


# Übungsprotokoll

## SYTB – Systemtechnik Betriebssysteme

	<b>Übungsdatum:</b> KW 36/2021 – KW /2021	<b>Klasse:</b> 3AHIT	<b>Name:</b> Felix Schneider
	<b>Abgabedatum:</b> -	<b>Gruppe:</b> SYTB_2	<b>Note:</b>
<b>Leitung:</b> DI (FH) Alexander MESTL	<b>Mitübende:</b> -		
<b>Übungsbezeichnung:</b>  SYTB Mitschrift			

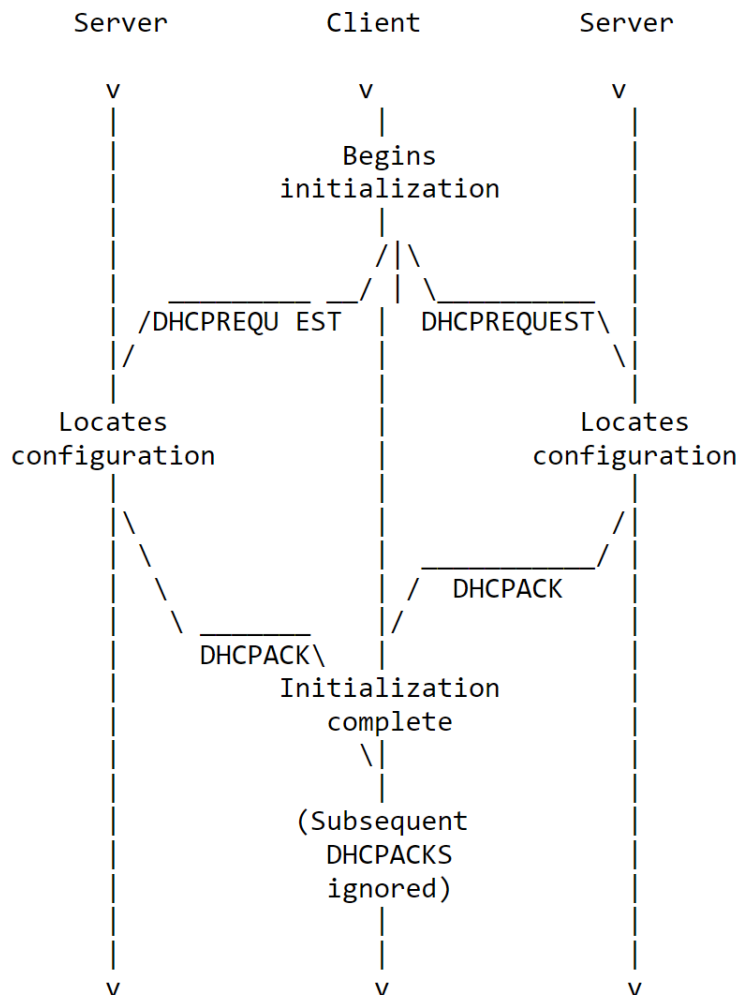
Inhaltsverzeichnis:

1	Theoretische Grundlagen .....	3
1.1	DHCP Server.....	3
1.1.1	Verbindungsaufbau mit 2 DHCP Servern im Netz .....	3
1.1.2	Lease Time .....	3
1.1.3	Verbindungsaufbau mit 1 DHCP Servern im Netz .....	4
1.1.4	ISC DHCP Server .....	4
1.2	DNS-Server .....	5
1.2.1	DNS .....	5
1.2.1.1	rekursiver DNS-Server .....	5
1.2.1.2	autoritärer DNS-Server .....	5
1.2.2	DDNS.....	7
1.2.2.1	Verschiedene Konfigurationen.....	7
1.2.2.2	rndc.....	7
1.3	Zonendatei.....	8
2	Übungsdurchführung .....	<b>Fehler! Textmarke nicht definiert.</b>

