Hochschule
Bonn-Rhein-Sieg
*University*
*of Applied Sciences*

Fachbereich Informatik
Department of Computer Sciences

# Projektbericht

im Studiengang Master Informatik

# Evaluation von Clientsystemen zur IPv6 Autokonfiguration

von

Christian Schneider
christian.schneider1@smail.inf.h-brs.de

| | |
|---|---|
| Betreuerin | Dipl. Inform. Martina Kannen |
| Erstprüfer | Prof. Dr. Martin Leischner |
| Zweitprüfer | Dipl. Inform. Wolfgang Pein |
| | |
| eingereicht am | 02.04.2012 |

## Eidesstattliche Erklärung

Ich versichere an Eides statt, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Niederkassel, den 2. April 2012

Christian Schneider

**Abstract**

Für die Autokonfiguration von Clients stehen in IPv6 zwei Verfahren zur Verfügung. Diese sind Stateless Address Autoconfiguration (SLAAC) und Stateful Address Autoconfiguration (DHCPv6). Mithilfe dieser Methoden kann ein Client mit einer globalen IPv6-Adresse, einer Default Route und der Adresse des DNS-Server automatisch versorgt werden.

In der Arbeit wird überprüft, ob die Clients mit allen nötigen Konfigurationsparametern in einer IPv6-Umgebung versorgt werden können. Dazu wurden unterschiedliche Testszenarien entwickelt, die die verschiedenen Möglichkeiten für eine Autokonfiguration abbilden. Es wird somit der Stand der Technik hinsichtlich der IPv6-Autokonfiguration der einzelnen Clients untersucht und dargestellt.

Es zeigte sich, dass nicht alle Clients den vollen Umfang an möglichen Autokonfigurationstechnologien unterstützen. Um eine möglichst große Menge an verschiedenen Clients automatisch konfigurieren zu können, eignet sich für den Netzverwalter die Kombination beider Verfahren. Einige Clients können dennoch nur in einer gemischten IPv4-/ IPv6-Umgebung sinnvoll betrieben werden, da ihnen in einer reinen IPv6-Umgebung die Möglichkeit fehlt, die DNS-Server Adresse zu ermitteln.

Zudem ergaben die Recherchen, dass auch weiterhin an Erweiterungen für die Autokonfiguration gearbeitet wird. Diese erweitern die Funktionalitäten der einzelnen Verfahren, wodurch diese in Zukunft möglicherweise unabhängiger voneinander und flexibler eingesetzt werden können.

# Inhaltsverzeichnis

# 1 Einleitung

Motivation

Auch wenn viele Systeme schon seit vielen Jahren IPv6-Unterstützung anbieten, ist dies nicht gleichbedeutend mit der Unterstützung von Verfahren zur automatischen IPv6 Adresskonfiguration. Für die Praxis sind diese Verfahren aber unerlässlich, um eine schnelle und einfache Autokonfiguration von Clients in einem Netz zu ermöglichen.

IPv6 sieht hierfür zwei Konzepte zur automatischen Adresskonfiguration vor:

- Stateless Address Autoconfiguration (SLAAC)
- Stateful Address Autoconfiguration (DHCPv6)

Es bieten jedoch nicht alle Clients diese Technologien vollständig an, wodurch nicht in jeder Umgebung immer eine Autokonfiguration des Clients gegeben sein muss. Es stellt sich die Frage, über welche dieser Technologien die Clients heutzutage verfügen, um eine Autokonfiguration durchzuführen bzw. in welchen Umgebungen sich diese Clients einsetzen lassen.

Ziel der Arbeit

Ziel dieser Projektarbeit ist es, einzelne Clients auf ihre Möglichkeit hinsichtlich IPv6-Autokonfiguration zu untersuchen und diesbezüglich den Stand der Technik darzustellen. Dazu wurden verschiedene Testszenarien entwickelt, die die verschiedenen Möglichkeiten für eine Autokonfiguration abbilden. Mithilfe dieser Szenarien wird geprüft, ob die Clients mit allen nötigen Konfigurationsparametern (Globale IPv6-Adresse, DNS-Server Adresse und Default-Route) in einer IPv6-Umgebung versorgt werden können.

# 2 Grundlagen

## 2.1 Autokonfiguration

Autokonfiguration sind Techniken zur automatischen Zuweisung von Konfigurationsparametern an Clients. Durch diese Verfahren ist das Anbinden eines Clients an ein Netz ohne manuelle Konfiguration möglich. Neben der Vermeidung von manuellen Konfigurationsfehlern, reduzieren diese Verfahren den Verwaltungsaufwand und ermöglichen das einfache Hinzufügen von weiteren Clients.

Wie schon bei IPv4 (BOOTP, DHCPv4) existieren auch für IPv6 Verfahren zur automatischen Schnittstellenkonfiguration eines Clients. DHCP ist in einer neuen Version (DHCPv6) auch für IPv6 vorhanden. Neben DHPC existiert noch ein weiteres Verfahren. Dieses wird als Stateless Address Autoconfiguration bezeichnet, da es keine Informationen speichert. DHCPv6 wird im Gegenzug als Stateful Autoconfiguration bezeichnet.

### 2.1.1 Parameter

Mithilfe einer automatischen Konfiguration können dem Client verschiedene Parameter automatisch zugewiesen werden. Dies können neben Schnittstellenparametern wie der IP-Adresse, MTU und der Default-Route auch die Adressen verschiedener Dienste wie DNS, SIP und NTP sein. Wichtig für die grundlegende Konfiguration eines Clients sind die folgenden drei Parameter:

- Globale IPv6-Adresse
- Default-Route
- DNS-Server Adresse

Bei der IPv6-Adresse des Clients ist zwischen einer link-lokalen und einer globalen IPv6-Adresse zu unterscheiden. Link-lokale Adressen werden vom Client ohne externe Informationen generiert und werden nicht von Routern weitergeleitet, wodurch nur eine Kommunikation in einem Netzsegment möglich ist. Daher ist eine globale Adresse und eine Default-Route wichtig, um mit Zielen außerhalb eines Netzes zu kommunizieren. Da diese Ziele oft mit einem Hostnamen statt einer IP-Adresse angegeben sind, ist es für die Namensauflösung dieser Hostnamen notwendig, dass der Client einen DNS-Server erreichen kann. Nur durch all diese drei Parameter ist gewährleistet (korrekte Funktionsweise des Netzes vorraugesetzt), dass der Client auch mit Zielen außerhalb seines lokalen Netzes kommunizieren kann.

Daher bezieht sich diese Arbeit auf diese drei Parameter und überprüft, ob diese mittels einer Autokonfiguration auf die einzelnen Clients gesetzt werden können.

### 2.1.2 Klassifizierung

Stateless Address Autoconfiguration

Eine Option zur automatischen Adresskonfiguration ist die Stateless Address Autoconfiguration (SLAAC). Diese verwendet keine zentralen Informationen und keinen Status, was der Grund für die Bezeichnung „Stateless“ ist. Hierfür kommuniziert der zu konfigurierende Host, mit dem im Netzsegment befindlichen Router, um die notwendigen Konfigurationsparameter zu erhalten. Dieses Verfahren gibt dem Netzverwalter zwar weniger Einfluss bei der Vergabe von IPv6-Adressen, befreit ihn aber gleichzeitig von der Konfiguration und der Verwaltung der zentralen Adressvergabe.

Stateless Address Autoconfiguration verwendet Router Advertisements, um die automatische Adresskonfiguration einer globalen IPv6-Adresse durchzuführen. Der genaue Aufbau einer Router Advertisment Nachricht wird in Kapitel 2.2 beschrieben. Stateless Address Autoconfiguration für IPv6 wird in dem RFC 4862 dargestellt (Thomson, 2007). Für eine Autokonfiguration werden folgende Schritte durchlaufen:

1. Generierung einer eindeutigen link-lokalen Adresse mit dem Präfix FE80 und dem angehängten Interface Identifier.
2. Zum Aufspüren von Routern und zur Ermittlung des Präfixes werden Router Solicitation Nachrichten an die Multicast Gruppe "All-Routers" (FF02::2) verschickt.
3. Alle Router, die diese Nachricht erhalten, antworten mit einer Router Advertisement Nachricht. Diese Nachrichten können einen Adresspräfix enthalten. Für jedes verteilte Präfix, wird eine globale IPv6-Adresse erstellt, die eine Kombination aus Präfix und Interface Identifier ist. Zum Beispiel:

   | | |
   |---|---|
   | Interface Identifier des Clients: | a667:06ff:fe3d:6082 |
   | Verteilter Adresspräfix: | 2001:0db8:0100:f101::0/64 |
   | Globale IPv6-Adresse: | 2001:0db8:0100:f101:a667:06ff:fe3d:6082 |

Da schnell erkannt wurde, dass diese Art der Adressgenerierung jederzeit zu einer eindeutigen Zuordnung von IPv6-Adresse und Hardware (Interface Identifier) führt, wurde eine Erweiterung Namens Privacy Extension eingeführt. Dies ist ein Verfahren, das die globale IPv6-Adresse nicht wie normalerweise mithilfe des Interface Identifiers generiert, sondern per Zufallszahl wechselnde IPv6-Adressen erzeugt. Dies schützt die Privatsphäre, da der Host nicht bei jedem Zugriff im Internet die gleiche IPv6-Adresse besitzt.

Stateful Address Autoconfiguration

Neben der Stateless Address Autoconfiguration existiert bei IPv6 auch noch die Stateful Address Autoconfiguration. Hierbei wird die Adresse, sowie weitere benötigte Informationen (DNS-Server Adresse etc.) durch einen zustandsbehafteten und zentralen Dienst (DHCPv6) vergeben. Diese kann verwendet werden, indem DHCPv6 auf dem Client aktiviert wird, oder indem in den Router Advertisements ein entsprechendes Feld gesetzt wird. Dies signalisiert dem Client, die Adresskonfiguration mittels eines DHCP-Servers durchzuführen.

Auch wenn es inzwischen mit der Stateless Autokonfiguration eine Alternative zu DHCP gibt, ist DHCP immer noch von Nöten, wenn z. B. ein spezielles IPv6-Adressschema verwendet wird, einzelnen Hosts separate Informationen zugeteilt werden, oder die Hosts weitere Konfigurationsinformationen wie DNS, NTP, etc. erhalten sollen. Diese Funktionen kann SLAAC nicht vollständig abdecken. Im Gegensatz zu DHCPv4 ist DHCPv6 aber nicht für die Verteilung des Default Gateways zuständig. Dies wird mithilfe der Router Advertisements durchgeführt.

