Studienarbeit
Untersuchung von DHCPv6

Patrick Cervicek, TI5, 725132
Patrick.Cervicek@stud.fht-esslingen.de

Sven Vogt, TI5, 725480
svvoit00@fht-esslingen.de

WS04/05

Fachbereich IT
Hochschule für Technik – Fachhochschule Esslingen

Betreuer: Herbert Wiese
Herbert.Wiese@fht-esslingen.de

Inhaltsverzeichnis

1. Ziel der Studienarbeit

DHCP (Dynamic Host Configuration Protocol) ist ein Protokoll, um Netzwerkrechner dynamisch über einen zentralen Server konfigurieren zu können.
Für IPv4 basierende Netzwerke gibt es hierbei schon erprobte Implementierungen [ISC-DHCP].
Bei DHCPv6 (DHCP für IPv6 Knoten) handelt es sich um ein neues Protokoll dass vor kurzem als RFC festgesetzt wurde. Im Rahmen dieser Studienarbeit sollte DHCPv6 (gemäss [RFC3315]) hinsichtlich verfügbarer Implementierungen sowie Verwendbarkeit innerhalb der FHTE untersucht werden. Die korrekte Konfiguration des DHCPv6 Clients sollte durch einen Ping6 zur Universität Münster sowie der Auflösung eines Namens in eine IPv6 Adresse nachgewiesen werden.

2. Vorgehensweise

Um DHCPv6 anwenden zu können sind Vorkenntnisse in IPv6 unumgänglich, da DHCPv6 gezielt die Möglichkeiten von IPv6 ausnutzt (Multicast, Stateless Address Autoconfiguration, ...). Desweiteren ist ein Grundverständniss für den Aufbau von IPv6 Adressen notwendig, da eben diese in Optionen (z.B. „DNS recursive name server“) übertragen werden. Hierzu empfiehlt sich das Buch „Das neue Internetprotokoll IPv6“ von Herbert Wiese [WIESE].

Desweiteren haben wir die RFCs

- „Dynamic Host Configuration Protocol for IPv6 (DHCPv6)“ [RFC3315]
- „Stateless DHCP for IPv6“ [RFC3736]

gelesen. Denn nur mit diesen Kenntnissen konnten die Etherealmitschnitte auf Konformität mit dem DHCPv6-Protokoll überprüft werden.

Implementierungen hatten wir in Google mit den Suchworten „dhcpv6 implementation“ gesucht und sind dabei auf 3 Implementierungen gestoßen:

[DHCPV6]
[DIBBLER]
[HYCOMAT]

Die Implementierung [HYCOMAT] schlossen wir allerdings aus, da das Projekt seit Dezember 2002 nicht mehr weiterentwickelt wird. Die anderen 2 unterzogen weiteren Tests.

3. Allgemeines zu DHCPv6

Mit DHCP können Knoten über das Netzwerk konfiguriert werden. DHCP ist Server/Clientbasierter Dienst. Bei DHCP können stateless (zustandslos) und stateful (zustandsbehaftet) Daten verwaltet werden.

- Stateless
  Stateless Daten können z.B. die feste IP des DNS-/Zeitserver sein.

- Stateful
  Unter dem Begriff stateful wird die Zuteilung/Verwaltung von Ressourcen verstanden. Stateful Daten kann z.B. die einmalige Vergabe einer IP sein. Dabei wird auf dem Server eine IP-Adresse einer MAC Adresse zugeordnet. Die IP-Adresse kann dabei dynamisch vergeben werden, so dass jeder Client bei jedem Booten eine erneute IP zugeteilt bekommt. Abhängig von der MAC Adresse kann auf dem DHCP Server bereits im Vorfeld festgelegt werden, welcher DHCP Client welche IP-Adresse bekommt.

Die Zuordnung zwischen IP Adresse und Host wird durch einen DHCP Unique Identifier (DUID) und Identity Association Identifier (IAID) erreicht. Die DUID kennzeichnet einen Client oder Server. Die DUID ist dabei nicht von der Hardware abhängig bzw. die DUID ändert sich nicht wenn Hardware ausgetauscht wird. Eine IAID kennzeichnet eins von vielen möglichen Schnittstellen (Interface). Eine genaue Übersicht kann von [STRAUF] entnommen werden.
DHCPv6 benutzt als Transportprotokoll UDP. DHCPv6 Clients verwenden dabei Port 546 und die Server 547. Das Protokoll besitzt einen Basisheader, indem Optionen variabler Länge eingebunden sein können. Grundsätzlich enthalten sind der *msg-type* sowie eine *transaction-id*.
Die *msg-type* kennzeichnet den Typ der Nachricht z.B. SOLICIT, ADVERTISE, REQUEST, REPLY,...
Die *transaction-id* ist eine Nummer, die den Zusammenhang zwischen einem Frage-/Antwortpaar herstellt. Beispielsweise ist die *transaction-id* von einem SOLICIT <> ADVERTISE sowie REQUEST <> REPLY gleich.

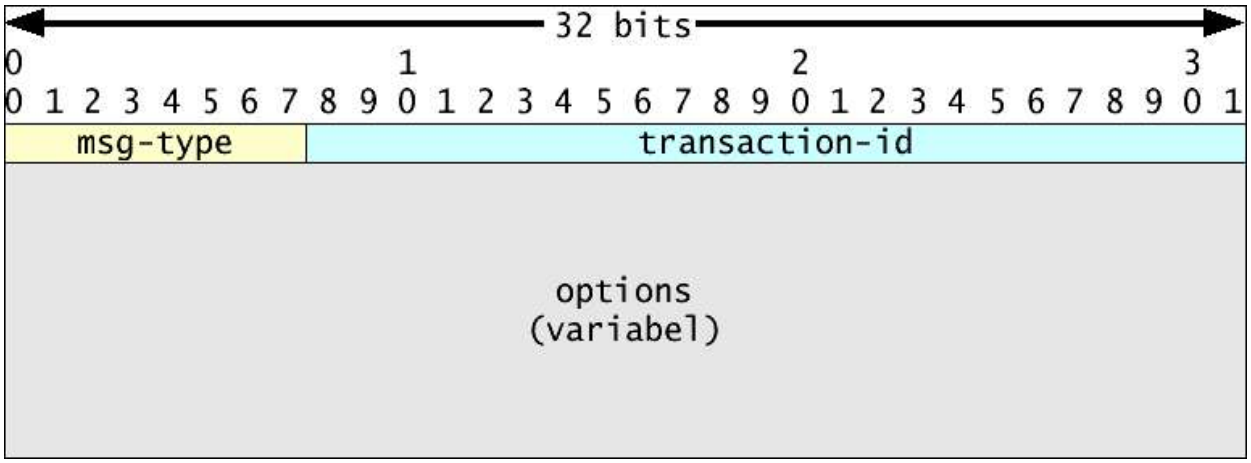

Abbildung 3.1: DHCPv6 Basisheader

Ein interessantes Merkmal bei DHCPv6:
Im Gegensatz zu DHCPv4 wird keine Option „routers“ zum Festlegen des Default Gateways verwendet. Sollte ein Default Gateway benötigt werden, muss dies in Router Advertisements verteilt werden.

DHCPv6 bietet eine Reihe an Features:

- Relay Agent Support
  Damit nicht in jedem Linksegment ein eigener DHCPv6 Server stehen muss, kann der Einsatz von Relay Agents in Erwägung gezogen werden. Relay Agents nehmen die Anfragen von Clients auf und leiten sie (mittels (RELAY_FORW) an die Unicast Adresse eines DHCPv6 Servers weiter. Der DHCPv6 Server antwortet dem Relay Agent (mittels RELAY_REPL) und dieser wiederum antwortet dem Client. Pro Linksegment sollte mindestens ein Relay Agent stehen. Es können zwischen Client und Server mehrere Relay Agents positioniert sein.

- Authentication of DHCP Messages
  Es gibt wie bei DHCPv4 auch bei DHCPv6 Sicherheitsprobleme. Die 3 wichtigsten sind:

  - Unbekannte externe DHCPv6 Server, welche den Clients falsche Adressen zuweisen (Hier vor kann man sich durch Sperren der entsprechenden Ports auf der Firewall schützen)
  - Unsachgemäß oder böswillig aufgesetzte DHCP Server im internen Netz, die die Clients konfigurieren.
  - Fremde Clients, die sich in das interne Netz hängen und automatisch eine interne Adresse erhalten.

  Schutz vor diesen Angriffen bietet der Einsatz von DHCP Authentication. Bei DHCPv6 kann man mit der Authentication Option die Herkunft einer DHCPv6 Nachricht eindeutig identifiziert werden. Außerdem ist sichergestellt, dass der Inhalt der Nachricht nicht verändert wurde.

- Dynamic DNS Update
  Der DHCP Server hat die Möglichkeit nach Vergabe einer IPv6 Adresse einen zentralen DNS-Server mit diesen Informationen zu aktualisieren. DDNS kann verwendet werden, damit der DNS Server nicht manuell konfiguriert werden muss, was bei der steigenden Anzahl von Knoten aufwändig wäre. Zu beachten ist: Bei einem Client, welcher sich mit Stateless Address Autoconfiguration eine IPv6 Adresse vergibt, muss der DDNS Mechanismus dort implementiert werden, da kein DHCPv6 Server verwendet wird.