# 1 Theoretische Grundlagen

## 1.1 DHCP Server

### 1.1.1 Verbindungsaufbau mit 2 DHCP Servern im Netz



### 1.1.2 Lease Time

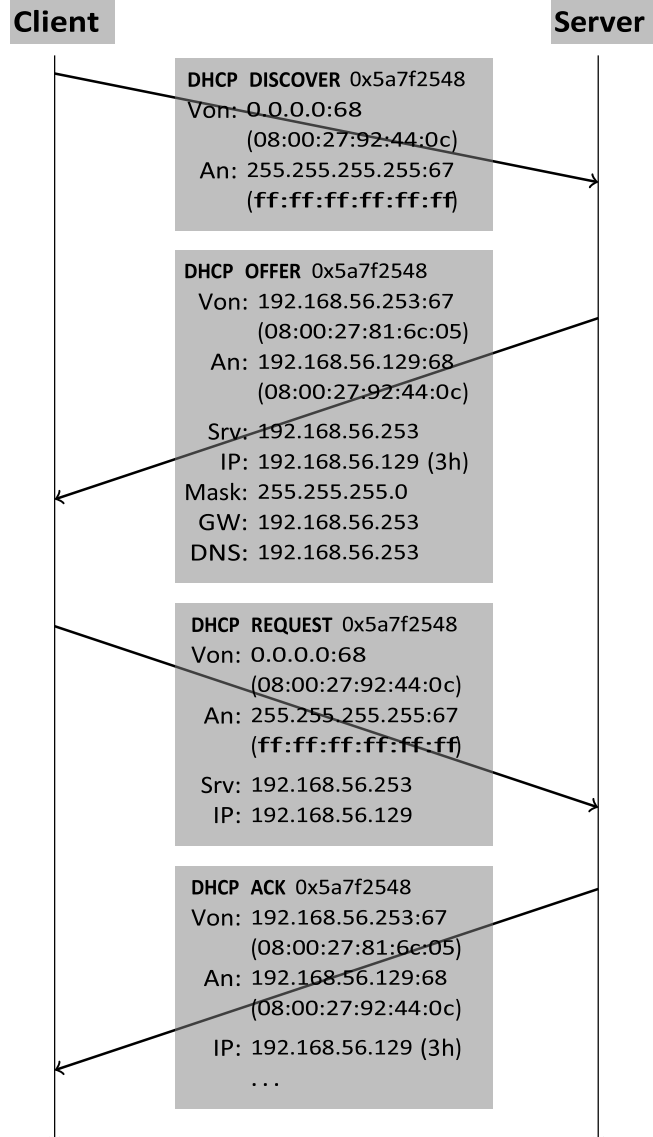
Bei öffentlichen Netzwerken macht es Sinn, die Lease Time kürzer (ca. 5 Minuten – 1 Stunde) einzustellen, weil sehr oft Geräte wechseln, und bei einer zu lange Lease Time bald keine IP-Adressen mehr verfügbar wären. Bei Netzwerken, wo sehr selten Geräte ein bzw. aussteigen, ist es sinnvoll eine längere Lease Time (ca. 1 Tag - 1 Woche) zu setzen.

Nach 50% der Lease Time fragt das Gerät an, ob es die Lease Time verlängern kann, wenn sich dieses noch im Netzwerk befindet.

Wenn der Client nach  $\frac{7}{8}$  der Lease Time (87,5%) keine Antwort vom DHCP Server bekommt, sucht sich der Client einen neuen DHCP Server.

Ein Client kann öfter dieselbe IP-Adresse erhalten, wenn diese vom DHCP Server gespeichert wird.

