Hochschule
Bonn-Rhein-Sieg
University
of Applied Sciences

Fachbereich Informatik
Department of Computer Sciences

Abschlussarbeit

im Studiengang Master Informatik

Einsatz von IPv6 Anycast, Multicast und Unicast am Beispiel von DNS - Analyse und Vergleich

von

Christian Schneider

Erstbetreuer	Prof. Dr. Martin Leischner Hochschule Bonn-Rhein-Sieg
Zweitbetreuer	Prof. Dr. Kerstin Uhde Hochschule Bonn-Rhein-Sieg
eingereicht am	24.06.2013

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 24.06.2013

Christian Schneider

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungen

ARP	Address Resolution Protocol
BSI	Bundesamt für Sicherheit in der Informationstechnik
DHCPv6	Dynamic Host Configuration Protocol Version 6
DNS	Domain Name System
DNSSEC	Domain Name System Security Extensions
DoS	Denial of Service
EDNS	Extended Domain Name System
FQDN	Fully Qualified Domain Name
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IPv6	Internet Protocol Version 6
ITU	International Telecommunication Union
MAC-Adresse	Media-Access-Control-Adresse
Mgmt	Management
MLD	Multicast Listener Discovery
NDP	Neighbor Discovery Protocol
NTP	Network Time Protocol
OSPF	Open Shortest Path First
PIM	Protocol Independent Multicast
RDNSS	Recursive Domain Name System Server
RFC	Requests for Comments
RIP	Routing Information Protocol
SLAAC	Stateless Address Autoconfiguration
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
TTL	Time to Live
UDP	User Datagram Protocol
ULA	Unique Local Addresses
WLAN	Wireless Local Area Network

1 Einleitung

Mit der Einführung von IPv6 rücken neben Unicast weitere Kommunikationsmodelle wie Multicast oder Anycast in den Mittelpunkt. Diese waren in IPv4 erst nachträglich hinzugekommen und fanden keine große Verbreitung. In IPv6 sind Anycast und Multicast seit der ersten Standardisierung fester Bestandteil [Deering S., 1998].

Es stellt sich die Frage, ob die Kommunikationsmodelle für IPv6 auch als eine Alternative zur Unicast-Kommunikation verwendet werden können und welche Vorteile bzw. Nachteile dies mit sich bringt.

Das bekannteste Beispiel für IPv6-Anycast ist die Anbindung einiger DNS-Root Server [RIPE NCC, 2013]. Unter anderem dient dies der Erhöhung der Verfügbarkeit der DNS-Root Server im Internet. Multicast eignet sich besonders für Anwendungen, an denen mehrere Empfänger teilnehmen, wie Audio- und Videostreaming. Aber auch hier existieren Ansätze für die Verwendung mit DNS [Strotmann, 2012].

1.1 Zielsetzung

Ziel der Arbeit ist es, die Kommunikationsmodelle Unicast, Anycast und Multicast am Beispiel von DNS in einer IPv6-Umgebung zu vergleichen. Untersucht wird, welche Eigenschaften diese Modelle besitzen und welche Vor- und Nachteile dies für den Einsatz von DNS mit sich bringt.

DNS wurde hierbei gewählt, da es einer der zentralen und wichtigsten Dienste im Internet ist, dessen Verfügbarkeit für den größten Teil der Kommunikation unabdingbar ist. Daher ist es grundsätzlich sinnvoll, die Kommunikation dieses Dienstes zu betrachten und ggf. Potenzial für Veränderungen aufzuzeigen. Zudem ist die Client-Server-Kommunikation einfach gestaltet, da in der Regel nur einzelne UDP-Pakete für eine Namensauflösung ausgetauscht werden. Diese Einfachheit erlaubt es, die Unterschiede der Kommunikationsmodelle deutlicher darzustellen.

1.2 Methodik

Für den Vergleich der IPv6-Kommunikationsmodelle (Anycast, Multicast und Unicast) anhand von DNS werden zunächst mittels Literaturrecherche geeignete Kriterien ermittelt. Für eine spätere Überführung der Ergebnisse auf andere Dienste werden die Kriterien möglichst allgemein gehalten. Eine Validierung der ermittelten Kriterien erfolgt mithilfe einer Unternehmensbefragung. Die einzelnen Kriterien werden anschließend durch die Erstellung von Testszenarien konkretisiert, um eine Anwendung auf eine DNS-Infrastruktur zu ermöglichen. Die Tests finden in einer für diese Arbeit eigens erstellten DNS-Umgebung statt. Für die drei Kommunikationsmodelle wird die Umgebung jeweils entsprechend an-

gepasst. Die Tests werden gruppiert nach Kriterium durchgeführt und deren Ergebnisse dargestellt. Für den Vergleich der Kommunikationsmodelle werden die einzelnen Ergebnisse ausgewertet und je Kommunikationsmodell wiedergegeben. Abschließend findet eine Überführung der Ergebnisse auf weitere Dienste statt.

1.3 Aufbau der Arbeit

Für den Vergleich der einzelnen Kommunikationsmodelle werden zunächst ihre grundlegenden Eigenschaften in Kapitel 2 dargestellt. Anschließend folgen im Abschnitt 3 der Aufbau und die Beschreibung einer DNS-Infrastruktur für eine organisatorische Einheit (Unternehmensnetz), die für den späteren Vergleich der Kommunikationsmodelle verwendet wird.

DNS ist schon lange standardisiert und für IPv6 existieren nur wenige grundlegende Veränderungen [Thomson, 2003]. Daher werden mithilfe einer Literaturrecherche im Umfeld von DNS und durch einzelne Unternehmensbefragungen geeignete Kriterien für den Vergleich der einzelnen Kommunikationsmodelle (Unicast, Anycast, Multicast) ermittelt. Dementsprechend werden in Kapitel 4 die Testszenarien erstellt. Mittels dieser Testszenarien und der vorher festgelegten Kriterien werden anschließend die einzelnen Kommunikationsmodelle verglichen. Die Ergebnisse des Vergleichs werden im Abschnitt 5 dieser Arbeit dargestellt. Das nachfolgende Kapitel enthält die Auswertung der Ergebnisse und deren Überführung auf weitere Dienste. Abschließend folgen eine Zusammenfassung der Arbeit und ein Ausblick.

2 Grundlagen

Neben den Unicast-Adressen (Link-Lokale, ULA etc.) existieren bei IPv6 mit IPv6-Multicast und IPv6-Anycast weitere Adresstypen. Diese sind – im Gegensatz zu IPv4 – fester Bestandteil von IPv6 und sollten von allen IPv6-Nodes unterstützt werden [Jankiewicz, et al., 2011].

Mithilfe dieser Adresstypen ergeben sich weitere Kommunikationsmodelle, die im weiteren Verlauf dieser Arbeit anhand von DNS untersucht werden. Dazu werden zunächst die einzelnen Adresstypen beschrieben, um die grundsätzlichen Eigenschaften der daraus resultierenden Kommunikationsmodelle darzustellen. Anschließend erfolgt eine Beschreibung des Domain Name System (DNS), samt der Erweiterungen für IPv6, Extended DNS (EDNS) und DNS Security Extensions (DNSSEC).