Mischform / Parallelbetrieb

Zusätzlich ist eine Kombination beider Verfahren vorhanden. Dazu besteht die Möglichkeit, die IPv6-Adresse und die Default-Route mittels Stateless Address Autoconfiguration zu generieren und zudem noch zusätzliche Informationen wie z. B. DNS-Informationen vom DHCPv6 zu erfragen (Droms, 2004). Dieses wird als "Stateless DHCP" bezeichnet.

Außerdem lässt der Standard auch einen Parallelbetrieb zu, indem eine oder mehr Adressen per SLAAC generiert werden und parallel dazu Adressen vom DHCPv6-Server bezogen werden (Thomson, 2007 S. 20). Dadurch ist der Client über mehrere globale IPv6-Adressen erreichbar.

## 2.2 Router Advertisements

Zum besseren Verständnis der vorgestellten SLAAC werden hier die von diesem Verfahren verwendeten Router Advertisements genauer betrachtet. Der RFC 4861 beschreibt den Aufbau der Router Advertisements (Narten, 2007). Im Folgenden beschränken sich die Ausführungen auf die in dieser Arbeit verwendeten Bereiche.

Router Advertisements werden als ICMPv6-Nachricht (Nachrichtentype 134) von Routern in regelmäßigen Intervallen verteilt. Das Standardintervall beträgt 600 Sekunden, kann aber geändert werden auf ein Intervall zwischen 4 und 1.800 Sekunden. Der Aufbau einer Router Advertisement Nachricht ist in Abbildung 1 zu sehen. Das ein Bit große Feld M (Managed address configuration) signalisiert dem Client, ob eine Adresse von einem DHCPv6-Server bezogen werden soll. Das O Feld (Other configuration), falls gesetzt, gibt an, dass andere Konfigurationsinformationen per DHCPv6 verfügbar sind. Dies können z. B. DNS-relevante Informationen

oder aber auch Informationen von anderen Diensten sein. Ist das M-Feld gesetzt, ist das O-Feld redundant und kann ignoriert werden, da DHCPv6 alle vorhandenen Informationen zurückliefert. Falls M- und O-Feld (auch M- und O-Flag genannt) nicht gesetzt sind, weißt dies darauf hin, dass keine Informationen per DHCPv6 verfügbar sind.

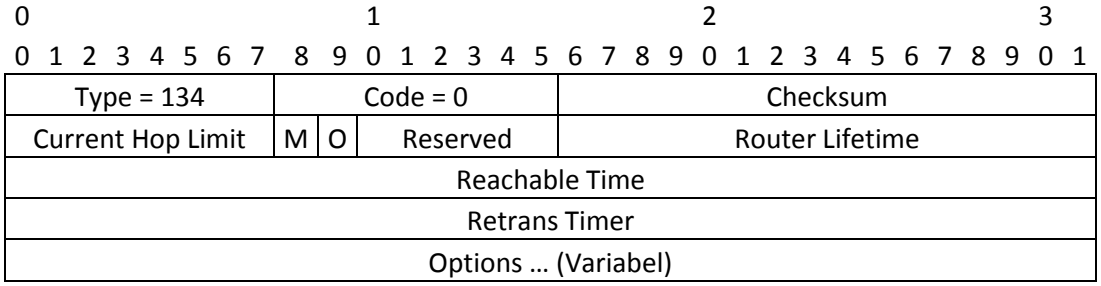


Abbildung 1: Router Advertisement Header

Das Feld Router Lifetime ist wichtig, falls der Router vom Client als Default-Router verwendet wird und gibt die Gültigkeitsdauer der Default-Route an. Ein Wert von 0 bedeutet, dass der Router kein Default-Router ist und auch nicht als Default-Route im Host eingetragen wird. Die Lifetime bezieht sich nur auf die Gültigkeit der Default-Route und hat keine Auswirkungen auf andere Nachrichtenfelder. Der Maximalwert ist auf der Senderseite auf 9000 Sekunden (2,5 Stunden) begrenzt. Ist die Gültigkeitsdauer verstrichen, ohne dass ein neues Paket von diesem Router empfangen wurde, muss der Host die Default-Route aus seiner Tabelle löschen.

Der Vollständigkeit halber sei hier erwähnt, dass Clients das Versenden von Router Advertisments außerhalb des regulären Intervalls mittels einer Router Solicitation Nachricht initialisieren können. Dazu wird die Router Solicitation Nachricht vom Client an alle Router versendet (Zieladresse ist die Router-Multicast-Adresse FF02::2).

### 2.2.1 Prefix Information

Als mögliche Option in einer Router Advertisement Nachricht können vom Router Präfixinformationen verteilt werden. Diese können für eine automatische Adresskonfiguration einer globalen IPv6-Adresse mittels Stateless Address Autoconfiguration benutzt werden.

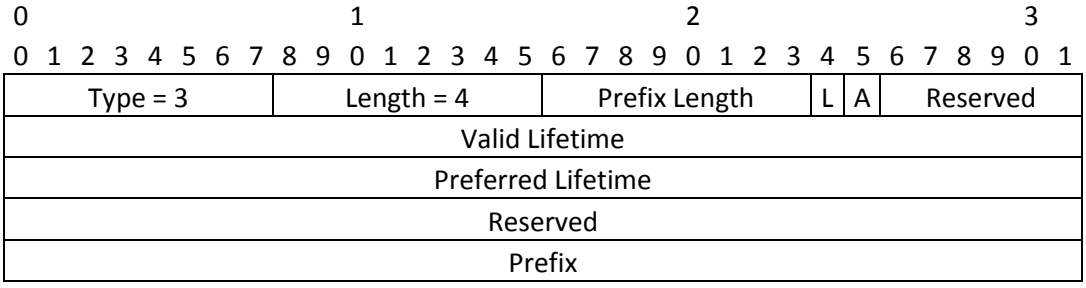


Abbildung 2: Prefix Information Header

Die Option enthält (siehe Abbildung 2) neben dem Adresspräfix noch zwei Werte für die Gültigkeitsdauer der Adresse. Valid Lifetime besagt, wie lange die Adresse gültig ist. Preferred Lifetime beschreibt, wie lange eine gültige Adresse bevorzugt verwendet wird. Das gesetzte

Flag "Autonomous Address-Configuration" (A-Flag) zeigt an, ob der Präfix überhaupt für die SLAAC genutzten werden darf.

### 2.2.2 DNS Configuration

Der RFC 6106 beschreibt eine Option (auch "RDNSS-Option" genannt) in einer Router Advertisement Nachricht, mit der DNS-Konfigurationsinformationen dem Client mitgeteilt werden können (Jeong, 2010). Dadurch ist der Client in der Lage DNS-Namensauflösungen durchzuführen. Ursprünglich war die Verteilung dieser Informationen nur per DHCP vorgesehen. Dies hatte zur Folge, dass neben einer Stateless Address Autoconfiguration immer noch ein DHCP-Server vorhanden sein musste, um dem Client die Informationen bzgl. des DNS-Servers mitzuteilen (abgesehen von einer manuellen Konfiguration). Durch den RFC 6106 ist es nun möglich, einen Client vollständig mittels Stateless Address Autoconfiguration zu konfigurieren und ihm folglich eine IPv6-Adresse, das Default Gateway und die Adresse des DNS-Server mitzuteilen.

Um die Informationen zu verteilen, erhalten die Router Advertisements eine RDNSS-Option (Type 25). In dieser befindet sich die Adresse des DNS-Servers und eine Zeit, die die Gültigkeitsdauer dieser Adresse angibt.

## 2.3 DHCPv6

Um einen DHCP-Server zu finden, sendet der Client eine Solicit-Nachricht an die Multicastadresse, die alle Server und Relais umfasst (ff02::1:2). DHCPv6-Client und -Server verwenden UDP für ihre Kommunikation. Der Client-Port hat die Nummer 546, Server und Relay-Port die Nummer 547. Der Client ist auch in der Lage, mit nur einem einzelnen DHCPv6-Server zu kommunizieren, indem er die ID (DUID) des Server in der Nachricht angibt (im Option Type "Server Identifier Option"). Alle Server erhalten zwar diese Nachricht, aber nur der mit der DUID angegebene Server wird auf diese Nachricht antworten (Hagen, 2006). Eine Übersicht der einzelnen Nachrichten und Options Typen sind ab Absatz 2.3.1 zu finden.

Erhält der Client innerhalb einer gewissen Zeitspanne keine Antwort auf seine Anfrage, wird eine weitere Solicit-Nachricht versendet oder der Konfigurationsvorgang beendet und eine Fehlermeldung generiert. Bekommt der Client eine Antwort mittels einer Advertise-Nachricht, sendet er eine Request-Nachricht an den DHCP-Server. Diese Nachricht enthält die IA-Option und die DUID des Clients sowie die Option "Option Request" mit den gewünschten DHCP-Optionen. Der Server antwortet mit einer Reply-Nachricht, die die angefragten Optionen enthält.

Eine typische DHCPv6-Kommunikation sieht also wie folgt aus:

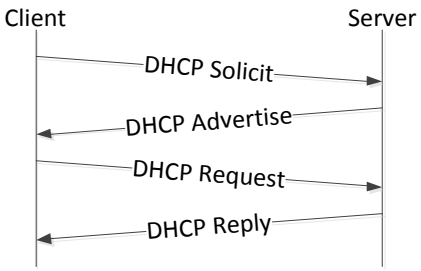


Abbildung 3: Ablauf einer DHCPv6-Kommuniktaion zur Adresskonfiguration

Falls der Client keine Adressinformation benötigt, sondern nur an anderen DHCP-Informationen, wie z. B. die IP-Adresse des DNS- oder NTP-Servers interessiert ist, sendet er eine Information-Request Nachricht anstelle der Request Nachricht. Diese enthält die Option "Option Request", mit der die gewünschten Informationen (Option) erfragt werden.

Kommunikationen mittels eines Agenten sind in dieser Arbeit nicht von Bedeutung. Daher sei hier nur erwähnt, dass nicht in jedem Subnetz ein DHCP-Server vorhanden sein muss, sondern die DHCP-Nachrichten auch mittels eines Agents an den DHCP-Server bzw. DHCP-Client weitergeleitet werden können. Weiter Information sind im RFC 3315 zu finden (Droms, 2003).