- Confirm
  Ein Client kann bei dem Wiedereinschalten überprüfen, ob er noch am selben Linksegment angeschlossen ist. Dazu sendet er eine CONFIRM Nachricht an den bekannten Server um die bereits zugewiesene Adresse zu validieren. Meldet sich der Server nicht bzw. verneint er die Anfrage, so geht der Client in den ganz normalen

- Rekonfiguration
  Ein Server kann eine RECONFIGURE Nachricht an bestimmte Clients schicken, um Ihn zur Neukonfiguration zu bewegen. In der RECONFIGURE Nachricht kann über enthaltene Optionen mitgeteilt werden, welche Daten erneut beim Server abgerufen werden sollen. Sendet der DHCP Server eine 'IA Address' Option mit, so beantragt der Client mittels RENEW eine neue IP Adresse. Ansonsten bezieht der Client die neuen Konfigurationsdaten mittels INFORMATION-REQUEST. Aus Sicherheitsgründen ist die Verwendung von DHCP Authentication bei einer Rekonfiguration zwingend notwendig. (Schutz vor Denial-of-Service Attacken)

4. IPv6 Stateless Address Autoconfiguration (SAA)

Bei IPv6 können sich Knoten auch ohne DHCP Server eine IPv6 Adresse vergeben. Dabei verwendet ein IPv6-Knoten seine MAC Adresse.
Jeder Client verfügt dabei über eine sogenannte LLA (Linklocale Adresse, nur gültig im angeschlossenen Linksegment) und setzt sich aus der Präfix fe80::/64 sowie der MAC-Adresse zusammen. Das 2. Bit des höchstwertigsten Byte der MAC-Adresse wird dabei getoggelt.
Bei einer MAC Adresse 00:30:48:28:29:BB würde die LLA fe80::230:48ff:fe28:29bb erzeugt.
Befindet sich noch ein Router im selben Linksegement, welcher periodisch ICMPv6 Router Advertisements mit der globalen Präfix versendet, so kann der IPv6-Knoten sich zusätzlich auch mit einer Globalen Adressen konfigurieren.

Im folgenden Beispiel verteilt ein Router die Präfix 3ffe:400:3d0::/64

```
Internet Protocol Version 6
    Version: 6
    ....
    Next header: ICMPv6 (0x3a)
    Hop limit: 255
    Source address: fe80::204:75ff:fe74:463a
    Destination address: ff02::1 (ff02::1) /* Alle_Knoten  */
Internet Control Message Protocol v6
    Type: 134 (Router advertisement)
    Code: 0
    Checksum: 0x6f40 (correct)
    Cur hop limit: 64
    Flags: 0x00
    Router lifetime: 30
    Reachable time: 0
    Retrans time: 0
    ICMPv6 options
          Type: 3 (Prefix information)
          Length: 32 bytes (4)
          Prefix length: 64
          ...
          Prefix: 3ffe:400:3d0::
```

In unserem Beispiel würde die Adresse 3ffe:400:3d0::230:48ff:fe28:29bb erzeugt.

In kleineren Netzen reicht es übrigens aus, das Netz mit Stateless Address Autoconfiguration zu konfigurieren und weitere Parameter (z.B.: DNS Server) über stateless DHCP zu konfigurieren. Die (LLA-)IPv6 Adresse des Default Gateways wird dabei über die Router Advertisements entnommen.

Bei DHCPv6 wird die Kommunikation zwischen Client und Server immer mit der LLA durchgeführt. Dies hat den Vorteil, dass der Server dem Client direkt antworten kann (ADVERTICE, REPLY) und kein Broadcast (den es bei IPv6 sowieso nicht mehr gibt) verwenden muss.

5. Testinstallation mit DHCPv6 @ Sourceforge

Wir hatten DHCPv6 heruntergeladen, entpackt, kompiliert und installiert. Einmal auf dem Server und einmal auf dem Client. Die ersten Test wurden auf Patricks Heimnetzwerk durchgeführt. Das Netzwerk bestand aus zwei Linux-Rechnern, der einer als Server der andere als Client. Gleichzeitig haben wir auf dem Server-Rechner mit Hilfe von Ethereal Protokoll mitschnitte gemacht.

Installationsschritte:

```
tar xzfv dhcp-0.10.tgz
cd dhcp-0.19
./configure
make
make install
```

Nach der Installation standen folgende, für uns relevante Dateien zur Verfügung:

`dhcp6c.conf`

Datei zum Konfigurieren des Clients. Alle Versuche irgend etwas einzustellen brachten keine Erfolge.

`dhcp6s.conf`

Datei zum Konfigurieren des Servers. Auch durch Änderungen in dieser Datei konnten die Knoten nicht sinnvoll konfiguriert werden.

Erste Testläufe waren vielversprechend, es konnte eine Art „Kommunikation“ zwischen Server und Client festgestellt werden. Allerdings bestand die Kommunikation nur darin, dass der Client sich „vorstellte, ich bin XY“, Serverantwort war „freut mich, du bist XY“. Änderungen an den oben genannten Konfigurationsdateien brachten keine Verbesserung. Nach weiteren erfolglosen Versuchen entschieden wir uns, weitere Tests mit der Dibbler-Implementierung vorzunehmen in der Hoffnung, bessere Ergebnisse zu bekommen. Anbei ein Beispiel dieser sinnlosen Kommunikation zwischen Server und Client, da praktisch keine verwertbaren Informationen vorhanden waren.

Etherealmitschnitt Sourceforge-Implementierung:

```
No.     Time        Source                Destination           Protocol Info
     69 3.356101    fe80::208:54ff:fe0a:eb81 ff02::1:2             DHCPv6   Information-request

Ethernet II, Src: 00:08:54:0a:eb:81, Dst: 33:33:00:01:00:02
    Destination: 33:33:00:01:00:02 (IPv6-Neighbor-Discovery_00:01:00:02)
    Source: 00:08:54:0a:eb:81 (Netronix_0a:eb:81)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 36
    Next header: UDP (0x11)
    Hop limit: 1
    Source address: fe80::208:54ff:fe0a:eb81 (fe80::208:54ff:fe0a:eb81)
    Destination address: ff02::1:2 (ff02::1:2)      // Alle DHCP-Server
User Datagram Protocol, Src Port: dhcpv6-server (547), Dst Port: dhcpv6-server (547)
    Source port: dhcpv6-server (547)
    Destination port: dhcpv6-server (547)
```

```
    Length: 36
    Checksum: 0x13c0 (correct)
DHCPv6
    Message type: Information-request (11)
    Transaction-ID: 0x0039c294
    Client Identifier
        option type: 1
        option length: 14
        DUID type: link-layer address plus time (1)
        Hardware type: 1
        Time: 154637009
        Link-layer address
    Elapsed time
        option type: 8
        option length: 2
        elapsed-time: 0 sec

No.     Time        Source                Destination           Protocol Info
     70 3.356787    fe80::204:75ff:fe84:29d fe80::208:54ff:fe0a:eb81 DHCPv6   Reply[Short Frame]

Ethernet II, Src: 00:04:75:84:02:9d, Dst: 00:08:54:0a:eb:81
    Destination: 00:08:54:0a:eb:81 (Netronix_0a:eb:81)
    Source: 00:04:75:84:02:9d (3Com_84:02:9d)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 92
    Next header: UDP (0x11)
    Hop limit: 64
    Source address: fe80::204:75ff:fe84:29d (fe80::204:75ff:fe84:29d)
    Destination address: fe80::208:54ff:fe0a:eb81 (fe80::208:54ff:fe0a:eb81)
User Datagram Protocol, Src Port: dhcpv6-client (546), Dst Port: dhcpv6-client (546)
    Source port: dhcpv6-client (546)
    Destination port: dhcpv6-client (546)
    Length: 92
    Checksum: 0xa4bd
DHCPv6
    Message type: Reply (7)
    Transaction-ID: 0x0039c294
    Client Identifier
        option type: 1
        option length: 14
        DUID type: link-layer address plus time (1)
        Hardware type: 1
        Time: 154637009
        Link-layer address
    Server Identifier
        option type: 2
        option length: 14
        DUID type: link-layer address plus time (1)
        Hardware type: 1
        Time: 154629517
        Link-layer address
```

Wie im Mitschnitt zu sehen ist, fehlen alle möglichen Informationen. Weder beantragt der Client eine IPv6-Adrese noch andere Konfigurationsdaten. Ausserdem sind Client und Server Port falsch.Aufgrund weiterer Fehler (**segmentation fault)** beendeten wir die Tests mit dieser Implementierung.