2.1 Unicast

Unicast bezeichnet eine Übertragung von Nachrichten an einen einzelnen eindeutigen Empfänger (Adresse). Zu den Unicast-Adressen in IPv6 wird eine ganze Reihe von Adresstypen gezählt:

- Globale-Unicast-Adressen
- Link-Lokale-Adressen
- Unique-Lokale-Adressen (ULA)
- Spezielle Adressen (z. B. localhost ::1)

Unicast-Adressen sind Adressen, die an eine Netzschnittstelle gebunden werden. Ein Beispiel einer Unicast-IPv6-Adresse ist die 2001:db8::1 (Global-Unicast) oder fe80:1234:abfe:ffcd:5678::1 (Link-Lokal). Im Gegensatz zu IPv4 ist es bei IPv6 üblich, dass mehrere IPv6-Adressen an eine einzelne Schnittstelle gebunden werden.

2.1.1 Adressformat

Die einzelnen Adresstypen, sowie das Adressformat sind im RFC 4291 und für die ULA-Adressen im RFC 4193 zu finden.

2.1.2 Kommunikation mittels Unicast

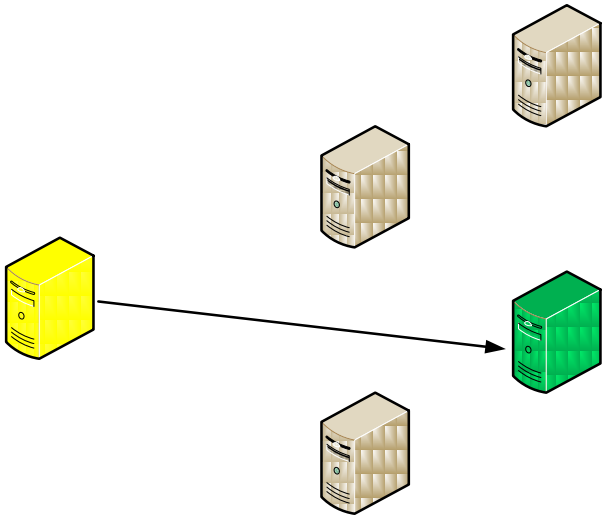

Abbildung 1: Unicast-Kommunikation

Mittels einer Unicast-Adresse wird ein eindeutiger Node identifiziert, hier grün dargestellt. Eine Unicast-Kommunikation ist somit eine 1-zu-1 Beziehung z. B. zwischen Client (gelb) und Server (grün).

2.2 Anycast

Anycast ist ein weiteres Kommunikationsmodell für IPv6. Es ermöglicht das Versenden von Nachrichten an ein einzelnes Ziel aus einer Gruppe von Zielen. Erstmalig wurde es für IPv4 in dem RFC 1546 von 1993 beschrieben. Es sah einen separaten Adressbereich vor. In der Praxis wurde diese Variante aber nie richtig implementiert. Für IPv6 ist Anycast erstmals im Standard RFC 1884 definiert.

Eine IPv6-Anycast-Adresse ist eine IPv6-Adresse, die an mehr als eine Schnittstelle gebunden ist, typischerweise an verschiedene Nodes. Pakete, die an eine Anycast-Adresse versendet werden, werden an die nächstgelegene Schnittstelle geroutet, die diese Adresse besitzt.

Aus Sicht des Senders besteht zwischen Unicast und Anycast kein Unterschied. Der Sender besitzt keine Informationen darüber, wer der nächstgelegene Node ist, dies übernimmt das Routing System. Der Sender hat somit keine Kontrolle über die Auswahl des Anycast-Nodes.

2.2.1 Adressformat

Anycast-Adressen sind dem Unicast-Adressbereich zugeordnet und besitzen das gleiche Format. Sie sind dadurch syntaktisch nicht von einer Unicast Adresse zu unterscheiden. In dem Moment, in dem sie mehr als einer Schnittstelle zugeordnet sind, werden sie zu Anycast-Adressen. Den Nodes, denen diese

Adresse zugewiesen wurde, muss explizit signalisiert werden, dass es sich um eine Anycast-Adresse handelt [Deering, et al., 2006].

Wie bei Multicast existieren auch für Anycast fest reservierte Anycast-Adressen, z. B. für die Kommunikation mit Home-Agents im Bereich von Mobile-IPv6. Die reservierten Anycast-Adressen sind im RFC 2526 definiert.

2.2.2 Kommunikation mittels Anycast

Nachfolgend werden zwei Szenarien einer möglichen Anycast-Kommunikation vorgestellt. Bei der ersten vorgestellten Kommunikation handelt es sich um eine Kommunikation innerhalb eines Subnetzes. Das bedeutet, dass sowohl die Sender als auch die Empfänger (Nodes mit der gleichen Anycast-Adresse) sich in einem Subnetz befinden. Damit zwei Nodes über IPv6 kommunizieren können, muss der Sender erst die zugehörige MAC-Adresse der IPv6-Adresse ermitteln. Dazu fragt er mittels des Neighbor Discovery Protocol im lokalen Netz nach der IPv6-Adresse. Alle Nodes, die die IPv6-Adresse besitzen, antworten auf die Anfrage und teilen dem Sender ihre MAC-Adresse mit. Der Sender speichert anschließend die Zuordnung von IPv6-Adresse und MAC-Adresse in seinem Neighbor Cache und benutzt den Eintrag für die zukünftige Kommunikation mit dieser IPv6-Adresse.

Bei IPv6-Anycast können sich jedoch mehrere Nodes mit der gleichen IPv6-Adresse in demselben Subnetz befinden. In diesem Szenario ist es daher relevant, dass die IPv6-Anycast-Adresse auf den Nodes auch als solche markiert ist. Das verursacht ein anderes Verhalten bei dem Neighbor Discovery Protocol. Das Antwortpaket einer Adressauflösung wird verzögert übertragen (zwischen 0 und 1 Sekunden), um eine Netzüberlastung zu vermeiden. Außerdem wird im Antwortpaket (Neighbor Advertisement) das Override-Flag auf 0 gesetzt. Das bewirkt, dass im Neighbor Cache die MAC-Adresse des ersten Antwortpaketes geschrieben (falls noch keine vorhanden ist) und nicht von anschließend eintreffenden Antwortpaketen überschrieben wird [Narten, et al., 2007]. Das Verhalten hat zur Folge, dass der Sender mit dem Node kommuniziert, der am schnellsten auf die Adressauflösung reagiert. In der Regel ist dies auch derjenige Node, der generell aus Sicht des Senders am besten angebunden ist.

Das zweite Szenario beschreibt eine Kommunikation über mehrere Netze hinweg. Abbildung 2 veranschaulicht eine Kommunikation mittels IPv6-Anycast. Alle Nodes mit derselben IPv6-Anycast-Adresse sind dabei grün dargestellt. Der Sender, der ein Paket an eine IPv6-Anycast-Adresse sendet, ist gelb, alle weiteren Nodes sind in grau abgebildet. Anhand der Routing-Metrik wird das Paket von den Routern an den nächstgelegenen Node geleitet. Abhängig davon, wo sich der Client im Netz befindet und wie die Routingpfade konfiguriert sind bzw. wie stabil diese sind, können der Weg zum Ziel oder das Ziel selbst für jedes Paket variieren.