### 2.3.1 Message Types und Option Codes

Die DHCP-Nachrichten zwischen Client und Server besitzen eine feste Headergröße und einen variable Optionsanteil (siehe Abbildung 4). Für jede neue Anfrage generiert der Client eine neue Transaction ID. Diese dient zur Zuordnung der einzelnen Nachrichten (Solicit, Request, etc.) zu einer Transaktion.

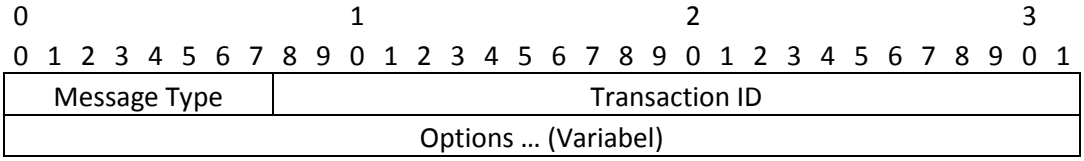


Abbildung 4: DHCPv6 Header

| Name | Message Type | Beschreibung |
|---|---|---|
| Solicit | 1 | Wird vom Client zur Lokalisierung des Servers verwendet. |
| Advertise | 2 | Wird vom Server als Antwort auf eine Solicit-Nachricht verschickt. |
| Request | 3 | Wird vom Client benutzt, um Informationen vom Server zu erhalten. |
| Reply | 7 | Wird vom Server als Antwort auf die Request, Release und Information Request-Nachricht verschickt. |
| Release | 8 | Wird vom Client benutzt, um seine Adresse wieder freizugeben. |
| Information Request | 11 | Wird vom Client benutzt, um zusätzliche Konfigurationsparameter zu erfragen. |

Tabelle 1: Auswahl von der in der Arbeit betrachteten DHCPv6 Message Types

Die möglichen Nachrichtenarten (Message Type) einer DHCPv6-Kommunikation sind in Tabelle 1 zu finden. Hierbei handelt es sich aber nur um einen Auszug der in der Arbeit verwendeten Nachrichten. Eine Nachricht kann mehrere Optionen enthalten, die jeweils entsprechende Konfigurationsinformationen enthalten.

Der Aufbau einer Option entspricht dem Format in Abbildung 5. Eine Übersicht der vorhandenen Optionen und deren Beschreibung findet sich in Tabelle 2. Es existieren inzwischen über 60 Options, wodurch auch hier wieder nur eine Auswahl der in dieser Arbeit verwendeten Option gezeigt wird.

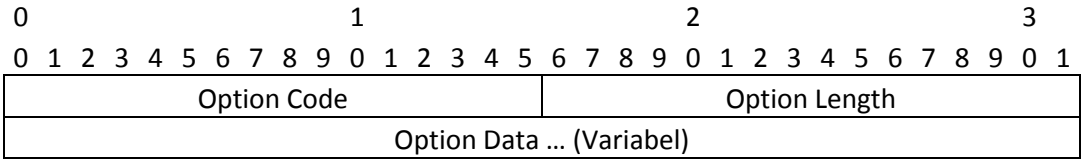


Abbildung 5: DHCPv6 Option Header

| Name | Option Code | Beschreibung |
|---|---|---|
| Client Identifier | 1 | Zur Übertragung einer eindeutigen Client ID (Client DUID) |
| Server Identifier | 2 | Zur Übertragung einer eindeutigen Server ID (Server DUID) |
| Identity Association for Nontemporary Addresses (IA_NA) | 3 | Wird verwendet, um die IA_NA, die Parameter und die dazugehörige nicht temporäre Adresse anzuzeigen. |
| IA Address | 5 | Wird verwendet, um eine Adresse mit einer IA_NA zu verknüpfen. |
| Option Request | 6 | Beinhaltet eine Liste von angeforderten Optionen. |
| Elapsed Time | 8 | Beinhaltet die Zeit, zu der der Client die DHCPv6 Transaktion gestartet hat. |
| Vendor Class | 16 | Wird vom Client benutzt um herstellerspezifische Angaben über sich bekannt zu machen. (z. B. OS) |
| DNS Recursive Name Server | 23 | Beinhaltet die Adresse des DNS-Servers. |
| Fully Qualified Domain Name | 39 | Wird vom Client zum Austausch des Fully Qualified Domain Names benutzt. |

Tabelle 2: Auswahl der in der Arbeit verwendeten DHCPv6 Options

### 2.3.2 DHCP Unique Identifier

Jeder DHCPv6-Client oder Server besitzt eine DHCP Unique Identifier (DUID) zur gegenseitigen Identifizierung. Der Server benutzt die DUID des Clients zur Auswahl der entsprechenden Konfigurationsparameter, die übertragen werden. Die DUID muss eindeutig sein und sollte nach der initialen Generierung nicht mehr geändert werden (Hagen, 2006). Bei DHCPv4 wurde zur Identifikation nur die Hardware Adresse genutzt. In IPv6 existieren zurzeit drei Verfahren zur Erstellung der DUID:

- Link-Layer Adresse plus Zeit (DUID-LLT)
- Lieferantenspezifische eindeutige ID, basierend auf der Enterprise Nummer (DUID-EN)
- Link-Layer Adresse (DUID-LL)

Das im Standard empfohlene Verfahren soll zur Generierung der DUID die Link-Layer Adresse plus Zeit verwenden. Da die DUID eine variable Länge haben kann und auch nicht in allen DHCP-Nachrichten erforderlich ist, wird die DUID als Option Feld übertragen. Eine DUID enthält einen zwei Byte großen Type Code (1 für Client DUID, 2 für Server DUID) gefolgt von einer variablen Anzahl an Bytes, die die ID enthalten (Droms, 2003).

### 2.3.3 Identity Association

Die Identity Association (IA) wird von den DHCP-Clients und Servern verwendet, um Gruppen von IPv6-Adressen zu identifizieren und zu verwalten. Jede IA enthält eine IAID und zugehörige Konfigurationsinformationen. Ein Client besitzt mindestens eine IA pro Schnittstelle, welche mittels DHCP konfiguriert wurde. Jede IA darf nur einer Schnittstelle zugeordnet werden. Der Client benutzt eine IA, um die entsprechenden Informationen für die Schnittstelle vom DHCP-Server zu erfragen. Der Client ist für die Generierung der IAID zuständig und muss dafür sorgen, dass diese in seinem Kontext eindeutig ist. Auch die IAID sollte nach der ersten Generierung gespeichert und nicht mehr verändert werden. Die Konfigurationsinformationen in einer IA enthalten eine oder mehr IPv6-Adressen sowie Timer, die angeben, wann eine Adresse erneuert werden muss.

Ein DHCP-Server wählt die Konfigurationsinformationen für eine IA entsprechend der Adressrichtlinien des Administrators und abhängig von der Schnittstelle, an die der Client angebunden ist, der DUID des Clients und anderen Informationen, die vom Client mittels Options-Feld übergeben wurden.

# 3 Tests

## 3.1 Testumgebung

In diesem Kapitel wird die Testumgebung vorgestellt. Dieses beinhaltet die zu testenden Clients, sowie die Serversysteme und das Netz, die zum Testen verwendet wurden. Zudem werden die Szenarien vorgestellt, nach denen die Clients auf ihre Funktionalitäten hin geprüft werden.

### 3.1.1 Clientsysteme

Um die IPv6-Autokonfiuration zu testen, wurden verschiedene Clients ausgewählt. In Tabelle 3 ist eine Übersicht der einzelnen Betriebssysteme samt Version zu sehen. Durch die Auswahl ist gewährleistet, dass eine breite Palette der gängigsten Betriebssysteme untersucht wird. So zählt Windows XP trotz seines Alters von knapp 11 Jahren immer noch zu den häufigsten Betriebssystemen (Refsnes Data, 2012). Als neuste Version von Windows wurde Windows 7 verwendet. Zudem wurde neben Windows auch Mac OS X und Linux (Ubuntu) betrachtet, da diese zu den drei am häufigsten verwendeten Betriebssystemen zählen. Auch immer mehr mobile Endgeräte (Smartphone & Tablets) werden verkauft, welche über WLAN und IPv6 konnektierbar sind, aus welchem Grund diese auch in die Untersuchung aufgenommen wurden. Untersucht wurden iOS und Android. Für diese und für Mac OS X ergaben sich Einschränkungen/Vorgaben, da diese Geräte von der Hochschule zur Verfügung gestellt wurden. Daher konnten keine anderen Versionen als die vorhandenen untersucht werden. Ein Blackberry Handy stand nicht zur Verfügung.

| Clients | |
|---|---|
| **Name** | **Version** |
| Linux Ubuntu | 10.04 & 11.10 |
| Windows XP | SP3 |
| Windows XP & Dibbler | SP3 & 0.8.1 |
| Windows 7 | SP1 |
| Mac OS X | 10.5.6 |
| iPAD mit iOS | 5.0.1 |
| Handy mit Android | 4.03 |

| Server | |
|---|---|
| **Name** | **Version** |
| Windows 2008 R2 | SP1 |
| Ubuntu mit<br>Dibbler Server &<br>Radvd | 11.10<br>0.8.1<br>1.8.3 |

Tabelle 3: Übersicht der untersuchten Systeme

Durch die Auswahl der unterschiedlichen Systeme wurden auch gleichzeitig verschiedene DHCPv6-Implementierungen getestet. So verwendet Ubuntu die Implementierung der Internet Systems Consortium. Dies ist eine Referenzimplementierung des Standards (Internet Systems Consortium, 2012). Mit Dibbler wurde eine weitere Implementierung eines DHCPv6-Clients

getestet. Diese Software ist kostenlos und auf Windows und Linux-Systemen lauffähig (Mrugalsk, 2011). Windows 7 verfügt über eine eigene (von Microsoft erstellte) Implementierung eines DHCPv6-Clients.

### 3.1.2 Serversysteme

Neben den Clients existieren auch eine Vielzahl von DHCPv6-Server Implementierungen. Für die Serversysteme in der Testumgebung wurden exemplarisch die DHCPv6-Server Software Dibbler (Version 0.8.1) und der Daemon radvd (Version 1.8.3) zum Versenden von Router Advertisements verwendet, sowie als alternative Wahl Windows 2008 R2 mit dem von Haus aus vorhanden Windows DHCPv6-Server.