6. Testinstallation Dibbler-Implementierung

6.1 Sytemvorrausetzungen

Die Dibbler-Implementierung war die zweite Lösung, die wir durch Internetrecherche fanden. Diese Implementierung schien auf den ersten Blick besser zu sein - das letzte Release kam November 2004 (im Vergleich zur Sourceforge Implementierung März 2004). Eine Windows Variante ist ebenfalls verfügbar. Bevor die erneute Testphase beginnen konnte, wechselten wir unser Versuchsfeld von Patricks Heimnetz zum KT-Laborraum im HZE. Dort gab es ebenfalls IPv6-fähige Rechner, bzw. ein IPv6-fähiges Netzwerk inkl. Router und „Tunnel “ zu einem IPv6-Router in der Universität Münster.
Ein erster Schritt war das Aufsetzen von zwei Linux-Systemen für die DHCPv6-Kommunikation zwischen Server und Client. Wir haben uns für die aktuellste Fedora-Version entschieden - Fedora Core 3. Alle neueren Linux-Systeme haben bereits das IPv6 Protokoll – meist als Kernelmodul - implementiert. Um eine möglichst einfache Testumgebung zu bekommen ohne die bestehende Infrastruktur zu beeinflussen, benutzten wir VMware. VMware bietet die Möglichkeit mehrere Betriebssysteme auf einem Windows Rechner parallel laufen zu lassen. Somit war es uns möglich das DHCPv6 Testnetzwerk (Server/Client) auf nur einem Rechner zu testen. Für die Analyse der Netzwerkpakete wurde Ethereal zusammen mit den von dem Programm tcpdump erzeugten Dateien eingesetzt. Das Programm tcpdump wurde auf dem client gestartet, bevor die DHCP Dienste gestartet wurden. Nach jedem Testlauf wurde die tcpdump Datei per scp aus VMware herauskopiert und mit Ethereal analysiert. Ethereal konnte nicht direkt eingesetzt werden, da es nicht den Verkehr innerhalb von VMware abhorchen kann. Ein weiterer Vorteil von VMware war es, dass nach dem Projekt sämtliche Installationen rückgängig gemacht werden konnten und das Anfangssystem unverändert wieder zur Verfügung stand. Man kann sich die tcpdump Dateien auch unter [DUMPS] herunterladen.

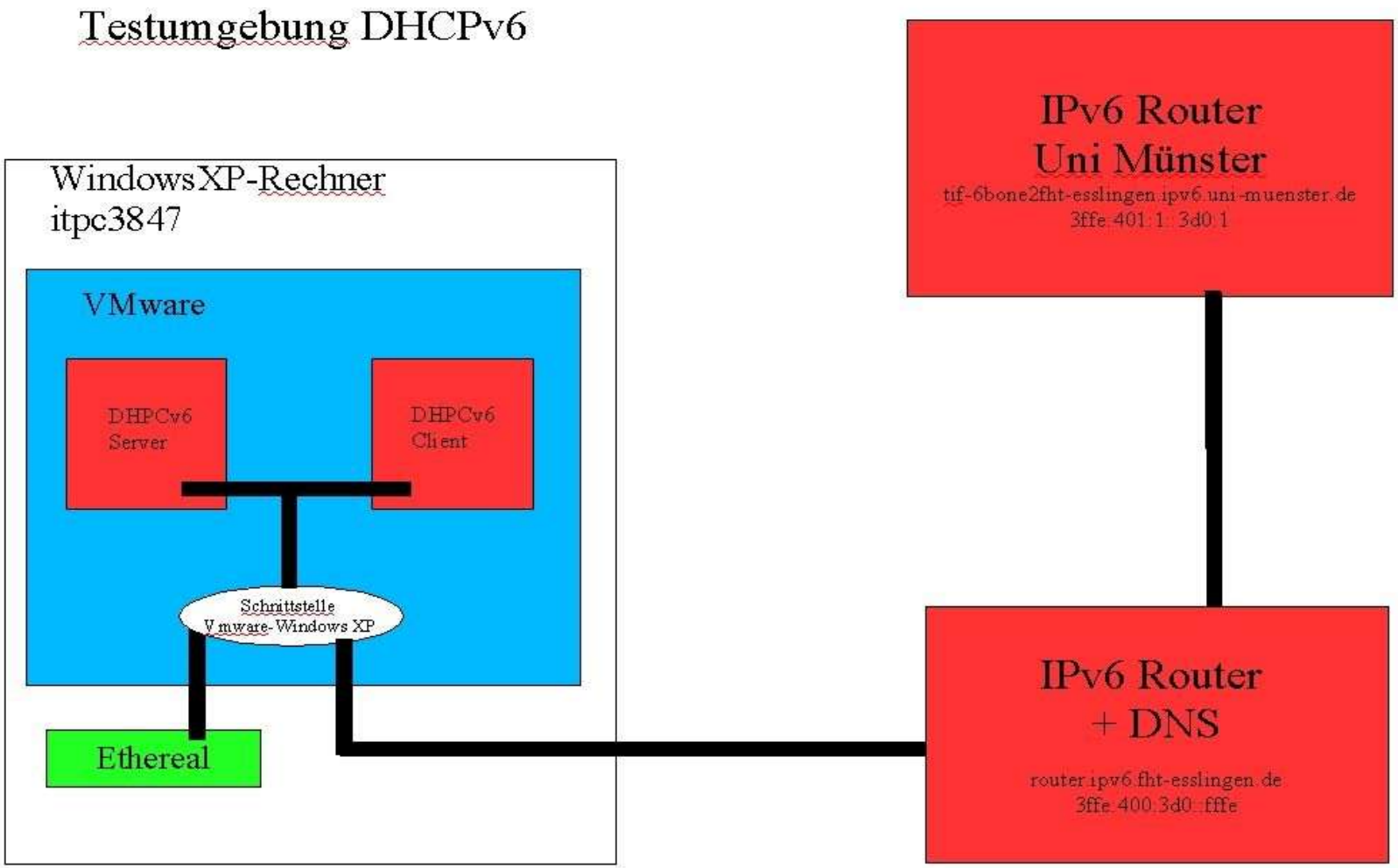

Abbildung 5.1: Versuchsaufbau KT-Labor

6.2 Installation Dibbler-Implementierung

Die Dibbler-Implementierung entstand im Rahmen einer Master Thesis. Der Autor dieser Implementierung war unzufrieden mit der Tatsache, dass es viele DHCPv6 Implementierungen ohne Namen gab. Wollte er in seiner Master Thesis auf eine DHCPv6 Implementierung hinweisen, so musste er immer benennen ob er nun die Sourceforge Implementierung oder KAME Implementierung meint. Er entschloss sich daher seiner Implementierung einen Namen zu geben: 'dibbler' (nach einer Figur in dem Roman „Scheibenwelt“von Terry Pratchett)
Für die Installation der Dibbler-Implementierung benötigten wir die Datei die unter [DIBBLER] zum Download zur Verfügung steht.

Installationsschritte:

`tar xzfv dibbler-0.3.0-RC2-linux.tar.gz` (Fertig kompilierte Version)

Nach diesem Schritt hatten wir folgende konfigurierbare Dateien zur Verfügung:

`server.conf`
`server-stateless.conf` (für den Stateless-Betrieb)
`client.conf`
`client-stateless.conf` (für den Stateless-Betrieb)

Wichtig: Die Konfiguration muss in `/var/lib/dibbler/` hinterlegt werden. Die DUID von Server und Client sind dort auch hinterlegt. Laut [RFC3315] darf die DUID nämlich nicht von der verwendeten Hardware abhängen, sondern muss einmalig erzeugt und dauerhaft gespeichert werden.

6.3 Versuche mit Dibbler

6.3.1 ohne Änderung der Konfigurationsdaten

Diese Dateien waren bereits allgemein konfiguriert und wir konnten einen Test durchführen. Eine Kommunikation sowie Ergebnisse waren sofort erkennbar:

Der Client sucht DHCPv6-Server über Anfrage mit der IPv6-Multicastadresse für alle DHCP Agenten (FF02::1:2) und der Server antwortet (Anfrage/Antwort bzw. Solicitation/Advertisement). Danach holt sich der Client die gewünschten Konfigurationsdaten (zweites Anfrage-/Antwort Paar, bzw. Request/Reply) wie z.B. seine globale IPv6-Adresse.

