatz. Wir müssen lediglich sicherstellen, dass Email an die richtige Adresse weitergeleitet werden können und dass Browser den Namen richtig auflösen (Natürlich muss unser Cache ebenfalls aktuell sein). Diese wenigen Aufgaben stellen sich für den DNS Administrator.

[9] Wir hätten uns auch anders entscheiden können. Es gibt Freeware Betriebsysteme oder verschiedene kommerzielle Unix Derivate u.v.m.

Komplexer wird es wenn mir unterschiedlichen Zonen gearbeiten werden muss. Z.B. bei einem international tätigen Unternnehmen - sogenannten Intranets. Dort können sehr komplexe DNS Architekturen vorkommen (siehe Beispiel in Kapitel 3.3)

5.1.3 Konsequenzen aus oben genannten Punkten

Jedes weltweit tätige Unternehmen ist bereits mit DNS erschlossen. Also wird der Bedarf an ReDesign nicht sehr gross sein. Wir sehen uns also mit der Integration von KMU beschäftigt bzw. von neuen Organisationseinheiten in Internationalen Unternehmen. Immer häufiger werden auch DNS Namen als "product placement". Alle diese Anforderungen sind eigentliche Peanuts. Du wirst es sehen ... denn ein Aliase und MX Records einzutragen ist wahrlich keine Hexerei.

5.2 Konfiguration und Installation eines DNS Servers

Aus Aktualitästsgründen wird dieser Teil als seperates Dokument geführt.

Siehe Dokument:
"DNS unter Windows NT - Die Konfiguration des Microsoft DNS Servers"

5.3 Konfiguration eines Client

6.0 Der URL

Uniform Resource Locators (URL)

6.1 Uniform Resource Locator (URL) ☺☺☺

Das World Wide Web-Projekt hat neben den beiden Features HTTP und der Hypertext-Funktionalität sowie dem DNS noch einen viertes Standbein, daß die revolutionäre Verbreitung erst ermöglichte: Dank der URL-Schemata ist es möglich, jede Ressource im Internet (auch außerhalb des WWW) eindeutig zu adressieren und anzusprechen.

Das grundlegende Schema

Ein gültiger URL hat grundsätzlich folgende Syntax:

<Protokoll>://<Userid>@<Host>:<Port>/<Pfad>

»Protokoll« steht für das anzuwendende Internet-Protokoll, also z.B. »http« für Hypertext Transfer Protocol, »ftp« für File Transfer Protocol oder »mailto« für den eMail-Dienst. Gefolgt wird diese Protokollkennzeichnung von einem

Doppelpunkt und, falls der URL eine nicht lokale Aktion auslösen soll, mit zwei Querstrichen (»//«).

»UseriD« enthält User-ID und/oder Paßwort-Informationen, die zum Benutzen der Ressource unbedingt erforderlich sind. Wird zum Beispiel eine paßwortgeschützte HTTP-Ressource angesprochen, können alternativ zur Dialogfenstereingabe diese Angaben auch hier eingesetzt werden, was z.B. so einen Bookmark ermöglicht. Also z.B :

```
http://ErnstEiswuerfel:GeheimesPasswort@www.barnes.ch/Telematik/index.html
```

Standardmässig meldest du dich an jedem System (Webserver) im Internet an. Und zwar mit dem „anonymous“ – Benutzer, den ein Webserver für den public Access braucht an.

»Host« enthält den Rechner, der angesprochen werden soll, also z.B. »barnes.ch«.

»Port« enthält die Nummer des TCP/IP-Ports, an den die Anfrage geschickt wird. Z.B. werden HTTP-Anfragen voreingestellt an Port 80 geschickt.

»Pfad« enthält nun die Informationen, die »vor Ort« beim Server verarbeitet werden. Dazu gehören Pfadangaben, aber auch Übergabeparameter, Sprungziele und Abfrageinformationen.

6.2 URL-Schema bei »http«

»http« kennzeichnet alle URL, die mit dem HTTP-Protokoll arbeiten. Voreingestellt ist bei »http« die Portnummer 80, bei Bedarf kann jedoch explizit auch ein anderer Port angesprochen werden (z.B. zum Ansprechen eines Proxy-Servers).

`http://www.netplanet.org`
Der Browser schickt eine GET-Anfrage direkt an den Host »www.netplanet.org«. Da explizit keine Datei angefordert wird, schickt der Server entweder einen Index des Verzeichnisses oder eine Default-Datei, d.h. die Datei, die vom Administrator des Servers dazu auserwählt wurde, immer dann verschickt zu werden, wenn nur ein Verzeichnis angesprochen wird.