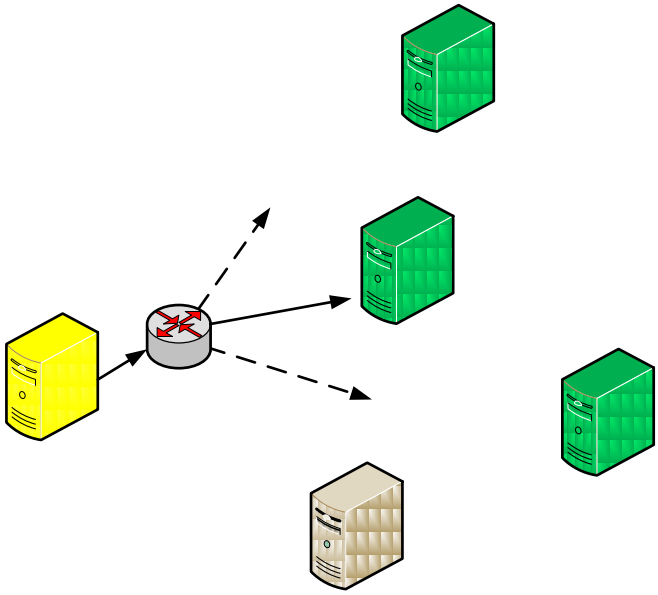

Abbildung 2: Anycast-Kommunikation

2.2.3 Einsatzbeispiele

Praktische Verwendung findet IPv6-Anycast unter anderem bei der Anbindung der Root DNS-Server. So befinden sich unter der IPv6-Adresse 2001:7fd::1 zur Zeit fünf verschiedene global erreichbare Nodes des K-Root Servers. Diese sind über die ganze Welt verteilt. Ein Node befindet sich z. B. in Miami/Florida USA, einer hier in Deutschland (Frankfurt am Main) und ein weiterer in Tokio, Japan. Je nachdem woher auf der Welt eine Anfrage an einen K-Root Server versendet wird, wird diese anhand des Anycast-Routings an den nächstgelegenen (abhängig von der Routing-Metrik) Node weitergeleitet [RIPE NCC, 2013].

2.3 Multicast

Multicast bezeichnet eine Kommunikation an eine Menge von Empfängern. In IPv4 war die Unterstützung von Multicast noch optional, bei IPv6 ist sie Pflicht [Davies, 2012]. Jedem Empfänger dieser Menge wird hierfür die gleiche IPv6-Multicast-Adresse zugeteilt. Diese Menge an Hosts, die auf eine IPv6-Multicast-Adresse hört, wird Multicast-Gruppe genannt. Alle Pakete an eine Multicast-Gruppe werden an alle Mitglieder gleichzeitig übermittelt. Die Anzahl an Hosts einer Multicast-Gruppe ist nicht begrenzt. Außerdem können Hosts eine Gruppe jederzeit verlassen oder betreten. Um Pakete an eine Multicast-Gruppe zu senden, ist es nicht notwendig, selbst Mitglied dieser Gruppe zu sein. Multicast-Gruppen können sich über mehrere Netze verteilen. Dazu ist IPv6-Multicast Support bei den Routern erforderlich und die Hosts müssen Multicast Listener Discovery unterstützen (siehe Kapitel 2.3.3). Im Gegensatz zu Unicast- und Anycast-Adressen dürfen Multicast-Adressen nicht als Source-Adressen oder in IPv6-Extension-Routing-Headern benutzt werden [Deering, et al., 2006].

Durch die zwingende Unterstützung von Multicast in IPv6 findet es eine größere Verwendung. So löst Multicast in IPv6 die Broadcast-Kommunikation von IPv4 ab. Dafür existieren vordefinierte Multicast-Gruppen wie beispielsweise die ff02::1, die alle Hosts in einem Subnetz beinhaltet.

2.3.1 Adressformat

IPv6-Multicast-Adressen besitzen einen eigenen Adressraum und lassen sich dadurch eindeutig von Unicast- und Anycast-Adressen unterscheiden.

Folgende Tabelle zeigt den Aufbau von IPv6-Multicast-Adressen:

8 Bits								4 Bits				4 Bits	112 Bits
1	1	1	1	1	1	1	1	0	R	P	T	Scope	Group ID
Festes Präfix								Flags					

Anhand des festen Präfixes lässt sich erkennen, dass alle IPv6-Multicast-Adressen aus dem Bereich FF00::/8 sind. Die Flags kennzeichnen verschiedene Arten von Multicast-Adressen, wie zum Beispiel permanente oder dynamische Multicast-Adressen:

- Kein Flag gesetzt:
  Permanent definierte wohlbekannte Multicast-Adressen (von der IANA zugewiesen)
- T-Bit gesetzt:
  Transient (vorübergehend) oder dynamisch zugewiesene Multicast-Adressen
- P-Bit gesetzt, erzwingt das T-Bit:
  Unicast-Prefix-based Multicast-Adressen (RFC 3306)
- R-Bit gesetzt, erzwingt P- und T-Bit:
  Multicast-Adressen, welche die Adresse des Rendezvous Point enthalten (RFC 3956)

Der Scope-Wert einer Multicast-Adresse gibt den Gültigkeitsbereich der Adresse an. Beispielweise ist die oben genannte Adresse ff02::1 (All Nodes) nur link-lokal gültig, wie am Scope-Wert von "2" zu erkennen ist. Sie wird somit nicht geroutet und erreicht nur Nodes im selben Subnetz. Die Router entscheiden anhand des Gültigkeitsbereichs, ob das Paket weiter geroutet wird oder seinen maximalen Gültigkeitsbereich erreicht hat. Eine Liste der Scope-Werte und deren Zuordnung enthält die nachfolgende Tabelle (nicht definierte Werte sind nicht zugewiesen):

Scope-Wert	Scope Beschreibung
0	reserviert
1	interfacelokal
2	link-lokal
3	reserviert
4	admin-lokal
5	site-lokal
8	organisationslokal
E	globaler Multicast
F	reserviert

Tabelle 1: Bedeutung der Scope-Werte

Vordefinierte Multicast-Adressen sind unter anderem (x entspricht einem beliebigen Scope-Wert):

- FF02::1 All Nodes
- FF02::2 All Router
- FF0x::114 Experimental
- FF0x::101 Network Time Protocol (NTP)

Eine vollständige Liste aller fest definierten Multicast-Adressen befindet sich auf der Webseite der Internet Assigned Numbers Authority [IANA, 2013].

Überführung von IPv6-Multicast-Adressen auf Ethernet-Adressen

Für die Übertragung eines IPv6-Multicast-Pakets über Ethernet ist eine Überführung der Ziel Multicast-Adresse (DST) auf eine MAC-Adresse notwendig. Die ersten 16 Bits (zwei Oktette) dieser Ethernet-Multicast-Adresse sind immer gleich und lauten hexadezimal geschrieben 3333. Die weiteren 4 Oktette der MAC-Adresse entsprechen den letzten 4 Oktetten der IPv6-Multicast-Adresse [Crawford, 1998].

So ergibt sich beispielsweise aus der IPv6-Multicast-Adresse FF08::114 die MAC-Adresse 33-33-00-00-01-14.

Tritt ein Node einer Multicast-Gruppe bei, lässt die Netzwerkkarte alle Pakete mit dieser link-layer Adresse zu und leitet sie an höhere Protokollschichten weiter.