Nachfolgend wird eine Konfiguration für die beiden Serversysteme zur Verteilung von Router Advertisements und DHCPv6-Informationen erstellt und erläutert. Anschließend findet ein Vergleich der Funktionalitäten statt.

#### *3.1.2.1 Windows 2008 R2 mit DHCP-Rolle*

Damit der Server als DHCPv6-Server arbeiten kann, muss die DHCP-Rolle hinzugefügt und konfiguriert werden. Der DHCP-Server ist zuständig für die Verteilung der Adressen im Bereich 2001:638:408:201::0/64. Als DHCPv6 Option wird noch "DNS Recursive Name Server" mit der IPv6-Adresse 2001:638:408:201::150 bereitgestellt.

Damit der Server auch Router Advertisements verteilt, müssen weitere Einstellungen über die Commandline vorgenommen werden. Mit den drei folgenden Befehlen kann das Versenden von Router Advertisements aktiviert und das M- und O-Feld in den Nachrichten gesetzt werden.

```
Netsh interface ipv6 set interface <id> advertise=enabled
Netsh interface ipv6 set interface <id> otherstateful=enabled
Netsh interface ipv6 set interface <id> managed=enabled

Netsh interface ipv6 set interface <id> forwarding=enabled
```

Es ist unbedingt auch darauf zu achten, dass das IP-Forwarding auf dem Server aktiviert ist, damit die Router Lifetime in den Router Advertisement Nachrichten vom Server ungleich 0 gesetzt wird. Ist dies nicht der Fall, setzt der Host, der die Router Advertisements empfängt, die Default-Route nicht.

Sollen zusätzlich auch noch Adresspräfixe per Router Advertisements verteilt werden, muss eine Route dafür in Windows angelegt werden und diese publiziert werden. Mit folgendem Befehl ist dies möglich:

```
Netsh interface ipv6 set route 2001:638:408:201::0/64 <id> publish=yes
```

### 3.1.2.2 Ubuntu mit Dibbler DHCPv6-Server und Radvd Daemon

Der Dibbler DHCPv6-Server und der Router Advertisement Daemon Radvd werden über Konfigurationsdateien gesteuert. Die Konfigurationsdatei von Radvd befindet sich unter /etc/radvd.conf. Unter /etc/dibbler/server.conf ist die Konfigurationsdatei des Dibbler DHCPv6-Servers zu finden.

```
iface <name> {
 prefered-lifetime 3600
 valid-lifetime 7200
 class {
   pool 2001:638:408:201::0/64
 }
 option dns-server 2001:638:408:201::150
}
```

Abbildung 7: Beispielkonfiguration des Dibbler DHCPv6-Servers

```
interface <name> {
 AdvSendAdvert on;
 AdvManagedFlag on;
 AdvOtherConfigFlag on;
 prefix 2001:638:408:201::0/64
 {};
};
```

Abbildung 6: Beispielkonfiguration des Radvd Daemons

Diese Konfigurationen entsprechen denen im Windows DHCPv6-Server und Router Advertisement Dienst vorgenommenen Einstellungen. Der DHCPv6-Server verteilt durch die Konfiguration nun Adressen aus dem Bereich 2001:638:408:201::0/64 und die IP-Adresse des DNS-Server 2001:638:408:201::150 mittels der DNS-Server Option. Der Radvd Daemon versendet Router Advertisements mit gesetztem M- und O-Feld und verteilt das Präfix 2001:638:408:201::0/64. Radvd warnt im Gegensatz zu Windows, dass das IP-Forwarding aktiviert werden muss. Dies geschieht temporär mittels folgendem Befehl:

```
sudo sysctrl -w net.ipv6.conf.default.forwarding=1
```

### 3.1.2.3 Vergleich

Die Linux Variante mit Dibbler und Radvd bietet im Allgemeinen mehr Möglichkeiten hinsichtlich Funktionsumfang und Konfiguration, als Windows. So können mittels Dibbler mehr DHCPv6 Optionen verteilt werden. Windows unterstützt zur Zeit nur zehn Optionen, obwohl die Internet Assigned Numbers Authority (IANA) schon mehr als 60 DHCPv6 Optionen vergeben hat. Die Router Advertisements können in Windows auch nicht vollständig konfiguriert werden. Radvd bietet feinere Konfigurationsmöglichkeiten und schon seit Version 1.0 die Funktion, RDNSS-Optionen zu verteilen (Litech Systems Design, 2012). Bei Windows existiert dies Möglichkeit nicht. Dadurch konnte die Unterstützung der RDNSS-Option bei den Clientsystemen nicht mit dem Windows-Server getestet werden. Der Radvd Dämon verteilt mit nachfolgender Konfiguration auch noch die RDNSS-Option.

```
interface <name> {
  AdvSendAdvert on;
  AdvManagedFlag on;
  AdvOtherConfigFlag on;
  prefix 2001:638:408:201::0/64 {
  };
  RDNSS 2001:638:408:201::150 {
  };
};
```

Abbildung 8: Radvd Konfiguration mit RDNSS-Option

Allgemein sind neue Standards schneller in Dibbler und Radvd implementiert, als in Windows. Aber auch nicht standardkonforme Funktionen sind vorhanden. Dabei handelt es sich meist um Erweiterungen wie z. B. eigene Optionen und Internet-Drafts (siehe Kapitel 5), die vom Client ignoriert werden sollten, falls dieser die Erweiterungen nicht unterstützt. Sie erschweren jedoch möglicherweise den Ausschluss von Fehlern.

Sollen also mehr als die Standardkonfigurationsparameter wie IPv6-Adresse, DNS-Server und Default-Route verteilt und insbesondere neuere Standards berücksichtigt werden, ist der Einsatz von Dibber und Radvd dem von Windows vorzuziehen.

### 3.1.3 Testszenarien

Zur Überprüfung der Funktionalität der einzelnen Clients, hinsichtlich der unterschiedlichen Autokonfigurationsverfahren, werden vier verschiedene Szenarien gebildet. Die Szenarien decken die sinnvolle Verwendung der einzelnen Verfahren bzw. deren Kombination zur automatischen Konfiguration ab.

Jedes Szenario beschreibt die Art und Weise, wie die Informationen bzgl. der Autokonfiguration im Netz verteilt bzw. vorhanden sind, wobei jedes davon eine vollständige Konfiguration des Clients ermöglicht. Es werden somit auf unterschiedliche Weise die drei notwendigen Parameter (IPv6-Adresse, Default-Route und DNS-Server) verteilt. Dadurch hängt der (Miss-) Erfolg einer Autokonfiguration von den unterstützten Funktionen des Clients ab. Jeder zu testende Client durchläuft alle vier Szenarien, wobei nach jedem Szenario das Ergebnis der Autokonfiguration überprüft wird.

1. **Szenario (SLAAC)**
   - **Verteilte / Vorhandene Informationen im Netz:**
     Nur Router Advertisements mit Präfixinformationen und RDNSS-Option. Die Adresse des DNS-Servers wird somit mithilfe der RDNSS-Option konfiguriert. Ein DHCP-Server ist im Netz nicht vorhanden.
   - **Überprüfte Funktionalität:**
     In diesem Szenario wird überprüft, ob der Client mittels SLAAC konfiguriert werden

kann. Zudem wird geprüft, ob die Generierung der IPv6-Adresse mittels der Privacy Extension erfolgen kann.

- **Einsatzmöglichkeiten:**

  Dieses Szenario eignet sich für Umgebungen, in denen Administration und Protokollierung nicht gebraucht oder nicht gewünscht sind. Da keine zentrale Verwaltung und Speicherung der Adressvergabe vorhanden ist und durch die mögliche Privacy Extension diese auch nicht aus der Hardware Adresse ableitbar ist, schafft dieses Szenario eine gewisse Vertraulichkeit und Privatssphäre.

2. **Szenario (Stateless DHCP)**

   - **Verteilte / Vorhandene Informationen im Netz:**

     Router Advertisements mit Präfixinformationen sowie gesetztem O-Flag und ein DHCPv6-Server

   - **Überprüfte Funktionalität:**

     In diesem Szenario wird überprüft, ob der Client Stateless DHCP unterstützt.

   - **Einsatzmöglichkeiten:**

     Dieses Szenario ist ähnlich wie das erste Szenario, eignet sich aber besonders für den Fall, wenn mehr Informationen verteilt werden sollen. So ist es möglich mithilfe des DHCPv6-Servers Informationen für weitere Dienste wie SIP oder NTP, sowie auch herstellerspezifische Informationen zu verteilen. Zudem kann dieses Szenario verwendet werden, wenn die Clients nicht die RDNSS-Option unterstützen.

3. **Szenario (DHCPv6)**

   - **Verteilte / Vorhandene Informationen im Netz:**

     DHCPv6-Server und Router Advertisements (ohne Präfixinformationen). Einmal mit und einmal ohne gesetztes M-Flag.

   - **Überprüfte Funktionalität:**

     In diesem Szenario wird überprüft, ob der Client Konfigurationsinformationen von einem DHCPv6-Server bezieht. Dies wird einmal mittels gesetztem M-Flag überprüft. In diesem Fall signalisiert das M-Flag dem Client die Konfigurationen vom DHCPv6-Server zu beziehen. Im zweiten Fall, ohne gesetztes M-Flag, wird durch entsprechende Konfiguration des Clients versucht, diesen auf DHCPv6 einzustellen.

   - **Einsatzmöglichkeiten:**

     Dieses Szenario ist ähnlich wie bei IPv4. Für die Verteilung der Konfigurationsinformationen ist eine zentrale Instanz (DHCP-Server) zuständig. Diese kann vollständig administriert werden. Zudem stehen Protokollierungsmöglichkeiten bereit.

4. **Szenario (Dualbetrieb)**

   - **Verteilte / Vorhandene Informationen im Netz:**

     Router Advertisements mit Präfixinformationen sowie gesetztem M-Flag und ein DHCPv6-Server

   - **Überprüfte Funktionalität:**

     In diesem Szenario wird überprüft, ob der Client beide Verfahren zur Autokonfiguration gleichzeitig verwenden kann.