`http://user-ID:Passwort@www.netplanet.org`
Informationen, die vor den Klammeraffen (»@«) gesetzt werden, der seinerseits vor der Hostadresse sitzt, werden als Identifikationsparameter, z.B. für serverseitig paßwortgeschützte Bereiche, benutzt. Moderne Browser blenden bei Anwahl einer solchen Seite ein Dialogfeld ein, in das die Zugangskennungen eingegeben werden müssen. Intern übersetzt der Browser diese Informationen in diese URL-Form. Durch direkte Eingabe der Zugangskennungen in einen URL können Sie z.B. einen Bookmark setzen, der direkten Zugriff auf die

paßwortgeschützte Seite ermöglicht. Hinweis: Werden die Zugangskennungen direkt in die URL eingesetzt, werden sie nicht veborgen angezeigt!

`http://www.netplanet.org:8080`
Zusätzlich kann auch direkt der Port gewählt werden, an den die Anfrage beim Host geschickt werden soll. Ist im URL kein Port angegeben, wird standardmäßig die Anfrage an den Port 80 geschickt, der für HTTP reserviert ist.

`http://www.netplanet.org/index.html`
Bei diesem Aufruf wird explizit die Datei »index.html« vom Host angefordert.

`http://www .netplanet.org/index.html#sprungziel`
Ist in einem URL eine Raute vorhanden, so wird die Information hinter der Raute bis zu einem nächsten URL-spezifischen Zeichen (oder dem Ende des URL) vom Browser als Sprungziel innerhalb der spezifizierten Datei definiert. Die komplette Datei wird angefordert und lokal nach dem angegebenen Sprungziel abgesucht. Ist das Sprungziel nicht vorhanden, wird die Datei so angezeigt, als wäre sie ohne Sprungziel angefordert worden.

`http://www.netplanet.org/index.html?ak=Abfrag e`
Ein Fragezeichen in einem URL kennzeichnet bis zu einem nächsten URL-relevanten Zeichen (oder dem Ende der URL) den sogenannten »Abfrageteil«. In diesem Bereich lassen sich Parameter oder Informationen mit dem URL verschicken und entfernte Aktionen, z.B. CGI-Programme, füttern. Zum Versenden muß jedoch der Text der Variablen angepaßt werden, d.h. Sonderzeichen und Leerschritte müssen maskiert werden.

`http://www.netplanet.org/editorial/`
In diesem Fall wird ein Zugriff auf das Verzeichnis »editorial« auf diesem Host vorgenommen. Da es sich um ein Verzeichnis handelt, schickt der Server automatisch die Default-Datei, ansonsten einen Index vom Verzeichnis. WICHTIG: Wird direkt auf ein Verzeichnis zugegriffen, sollte immer ein abschließender Schrägstrich folgen, da der Server bei fehlendem Schrägstrich immer davon ausgeht, daß es sich um eine Datei handelt. Findet er keine Datei, schickt er eine HTTP-Fehlermeldung (Klasse 3xx - Redirection), die jedoch, dank fehlertoleranter Programmierung moderner Browser, nicht angezeigt wird, sondern den Browser dazu veranlaßt, den gleichen URL nochmals mit abschließendem Schrägstrich abzusenden.

6.3 URL-Schema bei »mailto«

Beginnt ein URL mit »mailto«, so ruft ein Klick auf diesen Link in der Regel ein Mailprogramm auf und übernimmt die Parameter des URL. Der URL wird also nicht verschickt, sondern löst nur eine lokale Aktion aus.

`mailto:besim@netplanet.org`
Der »klassische« mailto-URL, der ein Mailfenster öffnet und lediglich die Adresse des Empfängers vorgibt.

`mailto:bes im@netplanet.org?to=hans@mustermann.de`

Mit »to=« können zusätzliche Empfänger angegeben werden.

`mailto:besim@netplanet.org?cc=hans@mustermann.de`
»cc=« ermöglicht die Angaben von eMail-Adressen, die als CC-Empfänger (»Carbon Copy«) angegeben werden können.

`mailto:besim@netplanet.org?bcc=hans@mustermann.de`
Mit diesem, ebenfalls fiktiven, URL wird hans@mustermann.de als BCC-Empfänger (»Blind Carbon Copy«) voreingestellt.