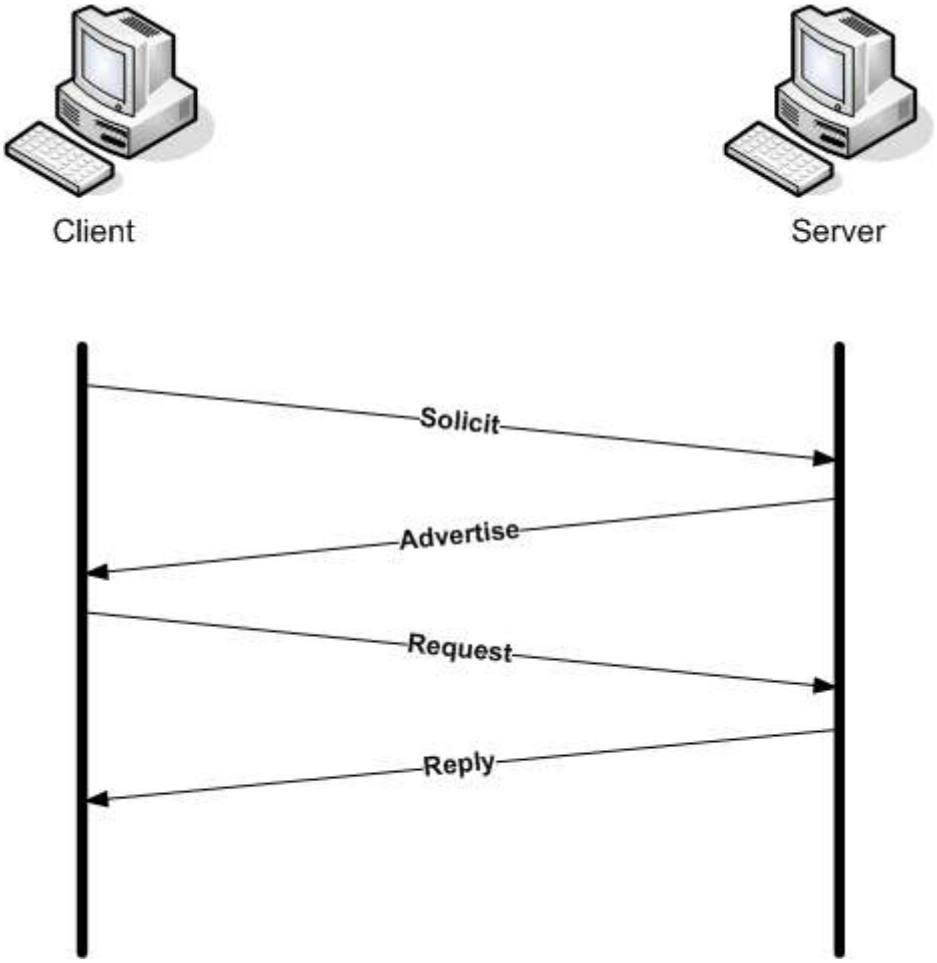

Abbildung 6.3.1: DHCPv6 Konfiguration

Etheralmitschnitt dhcpv6-dibbler-1.txt

```
//Clientsuche nach DHCPv6-Servern
No.     Time        Source                Destination           Protocol Info
      1 0.000000    fe80::20c:29ff:fe6c:1679 ff02::1:2          DHCPv6  Solicit

Ethernet II, Src: 00:0c:29:6c:16:79, Dst: 33:33:00:01:00:02
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 88
    Next header: UDP (0x11)
    Hop limit: 1
    Source address: fe80::20c:29ff:fe6c:1679 (fe80::20c:29ff:fe6c:1679)   //Clientadresse
    Destination address: ff02::1:2 (ff02::1:2)                           //Alle DHCPv6-Server
User Datagram Protocol, Src Port: 546 (546), Dst Port: 547 (547)   //verwendete Ports
DHCPv6
    Message type: Solicit (1)               //Nachrichtentyp wird hier festgelegt
    Transaction-ID: 0x005a6f04              //Dient zur Identifikation der Nachrichten
    Client Identifier                       //Benutzt DUID um Server und Client zu unterscheiden
        option type: 1
        option length: 14
        DUID type: link-layer address plus time (1)
        Hardware type: 0
        Time: 1102453397
        Link-layer address
    Identify Association
        option type: 3
        option length: 40
        IAID: 2                     //Identifiziert verschiedene Interfaces pro Client
        T1: 4294967295                      //Zeit nach der der Client zum bekannten DHCPv6-Server
connected
        T2: 4294967295                      // Zeit nach der der Client zum allen DHCPv6-Server
connected
        IA Address
            option type: 5
            option length: 24
            IPv6 address: 2000::123
            Preferred lifetime: infinity
            Valid lifetime: infinity
    Elapsed time
        option type: 8
        option length: 2
        elapsed-time: 3 sec
    Option Request                  //Liste der gewünschten Optionen vom Client an Server
        option type: 6
        option length: 4
        Requested Option code: DNS recursive name server (23)
        Requested Option code: Domain Search List (24)
               ...
```

```
//Serverantwort
No.     Time        Source                Destination           Protocol Info
      4 0.869604    fe80::20c:29ff:fe52:eee6 fe80::20c:29ff:fe6c:1679 DHCPv6   Advertise

Ethernet II, Src: 00:0c:29:52:ee:e6, Dst: 00:0c:29:6c:16:79
Internet Protocol Version 6
            ...
    Source address: fe80::20c:29ff:fe52:eee6              //Serveradresse
    Destination address: fe80::20c:29ff:fe6c:1679         //Clientadresse
User Datagram Protocol, Src Port: 547 (547), Dst Port: 546 (546)
DHCPv6
    Message type: Advertise (2)
    Transaction-ID: 0x005a6f04
    Client Identifier
        option type: 1
        option length: 14
        DUID type: link-layer address plus time (1)
        Hardware type: 0
        Time: 1102453397
        Link-layer address
    Identify Association
            ...
        IA Address
            IPv6 address: 2000::2e                 //Ipv6-Adresse für den Client
               ...
    DNS recursive name server
        option type: 23
        option length: 32
        DNS servers address: 2000::100             //DNS-Server, konfiguriert in der server.conf
        DNS servers address: 2000::101             //DNS-Server, konfiguriert in der server.conf
    Domain Search List
        option type: 24
        option length: 31
        DNS Domain Search List
    Server Identifier                              //Identifizierung vom Server über DUID
        option type: 2
        option length: 14
        DUID type: link-layer address plus time (1)
        Hardware type: 0
        Time: 1102451734
        Link-layer address
    Preference                              //beeinflussen die Auswahl des Server vom Client [1]
        option type: 7
        option length: 1
        pref-value: 0
```

[1] Die Option „Preference“ kann vom Server mitgeschickt werden und beeinflusst den Client bei der Auswahl des zu wählenden Servers. Sollten z.B. mehrere Server antworten, so wählt der Client den Server mit der höchsten „server preference“.

```
//Client Anforderung Konfigurationsdaten
No.     Time        Source                Destination           Protocol Info
      5 2.413820    fe80::20c:29ff:fe6c:1679 ff02::1:2            DHCPv6   Request

Ethernet II, Src: 00:0c:29:6c:16:79, Dst: 33:33:00:01:00:02
Internet Protocol Version 6
            ...
    Source address: fe80::20c:29ff:fe6c:1679              //Clientadresse
    Destination address: ff02::1:2                 //Serveradresse
User Datagram Protocol, Src Port: 546 (546), Dst Port: 547 (547)
DHCPv6
    Message type: Request (3)
    Transaction-ID: 0x0073c75a
    Client Identifier
            ...
    Identify Association
        option type: 3
        option length: 40
        IAID: 2
        T1: 4294967295
        T2: 4294967295
        IA Address
            option type: 5
            option length: 24                             //verschiedene Optionen
            IPv6 address: 2000::123                       //gewünschte IPv6-Adresse
            Preferred lifetime: infinity                  //bevorzugte Lebenszeit der IPv6-Adresse
            Valid lifetime: infinity                      //gültige Lebenszeit der IPv6-Adresse
     Elapsed time                                         // „Vergangene Zeit“ [1]
        option type: 8
        option length: 2
        elapsed-time: 3 sec
     Option Request
        option type: 6
        option length: 20
        Requested Option code: DNS recursive name server (23)
        Requested Option code: Domain Search List (24)
            ...
     Server Identifier                                          // Identifizierung des Servers
über DUID
        option type: 2
        option length: 14
        DUID type: link-layer address plus time (1)
        Hardware type: 0
        Time: 1102451734
        Link-layer address
```

[1] Mit der Option „Elapsed time“ zeigt der Client an, wie lange er schon versucht, eine gültige Konfiguration durch DHCP zu erlangen. Dies ist durch [RFC3315] vorgeschrieben. Je nachdem wie die DHCPv6 Server konfiguriert sind, kann ein anderer (Backup-) Server dem Client bei einem Request antworten, sollte die „Elapsed time“ anzeigen, dass ein Client schon längere Zeit versucht eine gültige Konfiguration vom (Master-) Server zu erlangen.

```
//Server-Antwort mit den Konfigurationsdaten
No.     Time        Source                Destination           Protocol Info
      6 2.913745    fe80::20c:29ff:fe52:eee6 fe80::20c:29ff:fe6c:1679 DHCPv6   Reply


Ethernet II, Src: 00:0c:29:52:ee:e6, Dst: 00:0c:29:6c:16:79
Internet Protocol Version 6
             ...
    Source address: fe80::20c:29ff:fe52:eee6              //Server-Adresse
    Destination address: fe80::20c:29ff:fe6c:1679         //Client-Adresse
User Datagram Protocol, Src Port: 547 (547), Dst Port: 546 (546)
DHCPv6
    Message type: Reply (7)
    Client Identifier
             ...
    Identify Association
        option type: 3
        option length: 74
        IAID: 2
        T1: 1000
        T2: 2000
        IA Address
            option type: 5
            option length: 24
            IPv6 address: 2000::2e                //Ipv6-Adresse für den Client
            Preferred lifetime: 1800              //bevorzugte Lebenszeit der IPv6-Adresse
            Valid lifetime: 3600                  //gültige Lebenszeit der IPv6-Adresse
        Status code
            option type: 13
            option length: 30
            Status Code: Success (0)
            Status Message: All addresses were assigned.
    Server Identifier                             //Identifizierung des Servers durch DUID
        option type: 2
        option length: 14
        DUID type: link-layer address plus time (1)
        Hardware type: 0
        Time: 1102451734
        Link-layer address
    DNS recursive name server
        option type: 23
        option length: 32
        DNS servers address: 2000::100            //DNS-Server
        DNS servers address: 2000::101            //DNS-Server
    Domain Search List
        option type: 24
        option length: 31
        DNS Domain Search List
```

Man sieht deutlich das der Client IPv6-Adresse 2000::123 haben möchte, vom Server allerdings 2000::2e zugewiesen bekommt.