2.3.2 Kommunikation mittels Multicast

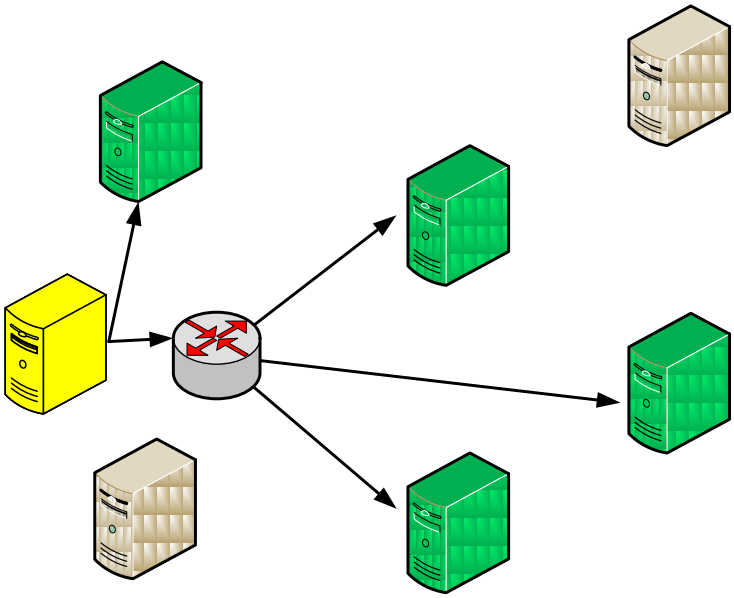

Abbildung 3: Multicast-Kommunikation

Abbildung 3 stellt eine Kommunikation über IPv6-Multicast dar. Die Mitglieder einer Multicast-Gruppe, z. B. alle NTP-Server (FF08::101), sind dabei grün dargestellt. Alle weiteren Nodes in dem Netz sind grau und der Sender ist gelb. Möchte der Sender mit mehreren Servern gleichzeitig kommunizieren, sendet er ein einziges IPv6-Paket mit der Multicast-Adresse FF08::101 als Zieladresse ins Netz. Alle NTP-Server im lokalen Netz empfangen das Paket, da sie, wie oben beschrieben, alle Pakete mit der entsprechenden MAC-Adresse entgegennehmen. Nicht-Gruppenmitglieder verarbeiten das Paket dementsprechend nicht. Da es sich bei der Multicast-Adresse mit Gültigkeitsbereich 8 um eine "organisationslokale" Multicast-Adresse handelt (siehe Tabelle 1), wird dieses Paket von den Routern im Subnetz ebenfalls empfangen und an alle Schnittstellen weitergeleitet, an denen auch Mitglieder dieser Multicast-Gruppe existieren. Die Pakete werden insgesamt so lange weitergeleitet, bis sie den angegebenen Gültigkeitsbereich erreichen, und dort verworfen. In diesem Beispiel wäre das die Grenze des Netzes der Organisation.

Für die dynamische Verwaltung der Gruppenzugehörigkeit auf den Routern ist das Multicast Listener Discovery Protokoll zuständig, das in Kapitel 2.3.3 näher beschrieben wird.

2.3.3 Multicast Listener Discovery (MLD) und MLDv2

Für die Organisation von IPv6-Multicast-Gruppen wird das Multicast Listener Discovery Protokoll verwendet. Dieses ermöglicht es, dynamisch Multicast-Gruppen zu verwalten, wobei die Verwaltung in den Routern stattfindet.

MLD verwendet für die Kommunikation ICMPv6, wobei ein MLD-Paket aus einem IPv6-Header, einem Hop-by-Hop-Header und der MLD-Nachricht besteht. Der Hop-by-Hop-Header enthält die IPv6 Router Alert Option, damit jeder Router das MLD-Paket verarbeitet, auch wenn er selbst nicht Mitglied der (Multicast-) Gruppe ist. Definiert ist das Protokoll in RFC 2710.

Es existieren drei Typen von MLD-Nachrichten,

- Multicast Listener Query (ICMPv6 Type 130)
- Multicast Listener Report (ICMPv6 Type 131)
- Multicast Listener Done (ICMPv6 Type 132)

Multicast Listener Query

Eine Multicast Listener Query Nachricht wird von IPv6-Multicast-fähigen Routern verwendet, um ein Subnetz nach Multicast-Gruppen-Mitgliedschaften zu überprüfen. Dazu existieren zwei Typen von Nachrichten:

Genereller Query: Wird periodisch an alle Hosts eines Subnetzes versendet, um jegliche Mitgliedschaften eines Hosts zu einer beliebigen Multicast-Gruppe zu erfragen.

Multicast-Adressen-Spezifischer Query: Wird verwendet, um alle Hosts eines Subnetzes, die Mitglied einer spezifischen Multicast-Gruppe sind, zu erfragen.

Erhält ein Host eine Multicast Listener Query Nachricht und ist Mitglied der Multicast-Gruppe, so wartet dieser eine zufällige Zeitspanne ab und sendet dem Router eine Multicast Listener Report Nachricht als Bestätigung zu. Der Router schreibt diese Informationen dann in seine Multicast-Forwarding-Tabelle.

Multicast Listener Report

Multicast Listener Report Nachrichten werden von Hosts verwendet, um Interesse an einer Multicast-Gruppen-Mitgliedschaft zu signalisieren oder zu bestätigen. Dies kann, wie oben beschrieben, nach dem Empfang einer Multicast Listener Query Nachricht erfolgen oder unabhängig davon, sobald der Host dieser Gruppe beitreten möchte.

Multicast Listener Done

Multicast Listener Done Nachrichten werden von den Gruppenmitgliedern versendet, um den lokalen Router darüber zu informieren, dass keine Mitglieder einer speziellen Multicast-Gruppe in einem Subnetz mehr existieren. Sobald ein Mitglied die Gruppe verlässt, sendet es diese Meldung an den Router, da es davon ausgehen muss, dass es das letzte Mitglied dieser Gruppe war. Wie viele Mitglieder eine Gruppe besitzt, wird nämlich weder bei den Mitgliedern noch bei den Routern gespeichert. Um zu kontrollieren, ob tatsächlich das letzte Mitglied die Gruppe verlassen hat, überprüft der Router dies mithilfe einer Multicast Listener Query Nachricht (spezifischer Query). Existiert noch ein Mitglied der Multicast-Gruppe, erhält er eine Antwort (Mutlicast Listener Report) und leitet weiterhin Pakete für diese Multicast-Gruppe an dieses Subnetz weiter.

MLDv2 erweitert MLD um die Möglichkeit, nur Multicast-Traffic von explizit angegebenen Quell-Adressen zu erhalten. Dadurch kann ein Host angeben, dass er z. B. der Multicast-Gruppe FF02::101 (alle NTP-Server im lokalen Netz) beitreten und somit den Traffic an diese Gruppe empfangen möchte, aber nur von dem Client mit der (Quell)-Adresse 2001:db8::10.

Für diese Erweiterung existieren eine modifizierte Version der Multicast Listener Query Nachricht, in der die Quell-Adressen angegeben werden können, und eine neue Version der Multicast Listener Report Nachricht (ICMPv6 Type 143). Die genaue Beschreibung der Änderungen ist im RFC 3810 erläutert.

2.3.4 Multicast Routing

Ist in einem Verbund von Netzen mehr als ein Multicast-Router vorhanden, ist es notwendig, dass sich diese Router gegenseitig über Multicast-Gruppen-Mitgliedschaften informieren. Nur so können Gruppenmitglieder den IPv6-Multicast-Traffic an jedem Ort des Netzes empfangen. Für den Austausch dieser Informationen wird ein Multicast-Routing-Protokoll verwendet, wie beispielsweise das Protocol Indepenent Multicast (PIM).