`mailto:besim@netplanet.org?newsgroups=de.test`
Dieser URL bereitet die Mail gleichzeitig auf den Versand zur eMail-Adresse und zur news-Gruppe »de.test« vor.

`mailto:besim@netplanet.org?body=Dies ist ein Text`
Mit »body=« kann der Body-Text der eMail vorgeschrieben werden, der nach Anklicken schon im Mailfenster erscheint (selbstverständlich aber gelöscht oder noch editiert werden kann).

6.4 URL-Schema bei »news«

URL beginnend mit »news« lösen ebenfalls nur lokale Aktionen aus, in dem sie den news-Client öffnen und ggf. die news-Gruppe und evt. den news-Server vorbestimmen.

`news:`
Dieser URL allein öffnet nur den news-Client und baut evtl. eine Verbindung zu angegebenen news-Servern auf, um die vorhandene Zahl der Beiträge bei evtl. abonnierten news-Gruppen zu aktualisieren.

`news:de.test`
Bei Eingabe dieses URL wird der news-Client aufgerufen und direkt in die angegebene news-Gruppe gesprungen. Meist wird die Gruppe damit auch gleich abonniert. Ist lokal kein news-Server angegeben oder die angegebene news-Gruppe auf dem lokalen news-Server nicht vorhanden, wird eine Fehlermeldung ausgegeben.

`news:news.server.de/de.test`
In diesem URL ist neben der news-Gruppe auch der news-Server angegeben, zu dem konnektiert werden soll. Die news-Gruppe wird dann von diesem news-Server abonniert.

7.0 Tools rund um DNS

7.1 NSLOOKUP

Note: The following information about nslookup is copied from the Windows NT Server Resource Kit v4.0 Help Command

This diagnostic tool displays information from Domain Name System (DNS) name servers. Before using this tool, you should be familiar with how DNS works. Nslookup is availab le only if the TCP/IP protocol has been installed.
Syntax

nslookup [-option …] [hostname | - [server]]

Modes

Nslookup has two modes: interactive and non -interactive.
If you only need to look up a single piece of data, use non -interactive mode. For the f irst argument, type the name or IP address of the host to be looked up. For the second argument, type the name or IP address of a DNS name server. If you omit the second argument, the default DNS name server will be used.
If you need to look up more than o ne piece of data, you can use interactive mode. Type a hyphen (-) for the first argument and the name or IP address of a DNS name server for the second argument. Or, omit both arguments (the default DNS name server will be used).

Parameters

-option …

Specifies one or more nslookup commands as a command -line option. For a list of these optional commands, see the following Table A.2, Nslookup Commands in Windows NT. Each option consists of a hyphen (-) followed immediately by the command name and, in some cases, an equal sign (=) and then a value. For example, to change the default query type to host information and the initial timeout to 10 seconds, you would type:

nslookup -querytype=hinfo -timeout=10

The command line length must be less than 256 charac ters.

hostname

Look up information for hostname using the current default server or using server if specified. If computer -to-find is an IP address and the query type is A or PTR, the name of the computer is returned. If hostname is a name and does not h ave a trailing period, the default DNS domain name is appended to the name. (You can change this behavior by using the nslookup set command. Refer to the following Table A.2, Nslookup Commands in Windows NT. To look up a computer not in the current DNS dom ain, append a period to the name.

If you type a hyphen (-) instead of hostname, the command prompt changes to nslookup interactive mode.

server

Host name of the DNS server to use. If you omit server, the default DNS name server is used.

Notes

Interactive Commands

- To interrupt interactive commands at any time, press CTRL+C.
- To exit, type exit.
- The command line length must be less than 256 characters.
- To treat a built-in command as a host name, precede it with the escape character (\).
- An unrecognized command is interpreted as a host name.

Diagnostics

If the nslookup command fails, an error message prints. Possible errors are:

- Timed out

The server did not respond to a request after a certain amount of time (changed with set timeout=value) and a certain number of retries (changed with set retry=value).

- No response from server

No DNS name server is running on the server.

- No records

The DNS name server does not have resource records of the current query type for the host, although the host name is valid. The query type is specified with the set querytype command.

- Non-existent domain

The host name or DNS domain name does not exist.

- Connection refused

- Or -

- Network is unreachable

The connection to the DNS name server or Fin ger server could not be made. This error commonly occurs with ls and finger requests.