6.3.2 mit Anpassung an das FHTE-Netz

Für den nächsten Test änderten wir die `server.conf` sowie die `client.conf` entsprechend dem Testnetzwerk, d.h. der IPv6-Adressraum und die entsprechend DNS-Server wurden an die Netzwerkumgebung der FHTE angepasst. Wichtig war es nun zu sehen, ob die von uns gewünschten Konfigurationsdaten benutzt werden. In der `server.conf` wurde ein bestimmter Adressrange (3ffe:400:3d0::ff00-3ffe:400:3d0::ff10) und DNS-Server (3ffe:400:3d0::fffe) angegeben, sowie eine Option zur festen Vergabe einer IPv6 Adresse für einen Client mit einer bestimmten MAC Adresse.
(IPv6-Adresse:3ffe:400:3d0:0:204:75ff:fe74:468c).

geänderte `server.conf`

```
log-level 8                    //ausführliche Meldungen anzeigen
log-mode short

iface eth0 {                   //Konfiguration Interface eth0
 T1 1000                       //Werte setzen für T1 und T2
 T2 2000

class {
        accept-only fe80::20c:29ff:fe6c:1679              //Statische Konfiguration
        pool 3ffe:400:3d0::0204:75ff:fe74:468c            //für Host mit MAC 00:0C:29:6C:16:79
      }

class {
         pool 3ffe:400:3d0::ff00-3ffe:400:3d0::ff10       //Adresspool für sonstige Hosts
      }
option dns-server 3ffe:400:3d0::fffe                      //Setze DNS-Server
option domain 'fht-esslingen.de', 'rznt.rzdir.fht-esslingen.de'    //Domain Suchliste

}
```

geänderte `client.conf`

```
log-level 8
log-mode short

iface eth0
{

# Wenn keine IPv6-Adresse benötigt wird (Stateless Betrieb)   (Default: stateful)
#   stateless

# Rapid Commit Option (Default: aus)
#   rapid-commit 1

   option dns-server
   option domain

# Weitere Optionen, welche der Client anfordern kann.
#   option ntp-server
#   option time-zone
#   option sip-server
#   option sip-domain
#   option nis-server
#   option nis-domain
#   option nis+-server
#   option nis+-domain

}
```

Etherealmitschnitt im FHTE-Netz:

```
//Client-Anfrage(Serversuche), „Hallo hier bin ich!“
No.     Time        Source                Destination           Protocol Info
      7 34.763032   fe80::20c:29ff:fe6c:1679 ff02::1:2            DHCPv6   Solicit

Ethernet II, Src: 00:0c:29:6c:16:79, Dst: 33:33:00:01:00:02
    Destination: 33:33:00:01:00:02
    Source: 00:0c:29:6c:16:79
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 108
    Next header: UDP (0x11)
    Hop limit: 1
    Source address: fe80::20c:29ff:fe6c:1679              //Quelladresse: Client (FHTE-Netz)
    Destination address: ff02::1:2                        //Zieladresse Alle DHCPv6-Server
             ...
User Datagram Protocol, Src Port: 546 (546), Dst Port: 547 (547)   //benutzte Ports
    Source port: 546 (546)
    Destination port: 547 (547)
    Length: 108
    Checksum: 0x192e (correct)
DHCPv6
    Message type: Solicit (1)                             //Art der Nachricht: hier eine Anfrage
    Transaction-ID: 0x00b6036f                            //Identifizierung der Nachricht
    Client Identifier                                     //Identifizierung des Client durch DUID
        option type: 1
        option length: 14
        DUID type: link-layer address plus time (1)
        Hardware type: 0
        Time: 1102453397
        Link-layer address
    Identify Association            //Identifizierung verschiedener Interfaces je Client-DUID
        option type: 3
        option length: 40
        IAID: 2
        T1: 4294967295                              //nach dieser Zeit connect zu bestimmten Server
        T2: 4294967295                              //nach dieser Zeit connect zu allen bekannten
Severn
        IA Address
            option type: 5
            option length: 24
            IPv6 address: ::
            Preferred lifetime: infinity
            Valid lifetime: infinity
               ...
    Option Request                          //Liste der gewünschten Optionen vom Client an Server
        option type: 6
        option length: 20
        Requested Option code: DNS recursive name server (23)
        Requested Option code: Domain Search List (24)
```

```
//Server-Antwort
No.     Time        Source                Destination           Protocol Info
     11 34.944497   fe80::20c:29ff:fe52:eee6 fe80::20c:29ff:fe6c:1679 DHCPv6   Advertise

Ethernet II, Src: 00:0c:29:52:ee:e6, Dst: 00:0c:29:6c:16:79
    Destination: 00:0c:29:6c:16:79
    Source: 00:0c:29:52:ee:e6
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 201
    Next header: UDP (0x11)
    Hop limit: 64
    Source address: fe80::20c:29ff:fe52:eee6                    //Quelladresse: Server (Router)

    Destination address: fe80::20c:29ff:fe6c:1679          //Zieladresse: Client (FHTE-Netz)
User Datagram Protocol, Src Port: 547 (547), Dst Port: 546 (546)
              ...
DHCPv6
    Message type: Advertise (2)             //Art der Nachricht: Antwort auf Anfrage
    Transaction-ID: 0x00b6036f              //Identifizierung der Nachricht
    Client Identifier                       //Identifizierung des Clientrechners durch DUID
        option type: 1
        option length: 14
        DUID type: link-layer address plus time (1)
        Hardware type: 0
        Time: 1102453397
        Link-layer address
    Identify Association            //Identifizierung verschiedener Interfaces je Client-DUID
        option type: 3
        option length: 74
        IAID: 2
        T1: 1000      //nach dieser Zeit connect zu einem bestimmten Server(in server.conf
festgelegt)
        T2: 2000      //nach dieser Zeit connect zu allen bekannten Servern(in server.conf
festgelegt)
        IA Address
            option type: 5
            option length: 24
            IPv6 address: 3ffe:400:3d0:0:204:75ff:fe74:468c       //neue IPv6-Adresse für Client
            Preferred lifetime: 1800        //bevorzugte Lebenszeit der IPv6-Adresse
            Valid lifetime: 3600            //gültige Lebenszeit der IPv6-Adresse
        Status code
            option type: 13
            option length: 30
            Status Code: Success (0)
            Status Message: All addresses were assigned.
    DNS recursive name server
        option type: 23
        option length: 16
        DNS servers address: 3ffe:400:3d0::fffe    //FHTE-Adresse DNS-Server (in server.conf
gesetzt)    Domain Search List
        option type: 24
        option length: 46
        DNS Domain Search List
    Server Identifier                       //Identifizierung des Servers durch DUID
        option type: 2
        option length: 14
        DUID type: link-layer address plus time (1)
        Hardware type: 0
        Time: 1102545042
        Link-layer address
    Preference                              //Server wird dadurch vom Client bevorzugt
        option type: 7
        option length: 1
        pref-value: 0
```

```
//Client-Konfigurationsanforderung
No.     Time        Source                Destination           Protocol Info
     12 36.789294   fe80::20c:29ff:fe6c:1679 ff02::1:2            DHCPv6   Request
               ...
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 122
    Next header: UDP (0x11)
    Hop limit: 1
    Source address: fe80::20c:29ff:fe6c:1679              //Quelladresse: Client
    Destination address: ff02::1:2                 //Zieladresse: alle DHCP-Server
User Datagram Protocol, Src Port: 546 (546), Dst Port: 547 (547)
               ...
DHCPv6
    Message type: Request (3)                          //Art der Nachricht: Anforderung Konfigdaten
    Transaction-ID: 0x00f470cd                         //Identifizierung der Nachricht
    Client Identifier                         //Identifizierung des Client durch DUID
        option type: 1
        option length: 14
        DUID type: link-layer address plus time (1)
        Hardware type: 0
        Time: 1102453397
        Link-layer address
    Identify Association                      //Identifizierung mehrerer Interfaces je Client-DUID
        option type: 3
        option length: 40
        IAID: 2
        T1: 4294967295
        T2: 4294967295
        IA Address
               ...
            IPv6 address: ::                  //Anforderung irgendeiner IPv6-Adresse
            Preferred lifetime: infinit       //bevorzugte Lebenszeit der Ipv6-Adresse
            Valid lifetime: infinity          //gültige  Lebenszeit der IPv6-Adresse
    Option Request
        option type: 6
        option length: 20
        Requested Option code: DNS recursive name server (23)
        Requested Option code: Domain Search List (24)
     Server Identifier                        //Identifiziert den gewünschten Kommunikationsserver
        option type: 2
        option length: 14
        DUID type: link-layer address plus time (1)
        Hardware type: 0
        Time: 1102545042
        Link-layer address
```