Multicast-Routing-Protokolle haben unter anderem folgende Ziele [Davies, 2012]:

- Den Verkehr weg von der Quelle leiten, um Schleifen zu vermeiden
- Reduzierung oder Eliminierung von Multicast-Verkehr in Subnetzen, die den Verkehr nicht benötigen
- Reduzierung von CPU- und Speicher-Auslastung auf den Routern

Unter Linux stehen dafür verschiedene Dämons bereit. Unter anderem unterstützt mrd6 diverse Multicast-Routing-Protokolle, wie z. B. PIM. Der Dienst SMCROUTE ermöglicht das Erstellen von statischen IPv6-Multicast Routen. Unter Windows ist eine Bearbeitung der Multicast-Routing Tabelle zum jetzigen Zeitpunkt nicht möglich [Davies, 2012].

2.3.5 Einsatzbeispiele

Einen besonderen Vorteil bietet Multicast bei Streaming-Diensten. Der Server sendet dabei den Stream an eine IPv6-Multicast Adresse und alle an diesem Stream interessierten Clients joinen diese Gruppe (IPv6-Multicast Adresse). Dadurch werden Last und Bandbreite auf dem Netz und auf dem Server reduziert. Denn dieser muss nur einen Stream anbieten und erst in den Routern auf dem Weg zu den Clients wird der Stream verteilt. In dem damaligen Test-Netz für IPv6-Mutlicast „M6Bone“ wurden einige Sendungen\Streams periodisch mittels Multicast angeboten [Durand, 2004].

Neighbor Discovery Protocol

Ein Beispiel für die aktive Verwendung von IPv6-Mutlicast ist das Neighbor Discovery Protocol (NDP). Dieses wird von allen IPv6-Nodes unterstützt und ist in grundlegende Abläufe einer IPv6-Kommunikation involviert. Dies betrifft unter anderem die Adressauflösung, also die Ermittlung einer entsprechende MAC-Adresse zu einer IPv6-Adresse. Dieses Verfahren löst somit das aus IPv4 bekannte Address Resolution Protocol (ARP) ab [Narten, et al., 2007].

Für die Ermittlung der MAC-Adresse eines Empfängers sendet ein Node eine Neighbor-Solocitation-Nachricht an die Solicited-Node IPv6-Multicast-Adresse des Empfängers. Bei dieser IPv6-Multicast-Adresse handelt es sich um eine Multicast-Gruppe, der jeder Node mit jeder seine Unicast- und Multicast-Adressen beitreten muss. Gebildet wird die Solicited-Node IPv6-Multicast-Adresse aus einem festen Präfix (FF02:0:0:0:0:1:FF00::/104) und aus den letzten 24 Bits der IPv6-Adresse. Dadurch ist in der Regel nur ein Node in dieser Gruppe. Die Nodes in dieser Gruppe antworten auf die empfangene Neighbor-Solocitation-Nachricht und übermitteln dem anfragenden Node ihre MAC-Adresse. [Deering, et al., 2006].

Konkretes Beispiel:
Soll die IPv6-Adresse 2001:db8::1234:5678 aufgelöst werden, so wird eine Neighbor-Solocitation-Nachricht an die Multicast-Adresse ff02::1:ff34:5678 versendet.

2.4 Domain Name System (DNS)

Domain Name System ist ein Dienst für die Zuordnung von Hostnamen zu IP-Adressen und umgekehrt. DNS wurde erstmals 1983 in RFC 882 und RFC 883 beschrieben. Beide RFCs wurden inzwischen von RFC 1034 und RFC 1035 abgelöst und durch zahlreiche weitere Standards ergänzt. Die Erweiterung für IPv6 ist in RFC 3596 beschrieben (siehe auch Kapitel 2.4.3).

2.4.1 Client-Server Kommunikation

Für die Namensauflösung mittels DNS sendet der Client, auch Resolver genannt, eine Anfrage (DNS-Query) an den DNS-Server per UDP Port 53. Abbildung 4 zeigt einen typischen Ablauf einer Namensauflösung. In der Regel ruft ein Programm (z. B. ein Webbrowser) eine Funktion des Resolvers auf und übergibt den gesuchten Hostnamen. Der Resolver leitet die Anfrage, falls diese nicht schon in seinem Cache existiert, an den DNS-Server und erhält als Antwort (DNS-Response) die entsprechende IP-Adresse oder eine Fehlermeldung. Anschließend liefert der Resolver die Antwort an die Anwendung.

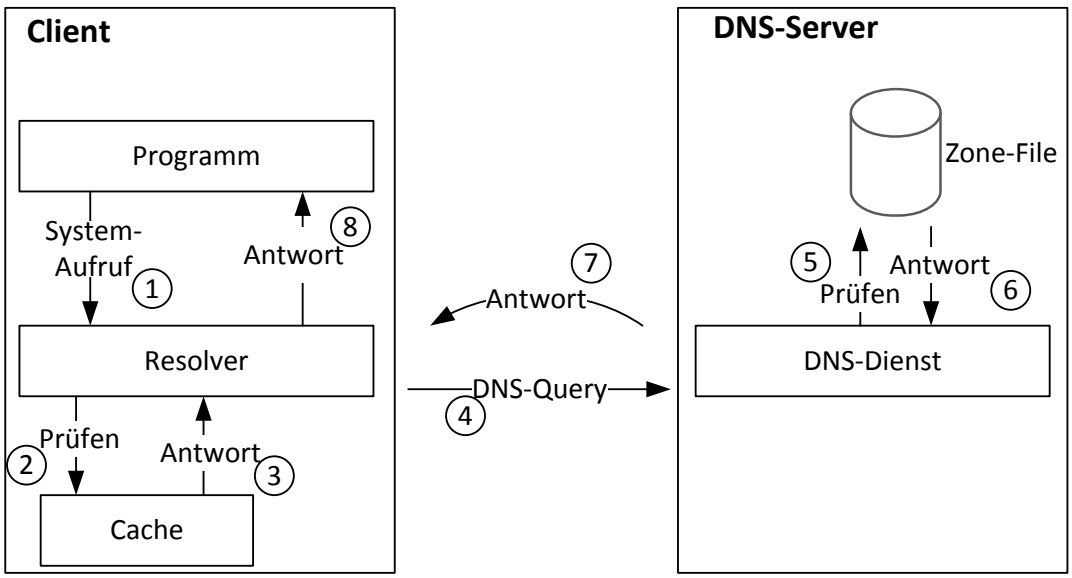

Abbildung 4: Namensauflösung mit DNS

Das genaue Verhalten des Resolvers unterscheidet sich je nach Betriebssystem und kann zudem vom Benutzer verändert werden. Besonders relevant werden diese Eigenschaften für die spätere Untersuchung der Verfügbarkeit und Response-Time des DNS-Dienstes. Dazu wird in Kapitel 3.4 das Verhalten diverser Clients genauer analysiert.

2.4.2 Domain Name System Security Extensions (DNSSEC)

Die Domain Name System Security Extensions sind eine Erweiterung des DNS und beschreiben Sicherheitsmechanismen zur Gewährleistung der Authentizität und Integrität der Daten.

Die erste Definition von Domain Name System Security Extensions erschien 1997 im RFC 2065. Diese Version funktionierte jedoch nur in kleinen Netzen, bei großen überforderte das Protokoll die Domain Name Server. Die aktuellen Implementierungen beruhen auf den überarbeiteten Standards RFC 4033, 4034 und 4035.

DNSSEC basiert auf einem asymmetrischen Verschlüsselungsverfahren. Der DNS-Server signiert dabei alle Records mit seinem privaten Schlüssel. Im für DNSSEC neu definierten RRSIG-Record wird die Signatur jedes Records anschließend abgelegt. Der zu dem privaten Schlüssel zugehörige öffentliche Schlüssel wird in dem ebenfalls neuen DNSKEY-Record gespeichert.