   - **Einsatzmöglichkeiten:**

     Dieses Szenario könnte verwendet werden, um eine möglichst hohe Komptabilität zu vielen Clients zu gewährleisten, indem alle Informationen über beide Verfahren gleichzeitig verteilt werden. Falls ein Client SLAAC und DHCPv6 unterstützt, würde er in diesem Szenario sogar zwei IPv6-Adressen erhalten.

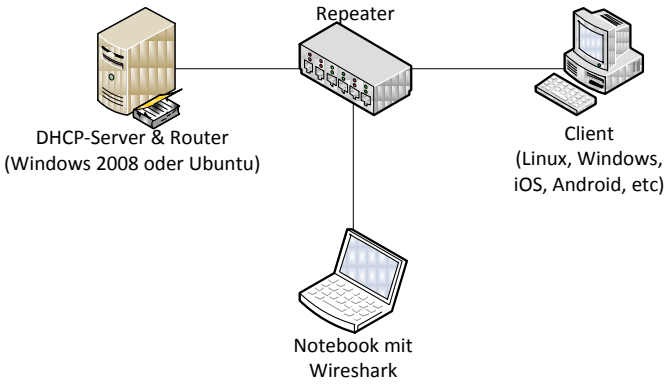


Abbildung 9: Aufbau der Testumgebung

Für die Durchführung der einzelnen Testszenarien wurde eine kleine Umgebung erstellt (siehe Abbildung 9). Hierbei sind die Clients über einen Repeater direkt an den Server/Router angeschlossen. Mit dem Repeater verbunden ist zudem noch ein Notebook, mit dessen Hilfe und Wireshark die Kommunikation mitgeschnitten werden kann. Der Server übernimmt neben der DHCP-Funktion noch die Rolle eines Routers, um Router Advertisements zu versenden.

Durch dieses einfache Szenario, konnten auch in der Einleitung auf weiterführende Erläuterungen (Relay-Agent/Nachrichten etc.) verzichtet werden, da diese für das Szenario nicht relevant sind.

## 3.2 Testergebnisse

### 3.2.1 Windows XP

Windows XP unterstützt IPv6 ab Service Pack 2, muss aber für den Gebrauch erst installiert werden (Befehl: `netsh interface ipv6 install`). Danach ist entweder eine statische IPv6 oder eine automatische IPv6 Konfiguration mittels Router Advertisements möglich. Ein DHCPv6-Client existiert nicht.

**Szenario 1 (SLAAC):**

Nach der Installation von IPv6 ist eine Konfiguration mittels SLAAC möglich. Nur die zusätzliche RDNSS-Option in den Router Advertisements wird von Windows XP nicht unterstützt (Palmer, 2012). Dadurch kann in Szenario 1 nur die Default-Route und die IPv6-Adresse konfiguriert werden. Die Generierung der IPv6-Adresse wird nach dem RFC 4941 "Privacy Extension" durchgeführt.

**Szenario 2 (Stateless DHCP):**

Mithilfe der Router Advertisements und der darin enthaltenen Präfixinformationen kann auf dem Client eine IPv6-Adresse und die Default-Route konfiguriert werden. Das O-Flag signalisiert dem Client weitere Informationen mittels eines DHCPv6 Information-Request anzufordern. Dies ist durch den fehlenden DHCPv6-Client nicht möglich, wodurch der DNS-Server Eintrag nicht konfiguriert wird.

**Szenario 3 (DHCPv6):**

Da Windows XP keinen DHCPv6-Client besitzt, können in Szenario 3 keine Informationen von dem DHCPv6-Server verarbeitet werden. Dies ist sowohl mit als auch ohne gesetztem M-Flag der Fall. Es wird somit keine IPv6-Adresse und DNS-Server konfiguriert. Nur die Default-Route könnte mithilfe der Router Advertisements gesetzt werden.

**Szenario 4 (Dualbetrieb):**

Der Windows XP Client kann sich nur eine Konfiguration mittels SLAAC automatisch erstellen (ohne DNS-Server). Die gleichzeitige Verwendung von DHCPv6 durch das gesetzte M-Flag ist durch den fehlenden DHCPv6-Client nicht möglich.

**Ergebnis:**

Auf Grund der Tatsache, dass die RDNSS-Option und der DHCPv6-Client nicht vorhanden sind, fehlt Windows XP eine automatische Funktion zum Erhalt der DNS-Server Informationen über IPv6. Nur in einer gemischten Umgebung von IPv4 und IPv6 hat der Client eine Möglichkeit, die Informationen automatisch von einem DHCPv4-Server zu erhalten. Ohne diese Möglichkeit bleibt nur noch eine manuelle Konfiguration des DNS-Servers.

Somit ist Windows XP ohne zusätzlich installierten DHCPv6-Client wie z. B. Dibbler nicht adäquat in einer reinen IPv6 Umgebung einsetzbar, da auf jedem Client einzeln ein DNS-Server zur Namensauflösung eingetragen werden muss.

### 3.2.2 Windows XP mit Dibbler DHCPv6-Client

Nach der Installation von Dibbler verfügt nun auch der Windows XP Host über einen DHCPv6-Client.

**Szenario 1 (SLAAC):**

Die Installation von Dibbler DHCPv6-Client ändert nichts am ersten Szenario. Die RDNSS-Option wird weiterhin nicht unterstützt. Im Szenario 1 kann nur die Default-Route und die IPv6-Adresse konfiguriert werden. Die IPv6-Adresse wird weiterhin mit der Hilfe der "Privacy Extension" durchgeführt.

**Szenario 2 (Stateless DHCP):**

Mithilfe der Router Advertisements und der darin enthaltenen Präfixinformationen kann auf dem Client eine IPv6-Adresse und die Default-Route konfiguriert werden. Das O-Flag in den Router Advertisements wird von dem Dibbler-Client nicht verarbeitet, wodurch keine DHCPv6 Information-Requests verschickt werden. Es kann dennoch eine Autokonfiguration im Sinne von Stateless DHCP durchgeführt werden, indem die Konfiguration des Dibbler-Clients angepasst wird.

```
stateless
iface eth1
{
  option dns-server
}
```

Hierdurch wird die IPv6-Adresse und die Default-Route mittels SLAAC konfiguriert und unabhängig von den gesetzten Flags in den Router Advertisements wird ein Information-Request an den DHCPv6-Server gesendet um die DNS-Server Adresse zu ermitteln.

**Szenario 3 (DHCPv6):**

Da die Flags in den Router Advertisements von Dibbler nicht verarbeitet werden, ist nur durch eine entsprechende Clientkonfiguration eine Autokonfiguration über DHCPv6 möglich.

```
iface eth1 {
  ia
  option dns-server
}
```

Durch die Konfiguration des Clients verschickt dieser automatisch (unabhängig vom gesetztem M-Flag) an den DHCPv6-Server eine Solicit- und Request-Nachricht und erhält von diesem eine IPv6-Adresse und den DNS-Server zugewiesen. Durch die zusätzlich im Netz verteilten Router

Advertisements kann zudem die Default-Route auf dem Client konfiguriert werden. Somit ist eine vollständige automatische Konfiguration des Clients möglich.

**Szenario 4 (Dualbetrieb):**

In diesem Szenario hängt die erzeugte Konfiguration wieder von den vorgenommenen Clienteinstellungen ab. Werden für den Dibbler-Client zum Beispiel die Einstellungen aus Szenario 3 übernommen, bezieht der Client eine IPv6-Adresse und den DNS-Server Eintrag vom DHCPv6-Server und die Default-Route sowie eine weitere IPv6-Adresse mittels SLAAC.

**Ergebnis:**

Da der Dibbler Client die Flags in den Router Advertisements nicht verarbeitet, ist für jedes Szenario eine Anpassung der Clientkonfiguration notwendig. Danach ist aber eine vollständige Autokonfiguration ist mithilfe von Dibbler in den Szenarien 2, 3 und 4 möglich.

### 3.2.3 Windows 7

Windows 7 bietet auch von Haus aus IPv6-Funktionalitäten an. Im Gegensatz zu Windows XP ist aber IPv6 standardmäßig aktiviert und auf eine automatische Adresskonfiguration eingestellt. Erreichen den Windows-Host auch nach mehrmaligen verschicken von Router Solicitations keine Antworten, versucht der Host über DHCPv6-Anfragen einen DHCP-Server zu finden. Daraus ist auch ersichtlich, dass Windows 7 über einen DHCP-Client verfügt. Dieser kann nur nicht explizit über die GUI konfiguriert werden. Die einzige Auswahl, die der Benutzer über die GUI hat ist "Statische Adressvergabe" zur manuellen Vergabe der IPv6-Adresse und "Automatische Adressvergabe", die den oben genannten Ablauf vollführt. Der DHCP-Client kann nur minimal per Command-Line Programm *netsh* konfiguriert werden.

**Szenario 1 (SLAAC):**

IPv6 ist standardmäßig unter Windows 7 aktiviert und auf eine automatische Adresskonfiguration eingestellt. Dies bedeutet bei Windows, dass eine Router Solicitation verschickt bzw. auf Router Advertisements reagiert wird. Die Router Advertisemets, die im Netz durch den Router verteilt werden, können somit verarbeitet werden. Nur wie bei Windows XP unterstützt Windows 7 auch nicht die RDNSS-Option (Palmer, 2012), wodurch nur die IPv6-Adresse und die Default-Route in Szenario 1 automatisch konfiguriert werden. Die Generierung der IPv6-Adresse erfolgt wieder per Privacy Extension.

**Szenario 2 (Stateless DHCP):**

Mithilfe der Router Advertisements und der darin enthaltenen Präfixinformationen kann auf dem Client eine IPv6-Adresse und die Default-Route konfiguriert werden. Das O-Flag signalisiert dem Client weitere Informationen mittels eines DHCPv6 Information-Request anzufor-

dern. Dadurch wird mithilfe des Windows 7 DHCPv6-Clients der DNS-Server Eintrag vom DHCPv6-Server erfragt und konfiguriert. Dieses Szenario ermöglicht eine vollständige automatische Konfiguration des Clients.

**Szenario 3 (DHCPv6):**

Das gesetzte M-Flag in den Router Advertisements signalisiert dem Client, dass die Adressinformationen mittels DHCP konfiguriert werden sollen. Der Client schickt daher einen DHCPv6-Request an den DHCPv6-Server um eine IPv6-Adresse und weitere Informationsparameter zu erhalten. Die IPv6-Adresse sowie der DNS-Server werden somit vom DHCP-Server bezogen. Die Default-Route wird mittels der Router Advertisements gesetzt.