Man sieht hier das Anforderungspaket für Konfigurationsdaten vom Client zum Server. Es ist wieder an alle gerichtet, wobei der Client eigentlich den Server kennen müsste. Durch den Server Identifier bearbeitet auch nur der Server, der geantwortet hat, die anderen bekommen zwar das Paket, machen aber nichts damit. Um eine direkte Kommunikation mit dem bekannten Server zu erzwingen, muss man in der `client.conf` die Option `unicast 1` einfügen.

```
//Server-Konfig.-Daten-Antwort
No.     Time        Source                Destination           Protocol Info
     13 36.964946   fe80::20c:29ff:fe52:eee6 fe80::20c:29ff:fe6c:1679 DHCPv6   Reply

Ethernet II, Src: 00:0c:29:52:ee:e6, Dst: 00:0c:29:6c:16:79
    Destination: 00:0c:29:6c:16:79
    Source: 00:0c:29:52:ee:e6
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Source address: fe80::20c:29ff:fe52:eee6              //Quelladresse: Server
    Destination address: fe80::20c:29ff:fe6c:1679   //Zieladresse: Client
User Datagram Protocol, Src Port: 547 (547), Dst Port: 546 (546)

DHCPv6
    Message type: Reply (7)                          //Art der Nachricht: Antwort auf eine
Anforderung
    Transaction-ID: 0x00f470cd
    Client Identifier                                //Identifizierung des Clients durch DUID
              ...
    Identify Association                    //Identifizierung mehrerer Interfaces je Client-DUID

        option type: 3
        option length: 74
        IAID: 2
        T1: 1000
        T2: 2000
        IA Address
            option type: 5
            option length: 24
            IPv6 address: 3ffe:400:3d0:0:204:75ff:fe74:468c       //neue IPv6-Adresse für Client
            Preferred lifetime: 1800
            Valid lifetime: 3600
        Status code                          //Adresse ist im Server für den Client zugeordnet
            option type: 13
            option length: 30
            Status Code: Success (0)
            Status Message: All addresses were assigned.
    Server Identifier                        //Identifizierung des server durch DUID
              ...
    DNS recursive name server
        option type: 23
        option length: 16
        DNS servers address: 3ffe:400:3d0::fffe            //IP des DNS-Servers
    Domain Search List
        option type: 24
        option length: 46
        DNS Domain Search List
```

6.3.3 Rapid commit

Wenn Client und Server die Option 'rapid commit' zulassen, dann kann die ganze Konfigurationsprozedur von vier (SOLICIT, ADVERTISE, REQUEST, REPLY) auf zwei Pakete (REQUEST, REPLY) verkürzt werden.

Vorteil:
Der Client ist schneller für den Netzwerkbetrieb konfiguriert. Ein weiterer Vorteil ist die geringere Netzbelastung.

Nachteil:
Die 'Elapsed time' Option ist nicht mehr im SOLICIT enthalten. Somit würden Backupserver nicht mehr reagieren – die Ausfallsicherheit ist nicht mehr gegeben. (s. 6.3.1)

Um 'rapid commit' zuzulassen muss `server.conf` und `client.conf` mit `rapid-commit 1` ergänzt werden.

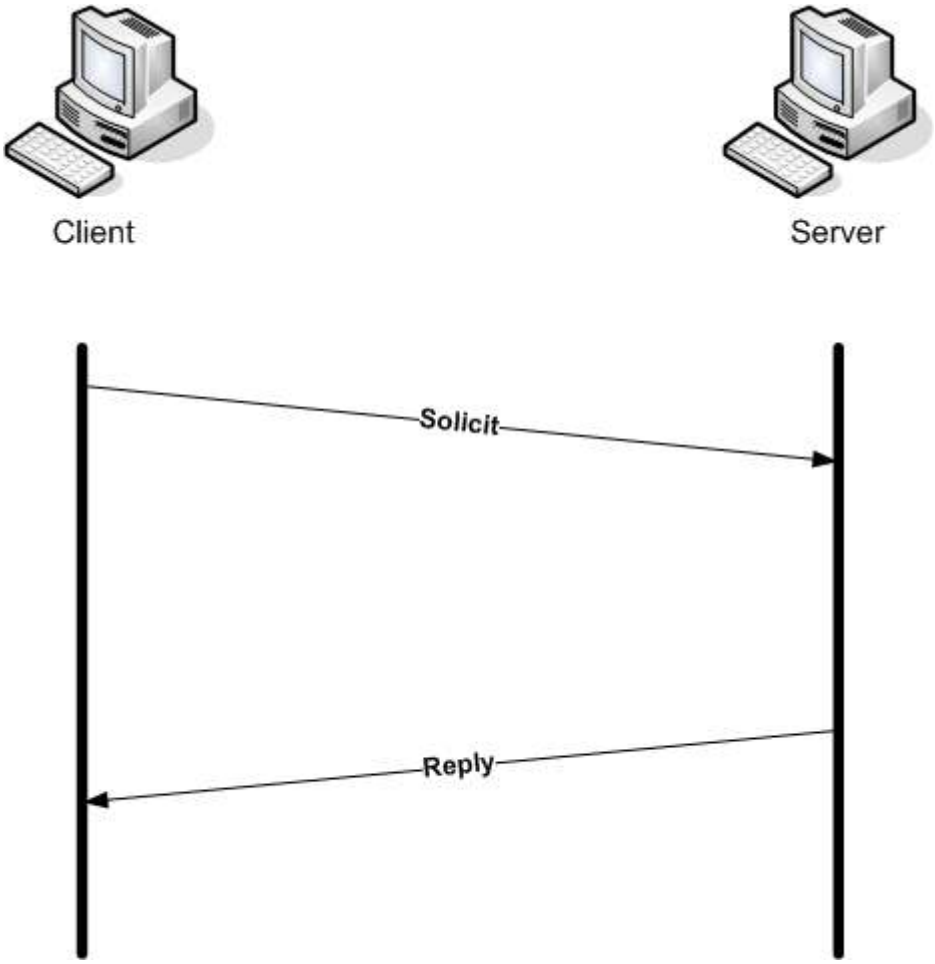

Abbildung 6.3.3: DHCPv6 Konfiguration mit Rapid Commit

Etherealmitschnitt mit rapid commit

```
No.     Time        Source                Destination           Protocol Info
      4 10.190673   fe80::20c:29ff:fe6c:1679 ff02::1:2            DHCPv6   Solicit

Internet Protocol Version 6
              ...
    Source address: fe80::20c:29ff:fe6c:1679              //Quelladresse: Client
    Destination address: ff02::1:2                        //Zieladresse: Alle DHCP-Server
User Datagram Protocol, Src Port: 546 (546), Dst Port: 547 (547)
              ...
DHCPv6
    Message type: Solicit (1)
    Transaction-ID: 0x00a4c428
    Client Identifier
              ...
    Identify Association
              ...
    Rapid Commit                              //rapid commit gesetzt: solicit und request in einem
        option type: 14
        option length: 0
    Option Request
        option type: 6
        option length: 20
        Requested Option code: DNS recursive name server (23)
        Requested Option code: Domain Search List (24)
              ...

No.     Time        Source                Destination           Protocol Info
      5 10.318875   fe80::20c:29ff:fe52:eee6 fe80::20c:29ff:fe6c:1679 DHCPv6   Reply

Internet Protocol Version 6
              ...
    Source address: fe80::20c:29ff:fe52:eee6       //Quelladresse: Server
    Destination address: fe80::20c:29ff:fe6c:1679   //Zieladresse:  Client
User Datagram Protocol, Src Port: 547 (547), Dst Port: 546 (546)
              ...
DHCPv6
    Message type: Reply (7)
    Transaction-ID: 0x00a4c428
    Server Identifier
              ...
    Rapid Commit                     //rapid commit option gesetzt: advertise und reply in einem
        option type: 14
        option length: 0
    Client Identifier                //Identifizierung des Clients durch DUID
              ...
    Identify Association
              ...
        IA Address
            IPv6 address: 3ffe:400:3d0:0:204:75ff:fe74:468c              //neue IPv6-Adresse für
Client
        Status code

              ...
    DNS recursive name server
              ...
    Domain Search List
```

6.3.4 Stateless DHCPv6

Beim stateless DHCPv6-Betrieb wird dem Client keine IPv6-Adresse über DHCPv6 zugewiesen. Der Client bekommt z.B. den DNS-Server und die Domain Suchliste mitgeteilt, je nach Wunsch des Clients. Dazu werden nur zwei DHCPv6-Nachrichten (INFORMATION-REQUEST,REPLY) benötigt. In der `client.conf` muss hierzu die Option `stateless` eingefügt werden, damit der Client keine IPv6-Adresse vom Server anfordert. Details zu Stateless DHCPv6 können [RFC3736] entnommen werden.