Bei einem DNS-Query mit DNSSEC (wie in Abbildung 5 dargestellt) wird neben dem eigentlich angefragten Record zudem der entsprechende RRSIG-Record mit übertragen (Schritt 3). Die Resolver können nun die Signatur mit dem öffentlichen Schlüssel (DNSKEY) validieren und damit die Authentizität und die Integrität der Nachricht überprüfen (Schritt 3.1 & 3.2). Dazu muss dem DNSKEY

aber vertraut werden. Wird dem öffentlichen Schlüssel (DNSKEY) des DNS-Servers (h-brs.de) nicht grundsätzlich vertraut, muss der Resolver diesen anhand einer Vertrauenskette (Chain of Trust) bei den übergeordneten DNS-Servern überprüfen (Schritt 3.3 bis 3.6). Dem öffentlichen Schlüssel der Root-Server wird von den Resolvern in der Regel vertraut. Anschließend signalisiert der Resolver dem Client, dass die Überprüfung erfolgreich war, indem in Schritt 4 im DNS-Responsepaket das AD-Flag (Authenticated Data) gesetzt wird. Die Clients (Stub-Resolver) können gewöhnlich keine eigene Überprüfung der öffentlichen Schlüssel durchführen und haben nur die Möglichkeit, dem AD-Flag des DNS-Servers blind zu vertrauen.

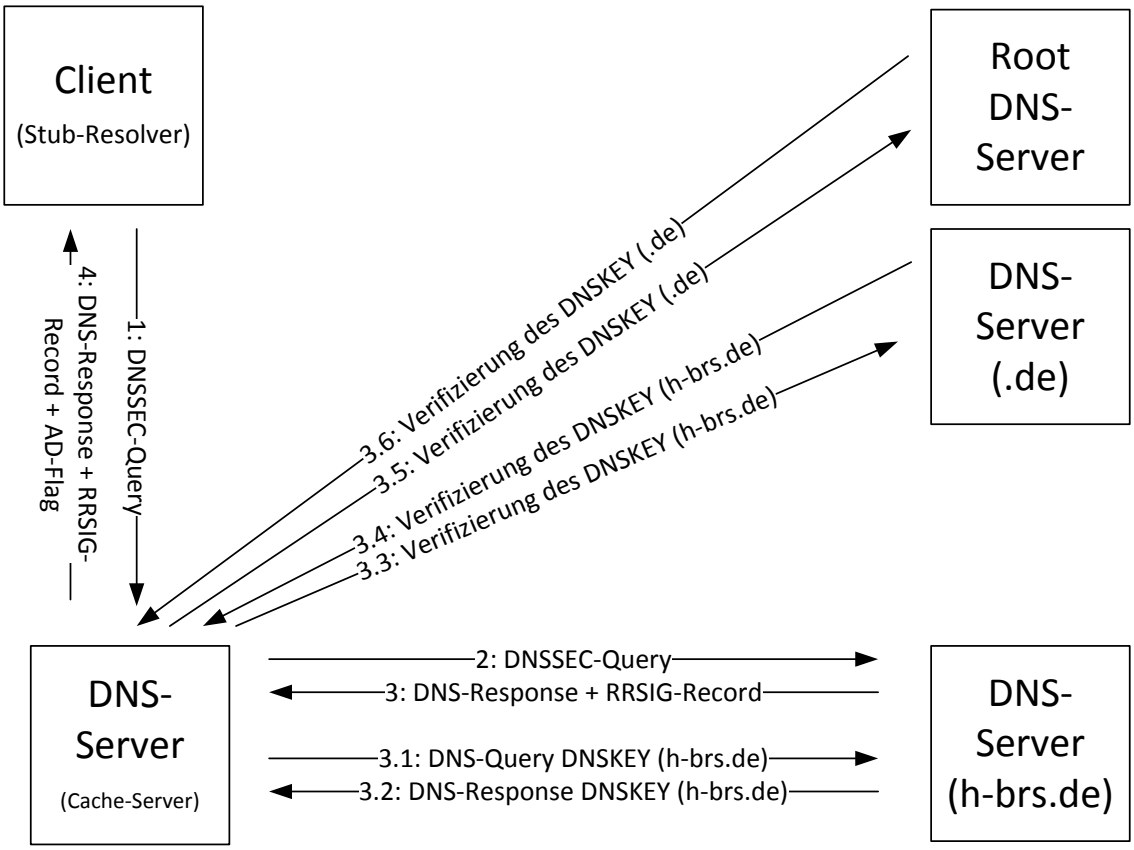

Abbildung 5: Kommunikationsablauf bei DNSSEC

Für die Verwendung von DNSSEC ist die DNS Erweiterung EDNS (RFC 2671) erforderlich, da zusätzliche Parameter übertragen werden müssen, die keinen Platz mehr im DNS-Header fanden. Die EDNS-Erweiterung hebt zudem die Beschränkung von DNS-Nachrichten über UDP von 512Byte auf. Die Beschränkung im ursprünglichen DNS Standard beruht auf der Gegebenheit, dass damals nur UDP-Pakete mit bis zu 512Byte von allen Nodes wieder zusammengesetzt werden konnten (Reassembly-Buffer-Size) [Mockapetris, 1987] [Vixie, 1999].

Durch die signierten Records und die längeren AAAA-Records wird die Grenze von 512Byte schnell überschritten, wodurch die DNS-Erweiterung (EDNS) erforderlich wurde. Dadurch sind nun auch Übertragungen von bis zu 4096Byte möglich [Liu, et al., 2006].

2.4.3 DNS-Erweiterung für IPv6

Die notwendigen Erweiterungen für die Namensauflösung mittels DNS in IPv6 beschreibt der RFC 3596. Grundlegendes an DNS selbst wurde nicht geändert, somit auch nicht an der Namensauflösung für IPv4. Nur Query-Types, die bisher zusätzliche Type-A Einträge mitgeliefert haben, wie beispielsweise MX-Query-Types, müssen nun beide Typen (A und AAAA) unterstützen.

Der RFC definiert lediglich für die Erweiterung auf IPv6 einen neuen Record-Type – AAAA (auch “Quad-A” genannt) – für die Namensauflösung von FQDN in eine einzelne IPv6-Adresse. Dieser Record-Type ist vergleichbar mit dem A-Record bei IPv4, verwendet aber als Record-Type-Wert die 28.

Ein Beispiel für einen AAAA-Record in einer typischen DNS-Datenbank:

Name	Class	Record-Type	Address
netlab.inf.h-brs.de	IN	AAAA	2001:638:408:200::5

Die Resolver der neueren Betriebssysteme, wie Windows 7 oder Ubuntu 12.04, erfragen beim DNS-Server automatisch beide Records (A-Record und AAAA-Record).

Für eine manuelle Abfrage eines solchen Eintrages, kann bei DNS-Tools (nslookup oder dig) explizit nach einem AAAA-Record oder aber nach allen Records gefragt werden. Abbildung 6 zeigt eine explizite Abfrage eines AAAA-Record und dessen Antwort.

Abbildung 6: Abfrage AAAA-Record

Für die Auflösung von Reverse-Querys, die mithilfe einer IPv6-Adresse den Hostname ermitteln, wurde die Domain IP6.ARPA definiert. Jede IPv6-Adresse stellt hierbei einen Namen in der Domain IP6.ARPA dar. Dazu wird die IPv6-Adresse rückwärts dargestellt, wobei jede Ziffer mit einem Punkt getrennt wird und am Ende noch das Suffix IP6.ARPA hinzugefügt wird.