Werden Router Advertisements ohne gesetztem M-Flag im Netz verteilt und soll der Client dennoch DHCP zur Autokonfiguration verwenden, ist die Aktivierung des DHCPv6-Clients mit folgendem Befehl möglich:

```
Netsh interface ipv6 set interface <id> managed=enabled routerdiscovery=disabled
```

Jetzt kann der Host eine DHCP-Anfrage an den DHCPv6-Server stellen. Jedoch ist das aktiveren des DHCPv6-Clients nur mit dem gleichzeitigen deaktivieren der "Routerdiscovery" möglich. Dies führt zum Ignorieren der Router Advertisements.

Dies hat zur Folge, dass zwar jetzt die IPv6-Adresse und der DNS-Server vom DHCPv6-Server bezogen werden, aber die Default-Route nicht mehr automatisch konfiguriert wird, da die Router Advertisements nicht mehr akzeptiert werden. Somit sollte die Verwendung von DHCPv6 nur mittels gesetztem M-Flag in den Router Advertisements geschehen.

**Szenario 4 (Dualbetrieb):**

Default-Route und eine IPv6-Adresse werden mittels Router Advertisements generiert. Durch das gesetzte M-Flag wird zudem noch der DHCPv6-Server kontaktiert. Von diesem wird der DNS-Server Eintrag und eine weitere IPv6-Adresse bezogen. Der Windows 7 Client erhält in diesem Szenario eine vollständige automatische Konfiguration und ist über zwei IPv6-Adressen gleichzeitig erreichbar.

**Ergebnis:**

Bei Windows 7 ist nur eine Kombination aus Router Advertisements und DHCPv6 möglich, um den Host vollständig automatisch zu konfigurieren (Stateless DHCP oder DHCP per gesetztem M-Flag).

Bei der genaueren Betrachtung der DHCPv6-Nachrichten des Windows 7 Clients stellt sich heraus, dass schon in der ersten Nachricht sechs DHCPv6-Optionen enthalten sind. Neben den Pflicht-Optionen "Client Identifier", "IA-Option", "Elapsed Time" und der empfohlenen Option

"Request Option" existieren in dieser Nachricht auch noch die Optionen für "Vendor Class" und "Fully Qualified Domain Name".

Zudem fällt auf, dass Windows 7 beim Herunterfahren keine Release Nachrichten an den DHCPv6-Server verschickt, um die Adresse wieder freizugeben. Windows speichert die Adressinformationen auf der Festplatte und verwendet diese nach dem Neustart weiter, falls diese noch gültig ist oder er den DHCP-Server nicht erreichen kann.

### 3.2.4 Ubuntu 11.10

Linux bietet generell ab Kernel 2.6 eine produktiv einsetzbare IPv6-Unterstützung. Für DHCPv6 verwendet Ubuntu den DHCP-Client der Organisation "Internet Systems Consortium". Dieser Client bietet ab der Version 11.04 von Ubuntu die Funktionalitäten für DHCPv6. In /etc/network/interfaces kann die Konfiguration der Schnittstellen eingetragen werden. Die Möglichkeit für statische IPv6-Adressen ist schon länger vorhanden. Die Funktionalität für die Verwendung eines DHCPv6-Clients ist auch erst ab Version 11.04 möglich. Auf älteren Systemen als 11.04 muss sich mit nachträglich installierten DHCPv6-Clients wie z. B. mit Dibbler beholfen werden.

**Szenario 1 (SLAAC):**

Durch die folgende Schnittstellenkonfiguration wird die Konfiguration mittels SLAAC ermöglicht:

```
Auto eth1
Iface eth1 inet6 auto
```

Mithilfe der Router Advertisement und deren Präfixinformationen werden die IPv6-Adresse und die Default-Route generiert. Die RDNSS-Option wird standardmäßig nicht unterstützt. Es kann aber ein weiterer Dienst (rdnss daemon) installiert werden, um diese Funktionalität bereitzustellen. Die Privacy Extension kann auch nachträglich über die Schnittstellenkonfiguration aktiviert werden.

**Szenario 2 (Stateless DHCP):**

Ubuntu unterstützt nicht das Stateless DHCP mittels gesetztem O-Flag in den Router Advertisements, da die Flags ignoriert werden. Sollen dennoch weitere Informationen vom DHCPv6-Server bezogen werden, muss dies explizit in der Konfiguration angegeben werden.

```
Auto eth1
Iface eth1 inet6 auto
dhcp 1
```

Die obige Konfiguration ermöglicht eine vollständige Autokonfiguration mittels Stateless DHCP. Die IPv6-Adresse und die Default-Route werden durch die SLAAC ermittelt und die DNS-Server Adresse vom DHCPv6-Server bezogen.

**Szenario 3 (DHCPv6):**

Da die Flags in den Router Advertisements von Ubuntu nicht verarbeitet werden, muss der DHCPv6-Client explizit auf dem Client initialisiert werden, damit dieser DHCP-Request verschickt. Mit der nachfolgenden Konfiguration wird die Schnittstelle automatisch gestartet und für die IPv6-Adressvergabe wird DHCPv6 benutzt.

```
Auto eth1
Iface eth1 inet6 dhcp
```

IPv6-Adresse und weitere Konfigurationsinformationen werden jetzt vom DHCP-Server bezogen. Der Befehl hat jedoch auch zur Folge, dass die Router Advertisements nicht mehr vom Client akzeptiert werden. Dadurch kann die Default-Route nicht mehr gesetzt werden.

Die bisherigen Szenarien haben bis jetzt folgendes gezeigt:

- Die Flags in den Router Advertisements werden nicht verarbeitet, falls die Konfigurationsmethode auf "auto" steht
- Die kompletten Router Advertisements werden nicht verarbeitet, falls die Konfigurationsmethode auf "dhcp" steht

Dadurch wäre eine vollständige Autokonfiguration in diesem Szenario durch keine dieser Konfigurationsmethoden alleine möglich. Im ersten Fall würden keine Informationen vom DHCPv6-Server bezogen und im zweiten Fall würde die Default-Route fehlen. Die folgenden Schnittstellenkonfiguration kann dennoch eine vollständige Autokonfiguration in diesem Szenario ermöglichen.

```
Auto eth1
Iface eth1 inet6 dhcp
Post-up sysctrl –w net.ipv6.conf.eth1.accept_ra=1
```

Es werden nun die IPv6-Adresse und die Adresse des DNS-Servers vom DHCPv6-Server bezogen und nachträglich das Akzeptieren der Router Advertisements wieder eingeschaltet, um die Default-Route zu setzen.

**Szenario 4 (Dualbetrieb):**

Wird in Szenario 4 die gleich Schnittstellenkonfiguration wie in Szenario 3 verwendet, erhält der Client eine vollständige Autokonfiguration, samt zwei IPv6-Adressen. Eine mittels SLAAC und die andere wird vom DHCPv6-Server bezogen.

**Ergebnis:**

Auch bei Ubuntu lässt sich das Clientverhalten nicht so gut mittels den gesetzten Flags in den Router Advertisements steuern. Mit der entsprechenden Schnittstellenkonfiguration und ggf. einem zusätzlichem Daemon zur Auswertung der RDNSS-Option lässt sich in allen Szenarien eine Autokonfiguration durchführen.

### 3.2.5 Ubuntu 10.04 mit Dibbler DHCPv6-Client

Neben der aktuellen Version von Ubuntu wurde auch noch die ältere aber dennoch aktuell Supportete Version 10.04 (long-term support) untersucht. Da wie in Kapitel 3.2.4 schon beschrieben ein DHCPv6-Client in den älteren Systemen nicht direkt integriert war, wurde hierfür auch nachträglich der Dibbler DHCPv6-Client installiert.

**Szenario 1:**

Für Szenario 1 müssen keine Einstellungen an den Schnittstellen vorgenommen werden. Mithilfe der Router Advertisement in deren Präfixinformationen werden die IPv6-Adresse und die Default-Route generiert. Die RDNSS-Option wird standardmäßig nicht unterstützt, kann aber nachträglich mittels eines weiteren Dienstes (rdnss daemon) hinzugefügt werden. Die Privacy Extension kann nicht über die Schnittstellenkonfiguration aktiviert werden.

**Szenario 2 (Stateless DHCP):**

Der Ablauf in Szenario 2 ist der gleiche wie schon bei Windows XP mit Dibbler-Client. Das O-Flag in den Router Advertisements wird von Dibbler nicht verarbeitet, wodurch keine DHCPv6 Information-Request verschickt werden. Es kann dennoch eine Autokonfiguration im Sinne von Stateless DHCP durchgeführt werden, indem die Clientkonfiguration angepasst wird.

```
stateless
iface eth1
{
  option dns-server
}
```

Hierdurch wird die IPv6-Adresse und die Default-Route mittels SLAAC konfiguriert und unabhängig von den gesetzten Flags in den Router Advertisements wird ein Information-Request an den DHCPv6-Server gesendet um die DNS-Server Adresse zu ermitteln.

**Szenario 3:**

Das gleiche gilt für Szenario 3. Hier muss auch wie bei Windows XP die Dibbler Konfigurationsdatei angepasst werden, da die Flags in den Router Advertisements von Dibbler nicht verarbeitet werden.

```
iface eth1 {
  ia
  option dns-server
}
```

Durch die Konfiguration des Clients fragt dieser automatisch (unabhängig vom gesetztem M-Flag) den DHCPv6-Server an und erhält von diesem eine IPv6-Adresse und den DNS-Server zugewiesen. Durch die zusätzlich im Netz verteilten Router Advertisements kann zudem die Default-Route auf dem Client konfiguriert werden. Somit ist eine vollständige automatische Konfiguration des Clients möglich.

**Szenario 4:**

In diesem Szenario hängt die erzeugte Konfiguration wieder von den vorgenommenen Clienteinstellungen ab. Werden für Dibbler zum Beispiel die Einstellungen aus Szenario 3 übernommen, bezieht der Client eine IPv6-Adresse und den DNS-Server Eintrag vom DHCPv6-Server und die Default-Route sowie eine weitere IPv6-Adresse mittels SLAAC.