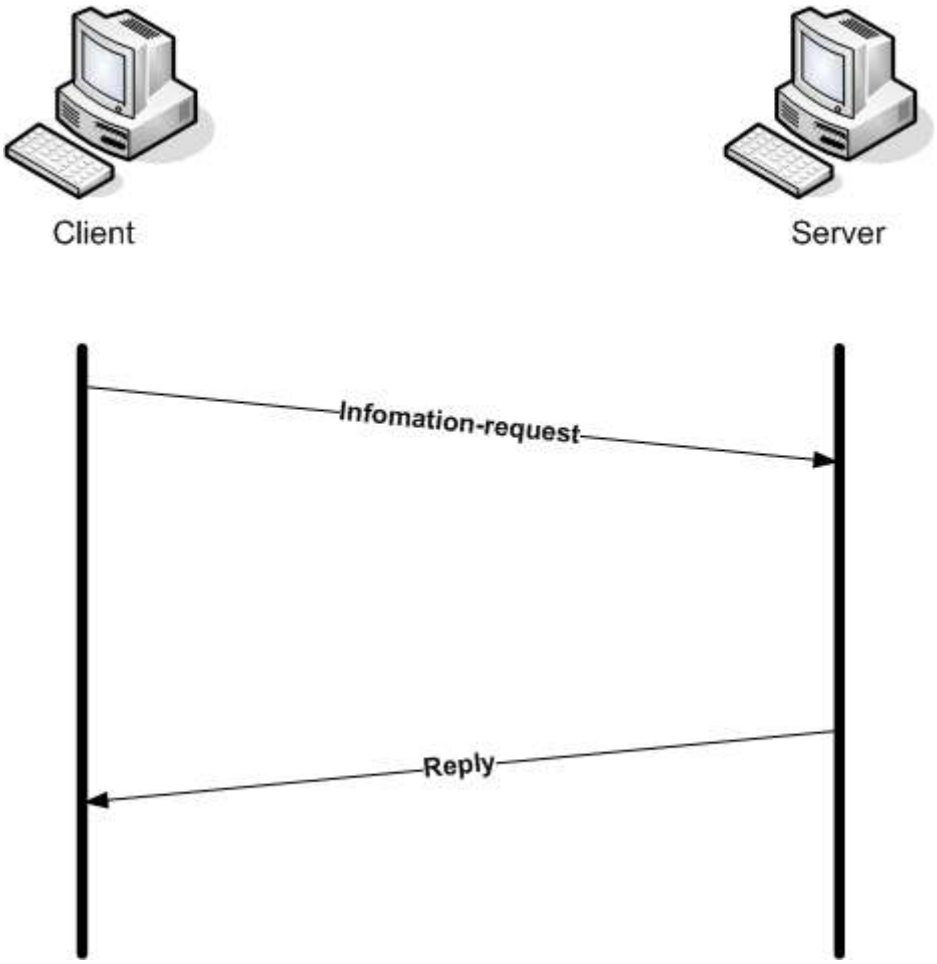

Abbildung 6.3.4: DHCPv6 Konfiguration mit stateless

Etherealmitschnitt mit stateless

```
// Client-Anfrage ohne Wunsch für IPv6-Adresse
No.     Time        Source                Destination           Protocol Info
      4 12.541503   fe80::20c:29ff:fe6c:1679 ff02::1:2            DHCPv6   Information-request
               ...
Internet Protocol Version 6
               ...
    Source address: fe80::20c:29ff:fe6c:1679
    Destination address: ff02::1:2
User Datagram Protocol, Src Port: 546 (546), Dst Port: 547 (547)
               ...
DHCPv6
    Message type: Information-request (11)
    Transaction-ID: 0x005f4a80
    Client Identifier
               ...
    Option Request

//Server-Antwort mit Konfigurationsdaten
No.     Time        Source                Destination           Protocol Info
      5 12.685041   fe80::20c:29ff:fe52:eee6 fe80::20c:29ff:fe6c:1679 DHCPv6   Reply
               ...
Internet Protocol Version 6
               ...
    Source address: fe80::20c:29ff:fe52:eee6       //Quelladresse: Server
    Destination address: fe80::20c:29ff:fe6c:1679 //Zieladresse: Client
User Datagram Protocol, Src Port: 547 (547), Dst Port: 546 (546)
               ...
DHCPv6
    Message type: Reply (7)
    Transaction-ID: 0x005f4a80
    Client Identifier                             //Identifizierung des Clients durch DUID
               ...
    DNS recursive name server
               ...
        DNS servers address: 3ffe:400:3d0::fffe     //IPv6-Adresse des DNS-Servers
    Domain Search List
               ...
    Server Identifier                             //Identifizierung des Server durch DUID

               ...
```

Man sieht der Client fordert keine IPv6-Adresse an und bekommt auch keine vom Server zugewiesen. Nur die Domain Suchliste und die Adresse des DNS-Servers werden übertragen.

6.3.5 Duplicate Address Detection (DAD)

Laut RFC sollte jeder Client die vom DHCP Server zugewiesene IPv6-Adresse überprüfen. Dibbler hat dies implementiert.
Hierzu wird eine Nachbarschafts-Anfrage (Neighbor Solicitation) gestellt, in welcher die eigene IP Adresse enthalten ist. Sollte die Adresse noch nicht vergeben sein, so wird auch keine Antwort zurückkommen und der DHCP Client behält die zugewiesene IP-Adresse. Sollte die Adresse jedoch vergeben sein, so wird der entsprechende Knoten eine Nachbarschafts-Bekanntmachung (Neighbor Advertisement) an alle Knoten versenden. Der DHCP Client weiss nun, dass seine Adresse bereits vergeben ist und sendet ein DECLINE an den DHCP Server und fordert (REQUEST) eine neue IP Adresse an.
Nachfolgend soll der Vorgang veranschaulicht werden, dazu wurde ein 2. Client mit der IPv6 Adresse 3ffe:400:3d0:0:204:75ff:fe74:468c konfiguriert.