Für die IPv6-Adresse
2001:638:408:201:4:3:2:1
bzw. komplett ausgeschrieben
2001:0638:0408:0201:0004:0003:0002:0001
bzw. rückwärts notiert und mit Punkten getrennt
1.0.0.0.2.0.0.0.3.0.0.0.4.0.0.0.1.0.2.0.8.0.4.0.8.3.6.0.1.0.0.2
ergibt sich der Reverse-Pointer
1.0.0.0.2.0.0.0.3.0.0.0.4.0.0.0.1.0.2.0.8.0.4.0.8.3.6.0.1.0.0.2.IP6.ARPA

Möchte der Client nun den Hostnamen zu der IPv6-Adresse 2001:638:408:201:4:3:2:1 ermitteln, sendet dieser einen DNS-Query mit dem Record-Type PTR und der IPv6-Adresse an den DNS-Server. Dieser kann anhand des Domain-Baums der Domain IP6.ARPA den passenden Hostnamen ermitteln.

3 Testumgebung

Die späteren Tests finden in einer einheitlichen Umgebung statt. Diese repräsentiert eine mögliche Unternehmensinfrastruktur. Notwendig ist dies, um die Tests zu vereinheitlichen und einzugrenzen. Die folgende Abbildung stellt die Infrastruktur dar:

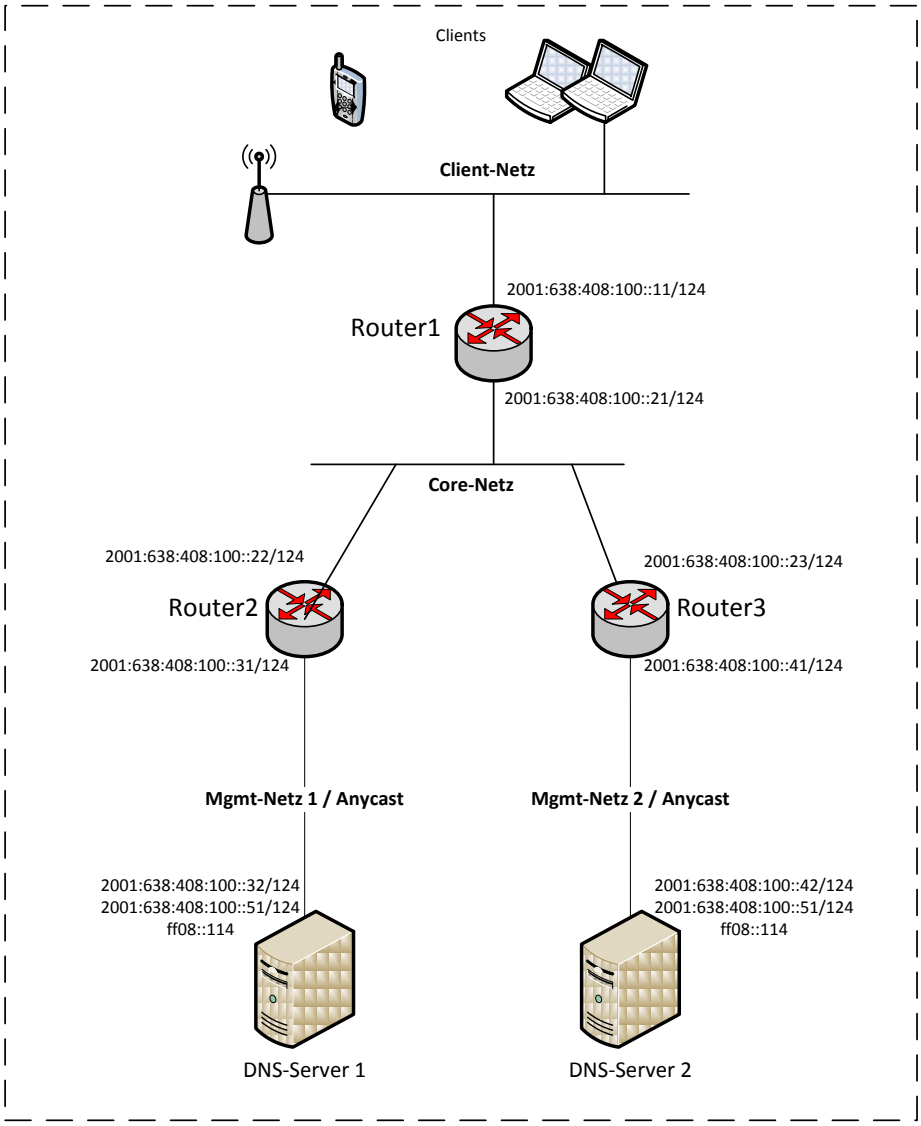

Abbildung 7: Aufbau der Testinfrastruktur

Der IT-Grundschutz-Katalog fordert für den Betrieb von Nameservern den Einsatz von mindestens zwei Servern. Sie sollen über getrennte Anbindungen verfügen und sich physikalisch voneinander getrennt befinden. Dadurch ist eine gewisse Ausfallsicherheit gewährleistet [BSI, 2012].

Für die Testumgebung werden die zwei DNS-Server (DNS-Server 1 und DNS-Server 2) bereitgestellt, welche für die Domain test-dns.netlab.inf.h-brs.de zuständig sind. Zudem sind die DNS-Server über die Router 1 & 2 angebunden und befinden sich in unterschiedlichen Subnetzen (Mgmt-Netz 1 und Mgmt-Netz 2).

Bei der verwendeten Domain handelt es sich nur um eine interne Unternehmensdomain. Sie sollte laut IT-Grundschutz nur von intern, also von dem Unternehmensnetz erreichbar sein und nicht über das Internet. Daher erhalten die DNS-Server in der Testumgebung keine Internetverbindung und sind nur von den Clients aus dem Netz "Client-Netz" erreichbar.

Als Betriebssystem bei den Servern kommt Ubuntu 12.04 zum Einsatz. Es ist die aktuelle Version mit Langzeit-Support und wird auch nach dieser Arbeit noch offiziell unterstützt. Als DNS-Dienst wird die Bind Version 9.8.4 eingesetzt [Internet Systems Consortium, 2012].

In dem Netz "Client-Netz" befinden sich diverse Clients (Windows, Linux, OS X) für die zukünftigen Testszenarien. Für die Clients, die nur eine WLAN Schnittstelle besitzen (z. B. Handys, Tablets), steht ein WLAN Accces-Point zur Verfügung.

Die Router verwenden das Betriebssystem Ubuntu 12.04. Je nach Kommunikationsmodell und Testszenario werden unterschiedliche Routingprotokolle verwendet, die in den einzelnen Testszenarien näher beschrieben sind. Die Router, sowie Server und Clients sind mit einer IPv6-Adresse versehen.

3.1 Unicast-Konfiguration

Zunächst wird die Systemkonfiguration für die Unicast-Kommunikation beschrieben. Dazu sind Konfigurationsschritte auf den Clients, den drei Routern und den beiden DNS-Servern notwendig.