**Ergebnis:**

Da der Dibbler Client die Flags in den Router Advertisements nicht verarbeitet, ist für jedes Szenario eine Anpassung der Clientkonfiguration notwendig. Danach ist aber eine vollständige Autokonfiguration ist mithilfe von Dibbler in den Szenarien 2, 3 und 4 möglich. In Szenario 1 ist eine vollständige Autokonfiguration mithilfe eines zusätzlichen RDNSS-Daemon möglich.

### 3.2.6 iOS

Das Betriebssystem iOS, welches Apple für das iPhone und das iPad einsetzt, bietet auch eine IPv6-Unterstützung. Diese ist auch nativ aktiviert. Einstellungen hinsichtlich IPv6 kann der Benutzer allerdings nicht machen. Die IPv6-Adresse ist auch nicht über die Standardmenüs sichtbar. Um Einsicht in die IPv6-Parameter zu erhalten, wurde die Applikation "Ip6config" in der Version 1.1 auf dem iPad installiert (Hamilton, 2010). Diese ermöglicht das Anzeigen von allen Schnittstellen samt ihren IPv6-Adressen und weiteren Parametern (wie MTU, MAC-Adresse etc.). Zudem können empfangene Router Advertisements dargestellt werden.

**Szenario 1 (SLAAC):**

Eine vollständige Autokonfiguration ist in diesem Szenario möglich. iOS unterstützt eine SLAAC und zudem noch die Verteilung des DNS-Servers mittels der RDNSS-Option. Empfängt das iPAD Router Advertisements mit Präfixinformationen, generiert es automatisch eine IPv6-Adresse und die Default-Route wird gesetzt. Ist in den Router Advertisements noch die RDNSS-Option enthalten, wird auch diese ausgewertet und die Adresse des DNS-Servers konfiguriert. Die IPv6-Adresse wird zudem automatisch per Privacy Extension generiert.

**Szenario 2 (Stateless DHCP):**

Wie schon in Szenario 1 gezeigt, funktioniert die SLAAC in iOS. Aber auch auf das gesetzt O-Flag reagiert das iOS korrekt und schickt daher dem DHCPv6-Server einen Information-Request, um zusätzliche Informationen zu erfragen. Dieser enthält die Request-Options Werte 23 und 24. Es wird also beim DHCP-Server der DNS Recursive Name Server und die Domain Search List angefodert. Die Adresse des DNS-Server wird somit über den DHCPv6-Server ermittelt. Also liegt auch hier eine vollständige Autokonfiguration vor.

**Szenario 3 (DHCPv6):**

Durch das gesetzte M-Flag wird vom iOS-Client mittels Solicit-Nachricht nach einem DHCPv6-Server gesucht und eine Request-Nachricht versendet. Diese enthält eine Anfrage für eine IPv6-Adresse und wie auch schon beim Information-Request die Bitte nach Request-Option 23 und 24. Die Konfiguration von IPv6-Adresse und DNS-Server Eintrag läuft dadurch über den DHCPv6-Server ab. Die Default-Route wird mithilfe der empfangenen Router Advertisements gesetzt.

Für den Benutzer gibt es keine Möglichkeit den DHCPv6-Client zu aktivieren. Dieser startet von alleine, wenn der Client Router Advertisements mit gesetztem M- oder O-Flag empfangen werden. Es können auch nicht manuell eine IPv6-Adresse und DNS-Server konfiguriert verwenden. Somit kann DHCPv6 nur in Verbindung mit gesetztem M-Flag in den Router Advertisements benutzt werden.

**Szenario 4 (Dualbetrieb):**

Auch die Kombination von Stateful und Stateless Konfiguration wird unterstützt. Indem in den Router Advertisements eine Präfixinformation vorhanden ist und das M-Feld gesetzt ist, generiert das iPAD gleichzeitig eine IPv6-Adresse mittels Stateless Adresskonfiguration und bezieht eine weitere vom DHCPv6-Server. Die Default-Route wird mittels Router Advertisements gesetzt. Die Adresse des DNS-Server erhält der Client zwar mittels beider Verfahren, aber die Adresse die der DHCP-Server verteilt hat Vorrang.

**Ergebnis:**

Die Autokonfiguration von iOS ist in jedem Szenario möglich.

### 3.2.7 Mac OS X

Mac OS X bietet seit Version 10.2 IPv6-Unterstützung. Aber erst ab der Version 10.7 (Codename Lion) DHCPv6-Funktionalitäten (Apple Inc., 2011). Für die Untersuchung stand nur ein MacBook mit der OS X Version 10.5.6 zur Verfügung.

**Szenario 1 (SLAAC):**

Die IPv6-Adresse und die Default-Route können mithilfe der Router Advertisements und SLAAC in Szenario 1 automatisch konfiguriert werden. Ist eine Generierung der IPv6-Adresse mittels Privacy Extension (RFC 4941) gewünscht, muss dies explizit eingeschaltet werden (Befehl: `sysctl net.inet6.ip6.use_tempaddr=1`). Die RDNSS-Option unterstützt Mac OS X in der Version nicht, wodurch hierrüber kein DNS-Server gesetzt werden kann. Eine vollständige Konfiguration ist somit nicht möglich.

**Szenario 2 (Stateless DHCP):**

Mithilfe der Router Advertisements und der darin enthaltenen Präfixinformationen kann auf dem Client eine IPv6-Adresse und die Default-Route konfiguriert werden. Das O-Flag signalisiert dem Client weitere Informationen mittels eines DHCPv6 Information-Request anzufordern. Dies ist durch den fehlenden DHCPv6-Client nicht möglich, wodurch der DNS-Server Eintrag nicht konfiguriert wird.

**Szenario 3 (DHCPv6):**

Wie schon bei Windows XP, können in Szenario 3 keine Informationen von dem DHCPv6-Server verarbeitet werden, da kein DHCPv6-Client vorhanden ist. Es wird somit keine IPv6-Adresse und DNS-Server konfiguriert. Nur die Default-Route wird mithilfe der Router Advertisements gesetzt.

**Szenario 4 (Dualbetrieb):**

Das MacBook mit dem Betriebssystem OS X 10.5.6 kann sich nur eine Konfiguration mittels SLAAC automatisch erstellen (ohne DNS-Server). Die gleichzeitige Verwendung von DHCPv6, durch das gesetzte M-Flag ist durch den fehlenden DHCPv6-Client nicht möglich.

**Ergebnis:**

Über IPv6 existiert keine Möglichkeit, automatisch DNS-Server Adressen an den Mac OS X Client zu verteilen. Mac OS X 10.5.6 eignet sich somit auch nur in einer gemischten IPv4/IPv6 Umgebung, da hier die DNS-Server Adresse über IPv4 und DHCP empfangen werden kann.

### 3.2.8 Android

Android verfügt seit der Version 2.1 IPv6-Funktionalitäten auf der WLAN Schnittstelle. Die Einstellungen bzw. die Sichtbarkeit von IPv6-Informationen sind für den Benutzer gering. Abhilfe schaffen zusätzliche Apps, wie "IPv6 and More". Mithilfe der App sind die IPv6-Adressen sichtbar, es können Pings durchgeführt und die Privacy Extension aktiviert werden (Sen, 2012). IPv6 ist standardmäßig aktiviert.

**Szenario 1 (SLAAC):**

Die IPv6-Adresse und die Default-Route können mithilfe der Router Advertisements und SLAAC in Szenario 1 automatisch konfiguriert werden. Die Aktivierung der Privacy Extension ist zum Beispiel mit der oben genannten App möglich. Die RDNSS-Option unterstützt Andriod in der Version nicht, wodurch hierrüber kein DNS-Server gesetzt werden kann. Eine vollständige Konfiguration ist somit nicht möglich.

**Szenario 2 (Stateless DHCP):**

Auch Android kann mittels dem O-Flag nicht aufgefordert werden DHCPv6 Information-

Requests zu erzeugen, da kein DHCPv6-Client vorhanden ist. Mithilfe der Router Advertisements und der darin enthaltenen Präfixinformationen kann auf dem Client nur eine IPv6-Adresse und die Default-Route konfiguriert werden.

**Szenario 3 (DHCPv6):**

In Szenario 3 werden auch bei Android aufgrund des fehlenden DHCPv6-Clients keine Request an den DHCPv6-Server geschickt, wenn dieser Router Advertisements mit gesetztem M-Flag empfängt. Es wird somit keine IPv6-Adresse und DNS-Server konfiguriert. Nur die Default-Route könnte mithilfe der Router Advertisements gesetzt werden.

**Szenario 4 (Dualbetrieb):**

Das Handy mit dem Betriebssystem Android 4.03 kann sich nur eine Konfiguration mittels SLAAC automatisch erstellen (ohne DNS-Server). Die gleichzeitige Verwendung von DHCPv6, durch das gesetzte M-Flag ist durch den fehlenden DHCPv6-Client nicht möglich.

**Ergebnis:**

Da es auch bei Android in einer reinen IPv6-Umgebung keine Möglichkeit gibt, die DNS-Server Adresse auf dem Client automatisch zu setzten, eignet sich Android ebenfalls nur in einer gemischten IPv4 / IPv6 Umgebung.

### 3.2.9 Überblick

Tabelle 4 gibt einen zusammenfassenden Überblick über die einzelnen Clients und deren unterstützten Funktionen. IPv6 wird von allen Clients generell unterstützt. Nur bei Windows XP ist IPv6 standardmäßig nicht aktiviert. Der Grund hierfür liegt wahrscheinlich darin, dass es das älteste Betriebssystem in der Auswahl ist. Stateless Address Autoconfiguration wird von allen Betriebssystemen unterstützt. Das Gleiche gilt für die Privacy Extension, die bei einigen aber erst manuell aktiviert werden muss. Zudem generieren alle Systeme ihre DUID, wie empfohlen, mittels Link-Layer Adresse plus der Zeit.

Ubuntu 11.10, Windows 7 und iOS 5 verfügen über einen DHCPv6-Client und unterstützen im Gegensatz zu den anderen Systemen die RDNSS-Option (was ggf. mit einer weiteren Installation verbunden ist). Auch ist es bei diesen Systemen möglich, dass sie sich eine IPv6-Adresse mittels Stateless Adresskonfiguration generieren und gleichzeitig eine zweite IPv6-Adresse vom DHCPv6-Server beziehen. Somit sind sie unter zwei Adressen aus zwei unterschiedlichen Konfigurationsverfahren erreichbar.