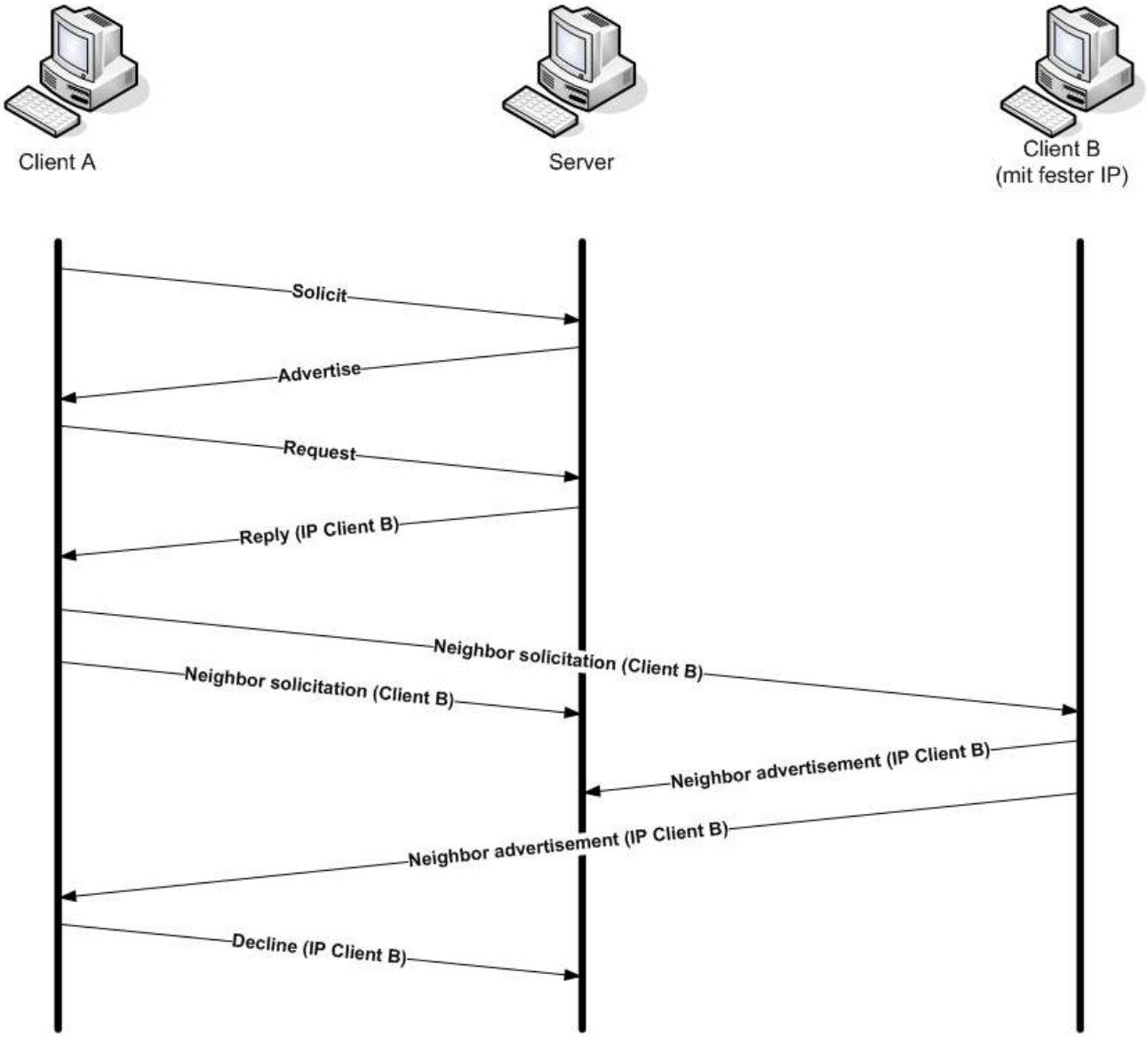

Abbildung 6.3.5: DHCPv6 Konfiguration mit DAD

```
No.     Time        Source                Destination           Protocol Info
     14 16.876544   fe80::20c:29ff:fe52:eee6 fe80::20c:29ff:fe6c:1679 DHCPv6   Reply


               ...
Internet Protocol Version 6
               ...
    Source address: fe80::20c:29ff:fe52:eee6        //Quelladresse: Server
    Destination address: fe80::20c:29ff:fe6c:1679   //Zieladresse: Client A
User Datagram Protocol, Src Port: 547 (547), Dst Port: 546 (546)
               ...
DHCPv6
    Message type: Reply (7)
               ...
    Identify Association
               ...
        IA Address
               ...
            IPv6 address: 3ffe:400:3d0:0:204:75ff:fe74:468c       //zugewiesene IPv6_Adresse
               ...


No.     Time        Source                Destination           Protocol Info
     16 17.129785   ::                    ff02::1:ff74:468c     ICMPv6   Neighbor solicitation

               ...
Internet Protocol Version 6
               ...
    Source address: ::                              //Quelladresse: Client A
    Destination address: ff02::1:ff74:468c          //Quelladresse: Client A (sich selbst)
Internet Control Message Protocol v6
    Type: 135 (Neighbor solicitation)
    Code: 0
    Checksum: 0x2ed5 (correct)
    Target: 3ffe:400:3d0:0:204:75ff:fe74:468c       //die zu überprüfenden IPv6-Adresse

No.     Time        Source                Destination           Protocol Info
     17 17.130953   3ffe:400:3d0:0:204:75ff:fe74:468c ff02::1   ICMPv6   Neighbor advertisement
               ...
Internet Protocol Version 6
               ...
    Source address: 3ffe:400:3d0:0:204:75ff:fe74:468c      //Quelladresse: Client B
    Destination address: ff02::1                           //Zieladresse: Alle Knoten
Internet Control Message Protocol v6
               ...
    Target: 3ffe:400:3d0:0:204:75ff:fe74:468c
    ICMPv6 options
       Type: 2 (Target link-layer address)
        Length: 8 bytes (1)
        Link-layer address: 00:04:75:74:46:8c
```

```
// Ablehnung der vorgeschlagenen IP Adresse
No.     Time        Source                Destination           Protocol Info
     22 23.419787   fe80::20c:29ff:fe6c:1679 ff02::1:2            DHCPv6   Decline
               ...
Internet Protocol Version 6
               ...
    Source address: fe80::20c:29ff:fe6c:1679              //Quelladresse: Client A
    Destination address: ff02::1:2                        //Zieladresse: Alle DHCPv6-Server
User Datagram Protocol, Src Port: 546 (546), Dst Port: 547 (547)
               ...
DHCPv6
    Message type: Decline (9)
               ...
    Server Identifier                       //Nachricht nur für ursprünglichen Server gedacht
               ...
    Identify Association
        option type: 3
        option length: 40
        IAID: 2
        T1: 1000
        T2: 2000
        IA Address
            option type: 5
            option length: 24
            IPv6 address: 3ffe:400:3d0:0:204:75ff:fe74:468c       //die zu verweigernde Adresse
            Preferred lifetime: 1800
            Valid lifetime: 3600

No.     Time        Source                Destination           Protocol Info   //ab hier
Wiederholung
     23 23.662870   fe80::20c:29ff:fe6c:1679 ff02::1:2            DHCPv6   Request
```

6.4 Nachweis der Netzwerkfähigkeit durch die DHCPv6 Konfiguration

In allen Fällen konnte der Client per DHCPv6 konfiguriert werden: Eine IPv6-DNS Anfrage sowie ICMPv6-Ping waren problemlos möglich

In der Datei `resolv.conf` fanden sich die vom DHCP-Server übermittelten Optionen

Inhalt von `/etc/resolv.conf`:

```
nameserver 3ffe:400:3d0::fffe
search fht-esslingen.de
search rznt.rzdir.fht-esslingen.de
```

Funktionsnachweis durch DNS Abfrage

```
client:/tmp # dig tolot.ipv6.uni-muenster.de AAAA
...
;; ANSWER SECTION:
tolot.ipv6.uni-muenster.de. 4955 IN     AAAA    2001:638:500:101:2e0:81ff:fe24:37c6
...
;; Query time: 1 msec
;; SERVER: 3ffe:400:3d0::fffe#53
```

Funktionsnachweis durch PING:

```
client:/tmp # ping6  -c 5 tolot.ipv6.uni-muenster.de
PING tolot.ipv6.uni-muenster.de(tolot.ipv6.uni-muenster.de) 56 data bytes
64 bytes from tolot.ipv6.uni-muenster.de: icmp_seq=1 ttl=61 time=19.2 ms
64 bytes from tolot.ipv6.uni-muenster.de: icmp_seq=2 ttl=61 time=19.2 ms
64 bytes from tolot.ipv6.uni-muenster.de: icmp_seq=3 ttl=61 time=19.2 ms
64 bytes from tolot.ipv6.uni-muenster.de: icmp_seq=4 ttl=61 time=19.1 ms
64 bytes from tolot.ipv6.uni-muenster.de: icmp_seq=5 ttl=61 time=19.4 ms

--- tolot.ipv6.uni-muenster.de ping statistics ---
5 packets transmitted, 5 received, 0% loss, time 4042ms
rtt min/avg/max/mdev = 19.170/19.255/19.405/0.171 ms
```

6.5 Zukunft mit Dibbler

Der Autor von der Dibbler-Implementierung möchte noch einige Features hinzufügen. Hier einige Beispiele:

- 'Relay Agent' Nutzung eines DHCPv6-Servers bei mehreren Subnetzen
- 'Crash Recovery' Nach Absturz des Clients, frägt er nach der selben IPv6-Adresse die ihm vor dem Crash zugeteilt worden war.
- 'DDNS Support' Hosteinträge dynamisch zum DNS-Server hinzufügen

7. Fazit

Nach ausgiebigen Tests mit der Dibbler-Implementierung können wir sagen, dass sie den jetzigen Anforderungen der FHTE genügt. Das Übertragen einer dynamischen sowie einer statischen IPv6 Adresse gelang auf Anhieb. Ebenso das Übertragen von stateless Daten wie etwa die IPv6 Adresse des DNS Servers und die Domain Suchliste.

Wenn die Implementierung mit den geplanten Features ergänzt wird, wäre sie an der FHTE einsetzbar. Insbesondere die 'Relay Agent' Unterstützung wäre wichtig für die FHTE, wenn das IPv6 Protokoll in weiteren Subnetzen verwendet werden soll.

8. Quellen

[WIESE] Wiese, Herbert: Das neue Internetprotokoll IPv6
ISBN 3-446-21685-5

[HAGEN] Hagen, Silvia: IPv6. Grundlagen - Funktionalität - Integration
ISBN: 3952294209

[STRAUF] Präsentation Christian Strauf zum Thema DHCPv6

http://www.join.uni-muenster.de/Dokumente/Folien/strauf/39.DFN-BT/DHCPv6/DHCPv6.pdf

[RFC3315] Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
ftp://ftp.rfc-editor.org/in-notes/rfc3315.txt

[RFC3736] Stateless DHCP for IPv6
ftp://ftp.rfc-editor.org/in-notes/rfc3736.txt

[ISC-DHCP] ISC DHCP (IPv4)
http://www.isc.org/sw/dhcp/

[DIPPLER] DHCPv6 Implementierung: Dibbler - a portable DHCPv6
http://klub.com.pl/dhcpv6/

[DHCPV6] DHCPv6 Implementierung : DHCPv6 @ Sourceforge
http://dhcpv6.sourceforge.net/

[HYCOMAT] DHCPv6 Implementierung: DHCPv6 @ Hycomat
http://www.hycomat.co.uk/dhcp/

[DUMPS] Ethereal Mitschnitte
http://cervicek.de/dhcpv6