### 1.1.3 Verbindungsaufbau mit 1 DHCP Servern im Netz



### 1.1.4 ISC DHCP Server

weit verbreiteter DHCP Server

- statische IP-Adresse festlegen
- apt install isc-dhcp-server

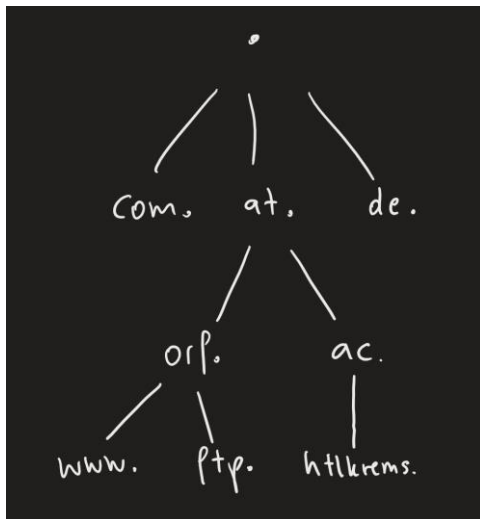
```
ddns-update-style none;  
ddns-updates off;  
authoritative;  
  
option domain-name "example.com";  
option domain-name-servers 192.168.56.253, 192.168.56.254;  
  
subnet 192.168.56.0 netmask 255.255.255.0 {  
    range 192.168.56.1 192.168.56.127;  
    option routers 192.168.56.254;  
}
```

authoritative:

Wenn dieses Flag gesetzt ist, dann schickt der DHCP Server bei Anfrage einer Adresse, die in diesem Netzwerk nicht mehr funktioniert (z.B.: vom vorherigen Netzwerk vom Rechner), ein Paket, das dem Client sagt, dass er diese Konfiguration wegwerfen kann. Ansonsten wartet der Client ewig.

## 1.2 DNS-Server

### 1.2.1 DNS



Der Browser löst den Namen von hinten nach vorne auf (von oben nach unten in Abbildung). Von der Topleveldomain bis zum Host.

Der DNS-Server speichert diese Anfragen (IP-Adressen und Informationen) im Cache.

DNS läuft auf Port 53. DNS überträgt mithilfe von UDP-Paketen.

#### 1.2.1.1 rekursiver DNS-Server

Behandelt Anfragen, indem er versucht, die IP-Adresse im Internet aufzulösen.

#### 1.2.1.2 autoritärer DNS-Server

Behandelt Anfragen, indem er Informationen hergibt, die er selbst gespeichert hat.

In der Datei `/etc/nsswitch.conf` wird gespeichert, welcher Ort der DNS-Server zuerst nach Informationen durchsucht; zum Beispiel steht dort dies drinnen:

```
hosts: files dns
```

In der Datei `/etc/resolv.conf` wird gespeichert, wie eine bestimmte Anfrage aufgelöst werden soll:

```
nameserver 192.168.10.1
nameserver 192.168.0.99
search      foo.example.com bar.example.com example.com
```

Die Befehle `dig` und `nslookup` prüfen, ob der DNS-Server funktioniert.

Hier ein Beispiel, wie eine Konfigurationsdatei in `/etc/bind` aussehen kann. Dies ist die `/etc/named.conf` Datei. Diese verweist hauptsächlich auf Zonendateien.

```
// BIND-Konfigurationsdatei

options {
    directory "/var/cache/bind";
    listen-on { 192.168.0.254; 127.0.0.1; };
    statistics-file "/var/log/bind/named.stats";
};

zone "." {
    type hint;
    file "/etc/bind/db.root";
};

zone "0.0.127.in-addr.arpa" {
    type master;
    file "/etc/bind/db.127";
};

zone "localhost" {
    type master;
    file "/etc/bind/db.local";
};

zone "example.com" {
    type master;
    file "/etc/bind/db.example.com";
};

zone "33.22.11.in-addr.arpa" {
    type master;
    file "/etc/bind/db/11.22.33";
};
```

Diese Datei wird eigentlich in 3 verschiedene Dateien aufgeteilt.

In der `/etc/bind/db.root` Datei stehen die 13 root-DNS-Server drinnen.