Windows XP, Mac OS X 10.5.6 und Android 4.03 sind nur in einer gemischten IPv4 und IPv6 Umgebung sinnvoll, da sie nicht automatisch über IPv6 mit einem DNS-Server Eintrag versorgt werden können.

| Funktion / OS | IPv6 | Nativ aktiviert | SLAAC | RDNSS | Privacy addresses | DHCPv6 | Dual Stack erforderlich[3] |
|---|---|---|---|---|---|---|---|
| **Windows XP** | Ja | Nein | Ja | Nein | Ja | Nein[7] | Ja |
| **Windows XP & Dibbler Client** | Ja | Nein | Ja | Nein | Ja | Ja | Nein |
| **Windows 7** | Ja | Ja | Ja | Nein[4] | Ja | Ja | Nein |
| **Ubuntu 10.04** | Ja | Ja | Ja | Nein[5] | Nein | Nein | Ja |
| **Ubuntu 10.04 & Dibbler Client** | Ja | Ja | Ja | Nein[5] | Nein | Ja | Nein |
| **Ubuntu 11.10** | Ja | Ja | Ja | Nein[5] | Ja[6] | Ja | Nein |
| **iOS 5** | Ja | Ja | Ja | Ja | Ja | Ja[2] | Nein |
| **Mac OS X 10.5.6** | Ja | Ja | Ja | Nein[1] | Ja[6] | Nein | Ja |
| **Android 4.03** | Ja | Ja | Ja | Nein | Ja[6] | Nein | Ja |

Tabelle 4: Übersicht der getesteten IPv6-Autokonfigurationsmerkmale

[1] Erst ab Version Mac OS X 10.7 (Lion) vorhanden
[2] Nur in Verbindung mit Router Advertisements
[3] Diese Spalte gibt an, ob der Client nur über eine IPv6 Verbindung alle Informationen wie IPv6-Adresse, Default Route und DNS-Server beziehen kann
[4] Per externes Programm rdnssd-win32 aber möglich
[5] Nur per zusätzlichem Dämon möglich
[6] Manuelle Aktivierung erforderlich
[7] DHCPv6 aber mittels zusätzlichen Programmen (DHCPv6-Clients) wie z. B Dibbler möglich

# 4 Zusammenfassung

Selbst viele Jahre nach der IPv6 Einführung bieten nicht alle Clients den vollen Umfang an möglichen Autokonfigurationstechnologien. Eine IPv6-Adresskonfiguration mittels SLAAC wird jedoch von allen Systemen unterstützt. Aber die Hersteller haben eine unterschiedliche Vorgehensweise, wie ihre Systeme zu einer automatischen Adresskonfiguration gelangen. Auch die Einsicht oder der Einfluss des Benutzers in die Netzkonfiguration ist nicht immer vorhanden. Dies führt nicht immer zu mehr Benutzerfreundlichkeit.

Nützlich ist vor allem die nachträglich entstandene RDNSS-Option für Router Advertisements. Durch diese Option können Clients jetzt per SLAAC mit einer IPv6-Adresse, Default-Route und (was bis dahin fehlte) auch mit der IPv6-Adresse des DNS-Servers automatisch konfiguriert werden. Dadurch ist die Beschränkung auf nur noch ein Protokoll möglich und Endgeräte können auf einen DHCP-Client verzichten. Zwar unterstützen nicht alle Clients diese Option, aber da dieser Standard aus dem Jahr 2010 und damit relativ neu ist, ist es möglich, dass diese Funktion in Zukunft in mehr Clients aufzufinden ist.

Für den Netzverwalter hat sich zumindest der Weg über Router Advertisements mit gesetztem M-Flag und Präfixinformationen samt DHCPv6 als sinnvoll erwiesen, um möglichst viele verschiedene Systeme automatisch mit einer Adresskonfiguration zu versorgen. Mittels diesem zweigleisigen Verfahren (SLAAC und DHCPv6) ist gewährleistet, dass alle Systeme zumindest eine IPv6-Adresse und ein Default Gateway erhalten. Die Systeme, die zusätzlich noch über einen DHCPv6-Client verfügen, werden zudem noch mit einem DNS-Server Eintrag versorgt. Einige Systeme haben können trotz alledem nur in einer gemischten IPv4-/ IPv6-Umgebung sinnvoll betrieben werden, da ihnen die Möglichkeit bei IPv6 fehlt den DNS-Server automatisch zu ermitteln.

# 5 Ausblick

Die (Weiter-)Entwicklung an Teilen der automatischen Adresskonfiguration für IPv6 ist immer noch nicht abgeschlossen. Daher wird hier noch ein Entwurf (Internet-Draft) einer zusätzlichen Erweiterung dargestellt. Internet-Drafts sind unverbindliche Arbeitspapiere, die keinen formalen Zustand besitzen und maximal sechs Monate gültig sind. Diese sind ggf. später Grundlage für einen eigenen Standard.

Der hier vorgestellte Entwurf geht in eine ähnliche Richtung, wie die oben genannte RDNSS-Option, jedoch kann hier statt auf DHCP auf Router Advertisements verzichtet, um eine vollständige Adresskonfiguration durchzuführen. Dieses Verfahren wird in dem Internet-Draft "DHCPv6 Route Options" vorgestellt (Dec, 2012). Diese Prozedur bietet die Möglichkeit, per DHCPv6 die Default-Route des Clients zu konfigurieren. DHCPv6 bot bis jetzt keine Option dafür (im Gegensatz zu DHCPv4), wodurch nur eine statische Konfiguration der Default-Route oder die Ermittlung mittels Router Advertisements möglich war. Durch den Internet-Draft besteht nun die Möglichkeit, komplett auf Router Advertisements zu verzichten (Verteilung von Router Advertisements von Routern und Verarbeitung der Router Advertisments auf dem Client). So ist ausgeschlossen, dass fehlerhafte oder böswillig falsche Router Advertisments durch Setzten der Default-Route den Netzverkehr umleiten. Auch reduziert die Verwendung von nur noch einem Protokoll die Komplexität.

Die Routing-Informationen werden über eine weitere DHCPv6 Option verteilt bzw. können vom DHCPv6-Client angefragt werden. Da es sich aber noch um einen Internet-Draft handelt, hat die IANA auch noch keinen Options-Wert dafür spezifiziert. Dibbler 0.8.1 unterstützt die Verteilung der Routing-Informationen mittels dieser DHCPv6 Option schon. Da der Option-Wert noch nicht spezifiziert ist, hat Dibbler einen eigenen Wert (Option-Wert 242) definiert (Mrugalsk, 2011). Falls sich der Internet-Draft als Standard etabliert, wird sich dieser Wert noch ändern. Da der Internet-Draft nur bis zum 27.12.2012 gültig ist, stellt sich die Frage, ob weitere Anstrengungen hinsichtlich dieses Entwurfes gemacht werden oder demnächst sogar schon erste Entwürfe eines RFCs vorhanden sind.

Durch die RDNSS-Option und den vorgestellten Internet-Draft könnte in Zukunft entweder auf Stateful oder auf Stateless Address Autoconfiguration verzichtet werden, da IPv6-Adresse, Default-Route und DNS-Server über beide Verfahren verteilt werden können. Möglicherweise eignet sich eines dieser Verfahren in Theorie und Praxis besser zur alleinigen Autokonfiguration von Clients. Auch besteht die Möglichkeit die genauen Vor- und Nachteile zu betrachten, die es mit sich bringt, nur noch eines dieser Verfahren in einem Netz zu verwenden.

# 6 Literaturverzeichnis

**Apple Inc. 2011.** OS X for UNIX Users. [Online] Juli 2011. [Zitat vom: 2. März 2012.] http://images.apple.com/macosx/docs/OSX_for_UNIX_Users_TB_July2011.pdf.

**Dec, et al. 2012.** DHCPv6 Route Options. *draft-ietf-mif-dhcpv6-route-option-04.* s.l. : Internet Engineering Task Force, 2012.

**Deering S., Hinden R. 1998.** Internet Protocol, Version 6 (IPv6) Specification. *RFC 2460.* s.l. : Internet Engineering Task Force, 1998.

**Droms, et al. 2003.** Dynamic Host Configuration Protocol for IPv6 (DHCPv6). *RFC 3315.* s.l. : Internet Engineering Task Force, 2003.

**Droms, R. 2004.** Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6. *RFC 3736.* s.l. : Internet Engineering Task Force, 2004.

**Hagen, S. 2006.** *IPv6 Essentails.* 2. Auflage. Sebastopol : O'Reilly, 2006.

**Hamilton, J. 2010.** Ip6config. [Online] 2010. [Zitat vom: 2. März 2012.] http://www.jhtech.us/projects/ip6config.

**Internet Systems Consortium. 2012.** ISC DHCP. [Online] Internet Systems Consortium (ISC), 2012. [Zitat vom: 26. Februar 2012.] http://www.isc.org/software/dhcp.

**Jeong, et al. 2010.** IPv6 Router Advertisement Options for DNS Configuration. *RFC 6106.* s.l. : Internet Engineering Task Force, 2010.

**Litech Systems Design. 2012.** Linux IPv6 Router Advertisement Daemon. [Online] 26. Januar 2012. [Zitat vom: 26. Februar 2012.] http://www.litech.org/radvd/.

**Mrugalsk, T. 2011.** DHCPv6: Dibbler - a portable DHCPv6. *User's guide.* [Online] 11. Oktober 2011. [Zitat vom: 14. Januar 2012.] http://klub.com.pl/dhcpv6/doc/dibbler-user.pdf.

**Narten, et al. 2007.** Neighbor Discovery for IP version 6 (IPv6). *RFC 4861.* s.l. : Internet Engineering Task Force, 2007.

**Palmer, C. 2012.** Does Win7 or W2K8 server support RFC 6106? [Online] 22. Januar 2012. [Zitat vom: 1. März 2012.] http://social.technet.microsoft.com/Forums/en-US/ipv6/thread/5757980a-5983-4efc-a5f3-27687b90fe41.

**Refsnes Data. 2012.** OS Platform Statistics. [Online] Februar 2012. [Zitat vom: 26. Februar 2012.] http://www.w3schools.com/browsers/browsers_os.asp.

**Sen, R. 2012.** IPv6 and More. [Online] 2012. [Zitat vom: 3. März 2012.] http://ipv6andmore.blogspot.com/.

**Thomson, et al. 2007.** IPv6 Stateless Address Autoconfiguration. *RFC 4862.* s.l. : Internet Engineering Task Force, 2007.