- Server failure

The DNS name server found an internal inconsistency in its database and could not return a valid answer.

- Refused

The DNS name server refused to ser vice the request.

- Format error

The DNS name server found that the request packet was not in the proper format. It may indicate an error in nslookup.

The following table shows the nslookup commands. For details about syntax for individual nslookup comm ands, choose the nslookup commands topic in the TCP/IP Procedure Help.

Table A.2 Nslookup Commands in Windows NT

Command	Purpose
exit	Exits interactive nslookup.
finger	Connects with the Finger server on the current host. The current host is def ined when a previous lookup for a host was successful and returned the address information.
help	Displays a brief summary of nslookup commands.
ls	Lists information for a DNS domain. The default output contains host names and their IP addresses. (When output is directed to a file, hash marks (###) are printed for every 50 records received from the server.)
lserver	Changes the default server to the specified DNS domain.
Lserver	uses the initial server to look up the information about the specified DNS domain. (This is in contrast to the server command, which uses the current default server.)
root	Changes the default server to the server for the root of the DNS domain name space. Currently, the host G.ROOT - SERVERS.NET. is used. (This command is a synonym for lserver g.root -server.net.) The name of the root server can be changed with the set root command.
server	Changes the default server to the specified DNS domain. Server uses the current default server to look up the information about the specified DNS domain. (This is in contrast to the lserver command, which uses the initial server.)
set	Changes configuration settings that affect the behavior of the nslookup commands.
set all	Prints the current values of the configuration settings. Also prints information about the default server and host..
set cl[ass]	Changes the query class. (The class specifies the protocol group of the information.)
set [no]d2	Turns exhaustive debugging mode on or off. Essentially all fields of every packe t are printed.
set [no] deb[ug]	Turns debugging mode on or off. With debugging on, more information is printed about the packet sent to the server and the resulting answer.
set [no] def[name]	If set, appends the default DNS domain name to a single-component lookup request. (A single component is a component that contains no periods.)
set do[main]	Changes the default DNS domain to the name specified. The default DNS domain name is appended to a lookup request depending on the state of the defname and s earch options. The DNS domain search list contains the parents of the default DNS domain if it has at least two components in its name. For example, if the default DNS domain is mydomain.mycompany.com, the search list is

```
                  mydomain.mycompany.com and my company.com. Use the set
                  srchlist command to specify a different list. Use the set
                  all command to display the list.
Set [no] ig[nore]       If set, ignores packet truncation errors.
Set po[rt]        Changes the default TCP/UDP DNS name server port to the
                  value specified.
Set q[uerytype]   Changes the type of information query. More information
                  about types can be found in RFC 1035. (The set type
                  command is a synonym for set querytype.)
set [no] rec[urse]      If set, tells the DNS name server to query other
                        servers if it does not have the information.
Set ret[ry]       Sets the number of retries. When a reply to a request is
                  not received within a certain amount of time (changed
                  with set timeout), the timeout period is doubled and the
                  request is resent. The retry value controls how many
                  times a request is re-sent before giving up.
Set ro[ot]        Changes the name of the root server. This affects the
                  root command.
Set [no] sea[rch]       If set and the lookup request contains at least one
                        period but does not end with a trailing period,
                        appends the DNS domain names in the DNS domain
                        search list to the request until an answer is
                        received.
Set srchl[ist]    Changes the default DNS domain name and search list. A
                  maximum of six names separated by slashes (/) can be
                  specified. This command overrides the default DNS domain
                  name and search list of the set domain command. Use the
                  set all command to display the list.
Set ti[meout]     Changes the initial number of seconds to wait for a reply
                  to a request. When a reply to a request is not received
                  within this time period, the timeout is doubled and the
                  request is re-sent. (The number of retries is controlled
                  with the set retry option.)
set ty[pe]        Changes the type of information query. More information
                  about types can be found in RFC 1035.  (The set type
                  command is a synonym for set querytype.)
set [no] v[c]     If set, always uses a virtual circuit when sending
                  requests to the server.
view              Sorts and lists the output of previous ls command(s).
References

For in-depth coverage of nslookup, see DNS and BIND by Paul Albitz and
Cricket Liu, published by O'Reilly and Associates.
```

Hiermit endet die Version 1.0 zum Thema "Webmaster und DNS" von Bernhard Kreinz. Eine stängig auf dem laufenden gehaltene Version findest du unter http://www.barnes.ch/telematik