### 1.2.2 DDNS

Wenn sich die IP-Adressen ändern (was bei DHCP normal ist), dann benötigt man einen DDNS Server, weil dieser ansonsten Namen zu falschen IP-Adressen zuordnen würden. Der DHCP Server informiert den DDNS Server dann. „ddns-updates off“ schaltet diese Funktion aus.

range:

Vergibt die Spannweite, in der der DHCP Server IP-Adressen vergibt.

```
host blue {  
    hardware ethernet 08:00:27:92:44:0c;  
    fixed-address 192.168.56.129;  
}
```

Das ist eine fixe Zuweisung.

Anhand der MAC-Adresse kann der DHCP Server IP-Adressen für bestimmte Clients reservieren.

#### 1.2.2.1 Verschiedene Konfigurationen

##### 1.2.2.1.1 ddns-update-style none | ad-hoc | interim

bestimmt die Methode, die für die dynamische Aktualisierung verwendet wird. none bedeutet, dass keine Aktualisierung gemacht werden soll, und ist der sichere Wert für den Fall, dass Sie diese Funktion nicht verwenden wollen. interim ist für den Fall nötig, dass Sie sich daran versuchen möchten.

##### 1.2.2.1.2 ddns-updates on|off

Bestimmt, ob der Server versucht, bei einem DHCPACK das DNS zu aktualisieren. Der Standardwert ist (etwas lästigerweise) on.

#### 1.2.2.2 rndc

= name server control utility

Dieser Schlüssel verschlüsselt die Update-Dateien, die vom DHCP-Server zum DNS-Server geschickt werden. Dies ist eine symmetrische Verschlüsselung. Wir müssen einen Schlüssel erstellen, der zur Sicherung des Informationsaustauschs zwischen DHCP- und DNS-Server verwendet wird. Nur unser DHCP-Server sollte DNS-Datensatzaktualisierungen durchführen dürfen, nicht irgendjemand.

Mit „dnssec-keygen“ kann man seinen eigenen Key generieren.

Dieser Key ist aber eh schon voreingestellt:

```
key "rndc-key" {  
    algorithm hmac-md5;  
    secret "LcZ+GcGU+VX4jAyCVtXnIA==";  
};
```

Man muss dann nur noch auf diese „/etc/bind/rndc-key“-Datei verweisen, damit man den Schlüssel mit seinem Name ansprechen kann.

```
include "/etc/bind/named-rndc.conf";
```

*Verzeichnis evtl. anders*

```
include "/etc/bind/ddns.key";  
  
zone "example.org" {  
    type master;  
    notify no;  
    file "/var/cache/bind/db.example.org";  
    allow-update { key DDNS_UPDATE; };  
};  
  
zone "2.168.192.in-addr.arpa" {  
    type master;  
    notify no;  
    file "/var/cache/bind/db.192.168.2";  
    allow-update { key DDNS_UPDATE; };  
};
```

<https://wiki.debian.org/DDNS>

### 1.2.3 Redundanz

#### primäre DNS-Server (Master) / sekundäre DNS-Server (Slave)

Mehrere DNS-Server dienen zur Ausfallsicherheit. Wenn einer ausfällt, sind andere DNS-Server immer noch erreichbar.

Wenn es mehrere DNS-Server in einem Netzwerk gibt, dann gibt es immer **einen** primären DNS-Server / Master DNS-Server und **mehrere** sekundäre DNS-Server / Slave-DNS-Server. Es gibt keine Maximalanzahl an Slave-Servern.

Damit der sekundäre Server, auch wenn er zuerst hochfährt, die Zonendateien zur Verfügung hat, macht man ein Backup der Zonendatei.



```
zone "example.com" {
    type master;
    file "/etc/bind/db/example.com";
    allow-transfer { 11.22.33.55; };
};
```

*Primärer Server*

*Sekundäre(r) Server*

```
zone "example.com" {
    type slave;
    file "bak/example.com";
    masters { 11.22.33.44; };
};
```

*Sekundärer Server*

*Sicherheitskopie*

*Adresse des primären Servers*

```
example.com. IN SOA ns.example.com. hostmaster.example.com. (
    2009102201      ; Seriennummer
    1d              ; Slave-Refresh (1 Tag)
    6h              ; Slave-Retry (6 Stunden)
    2w              ; Slave-Expiry (2 Wochen)
    1h              ; TTL für nicht vorhandene Namen
    )
```

Zum Aktualisieren ist wichtig, dass sich die Seriennummer der Master-Zonendatei ändert, damit der Slave-Server erkennt, dass sich die Datei geändert hat.

**Seriennummer** Die Seriennummer muss jedes Mal erhöht werden, wenn sich an den Zoneninformationen etwas ändert. Die sekundären DNS-Server rufen in periodischen Abständen das SOA-Record ab und prüfen, ob die Seriennummer höher ist als die letzte, die sie gesehen haben; falls ja, wird ein Zonentransfer ausgelöst. Siehe hierzu auch Abschnitt 7.2.

**Refresh-Zeit** Gibt den Abstand an, in dem der sekundäre DNS-Server prüft, ob seine Daten noch aktuell sind. Die hier angegebene Zeit von 1 Tag ist relativ hoch; eine sinnvolle Untergrenze wäre zum Beispiel 3 Stunden, und Werte zwischen 6 und 12 Stunden sind realistisch.

**Retry-Zeit** Wenn der sekundäre DNS-Server den primären DNS-Server bei einer Routineprüfung nicht erreichen kann, dann versucht er es in dem Zeitabstand wieder, den dieser Parameter angibt. Üblicherweise ist dieser Zeitabstand kürzer als die Refresh-Zeit, aber das muss nicht unbedingt so sein. Auch hier ist »6 Stunden« ein eher hoch angesetzter Wert.

**Verfallszeit** (engl. *expiry time*) Wenn der sekundäre DNS-Server so lange keinen Kontakt zum primären Server hatte, wie dieser Zeitraum angibt, dann hört er auf, Anfragen für die Zone zu beantworten. Eine Woche ist hier ein vernünftiger Wert; längere Zeiträume sind möglich, wenn die Zone sich nicht oft ändert. In jedem Fall sollte die Verfallszeit nicht kürzer sein als die Refresh-Zeit.

**TTL für nicht vorhandene Namen** Wie alle RRs haben auch Antworten der Form »Diesen Namen gibt es gar nicht« eine Haltezeit, die angibt, wie lange der Empfänger sie in seinem Cache liegen lassen soll. Diese Haltezeit wird hier angegeben.



Früher – bis BIND 8.2 – galt dieser Parameter nicht nur für die negative Haltezeit, sondern auch für die Standard-Haltezeit für RRs ohne genaue Angabe. Inzwischen gibt es dafür die \$TTL-Direktive.

Parameter	Minimum		Maximum	
	Refresh-Zeit	Retry-Zeit <sup>a</sup>	Verfallszeit	negative TTL
	1h	15m	1w	3m
	24h	6h	1000h	1d

## 1.3 Zonendatei

Hier ein Beispiel:

```
$ORIGIN example.com.          ; Daten über example.com
$TTL 3h                        ; Standard-TTL 3 Stunden
example.com. IN SOA ns.example.com. hostmaster.example.com. (
    2009102201                ; Seriennummer
    1d                        ; Slave-Refresh (1 Tag)
    6h                        ; Slave-Retry (6 Stunden)
    2w                        ; Slave-Expiry (2 Wochen)
    1h                        ; TTL für nicht vorhandene Namen
)
@ IN NS ns.example.com.        ; ein DNS-Server
@ IN NS ns.anderswo.de.        ; noch ein DNS-Server
@ IN MX 10 mail.example.com.   ; Mailserver
ns 1d IN A 10.0.0.1
mail IN A 10.0.0.2
www IN CNAME mail
```