Client-Konfiguration
Die einzelnen Clients aus dem "Client-Netz" erhalten eine IPv6-Adresse (2001:638:408:100::10/124) und verwenden den Router 1 (IPv6-Adresse 2001:638:408:100::11) als Default-Gateway. Die IPv6-Adressen der DNS-Server (2001:638:408:100::32 und 2001:638:408:100::42) werden dem Client per Router-Advertisement (RDNSS-Option) oder DHCPv6 mitgeteilt. Der IPv4-Stack ist vollständig deaktiviert.

Router-Konfiguration
Die Router verwenden Ubuntu 12.04 als Betriebssystem und nur der IPv6-Stack ist aktiv. Die IPv6-Adressen der einzelnen Schnittstellen werden manuell vergeben. Damit das System als Router fungiert, ist das IPv6-Forwarding mittels folgendem Eintrag in die /etc/sysctl.conf zu aktiveren:

```
net.ipv6.conf.default.forwarding=1
```

Die Schnittstellen, die IPv6-Adresse und die statischen Routen in der Routing-Tabelle für die Router lauten wie folgt:

Router 1:
Interface 1 (eth0): 2001:638:408:100::11
Interface 2 (eth1): 2001:638:408:100::21

Ziel	Gateway	Interface
DNS1 (2001:638:408:100::30/124)	Router 2 (2001:638:408:100::22)	eth1
DNS2 (2001:638:408:100::40/124)	Router 3 (2001:638:408:100::23)	eth1

Tabelle 2: Routing-Tabelle von Router 1 (Unicast)

Router 2:
Interface 1 (eth0): 2001:638:408:100::22
Interface 2 (eth1): 2001:638:408:100::31

Ziel	Gateway	Interface
Client (2001:638:408:100::10/124)	Router 1 (2001:638:408:100::21)	eth0
DNS2 (2001:638:408:100::40/124)	Router 3 (2001:638:408:100::23)	eth0

Tabelle 3: Routing-Tabelle von Router 2 (Unicast)

Router 3:
Interface 1 (eth0): 2001:638:408:100::23
Interface 2 (eth1): 2001:638:408:100::41

Ziel	Gateway	Interface
Client (2001:638:408:100::10/124)	Router 1 (2001:638:408:100::21)	eth0
DNS1 (2001:638:408:100::30/124)	Router 2 (2001:638:408:100::22)	eth0

Tabelle 4: Routing-Tabelle von Router 3 (Unicast)

Server-Konfiguration

Die Server sind über die IPv6-Adresse 2001:638:408:100::32 bzw. 2001:638:408:100::42 zu erreichen. Für die Namensauflösung sind die beiden Server für die Domäne test-dns.netlab.inf.h-brs.de, sowie die Reverse-Domäne 2001:638:408:100::0 zuständig (siehe /etc/bind/named.conf.local).

```
zone "test-dns.netlab.inf.h-brs.de" {
      type master;
      file "/etc/bind/zones/test-dns.netlab.inf.h-brs.de.db";
};

zone "0.0.1.0.8.0.4.8.3.6.1.0.0.2.ip6.arpa" {
      type master;
      file "/etc/bind/zones/2001:638:408:100::0.db" ;
};
```

Listing 1:Bind-Konfiguration - named.conf.local

Für die späteren Tests sind die Domains mit einer Million Test-Records gefüllt, die von den Clients in allen drei Kommunikationsmodellen abgefragt werden können. Die Einträge dazu befinden sich in der Zonendatei und sehen wie folgt aus (/etc/bind/zones/test-dns.netlab.inf.h-brs.de.db):

```
$TTL 3h
@     IN    SOA   ns1.test-dns.netlab.inf.h-brs.de   admin.h-
brs.de. (
                    2013040101 ; Serial
                    21600      ; Refresh nach 6 Stunden
                    3600       ; Retry nach 1 Stunde
                    604800     ; Expire nach 1 Woche
                    86400 )    ; Minimum TTL von 1 Tag
IN    NS    ns1.test-dns.netlab.inf.h-brs.de
IN    NS    ns2.test-dns.netlab.inf.h-brs.de
                  .
                  .
                  .
      (1.000.000 weitere AAAA-Records)
                  .
                  .
                  .
ns1   IN    AAAA  2001:638:408:100::32
ns2   IN    AAAA  2001:638:408:100::42
```

Listing 2: Zonendatei

Die eigentliche DNS-Server Konfiguration für die Unicast-Szenarien befindet sich in der Datei /etc/bind/named.conf.options. Hier wird angegeben, dass der DNS-Dienst nicht per IPv4 zu erreichen ist, sondern nur über die IPv6-Adresse 2001:638:408:100::32 bzw. 2001:638:408:100::42 auf dem Standard-Port 53 für DNS. Für weitere Informationen zur IPv6-Konfiguration von Bind siehe "DNS & Bind im IPv6" [Liu, 2011].

```
options {
      directory "/var/cache/bind";
      dnssec-validation auto;
      auth-nxdomain no;       #conform to RFC 1035
      listen-on { none; };
      listen-on-v6 port 53 { <IPv6Adresse>; };
};
```

Listing 3: Bind-Konfiguration für Unicast

3.2 Anycast-Konfiguration

Gegenüber Unicast wird bei der Erstellung der Anycast-Umgebung die Konfiguration der einzelnen Komponenten angepasst. Betroffen sind insbesondere die DNS-Server und die Router, die hier ein dynamisches Routing verwenden.

Client-Konfiguration
Die Clients beziehen auch im Anycast-Szenario ihre IPv6-Adresse automatisch. Dazu soll auch die IPv6-Anycast-Adresse (2001:638:408:100::53) der beiden DNS-Server über DHCPv6 bzw. RDNSS an die Clients verteilt werden.

Da sich eine IPv6-Anycast-Adresse nicht von anderen globalen IPv6-Adressen unterscheiden lässt, ergeben sich bei der automatischen Verteilung der DNS-Server-Adresse keine Probleme. Sowohl die Verteilung der IP-Adresse per Router-Advertisements und DHCPv6, als auch die anschließende korrekte Verwendung durch die Clients ist gegeben.

Router-Konfiguration
Im Anycast-Szenario ist es sinnvoll, ein dynamisches Routing-Protokoll zu verwenden, um die Vorteile von Anycast nutzen zu können. Dazu wird der Quagga-Dienst, ein für unixartige Systeme entwickeltes Routing Softwarepaket, in der Version 0.99.22 verwendet und auf allen Routern installiert. Unten sind von allen Routern die Konfigurationen von Quagga sowie die Schnittstellen/IPv6-Adressen dargestellt.

Router 1:
Interface 1 (eth0): 2001:638:408:100::11
Interface 2 (eth1): 2001:638:408:100::21

Router 2:
Interface 1 (eth0): 2001:638:408:100::22
Interface 2 (eth1): 2001:638:408:100::31

Router 3:
Interface 1 (eth0): 2001:638:408:100::23
Interface 2 (eth1): 2001:638:408:100::41

Für das dynamische Routing wird OSPFv3 verwendet. Es verfügt gegenüber Distanz-Vektor-Protokollen (z. B. RIP) über eine schnelle Konvergenz und ist

weit verbreitet. Zudem ist es IPv6-fähig. Die notwendigen Konfigurationen werden in der Datei /etc/quagga/ospf6d.conf vorgenommen:

```
...
interface eth0
      ipv6 ospf6 cost 1
      ipv6 ospf6 hello-interval 10
      ipv6 ospf6 dead-interval 40
      ipv6 ospf6 retransmit-interval 5
      ipv6 ospf6 priority 0
      ipv6 ospf6 transmit-delay 1
      ipv6 ospf6 instance-id 0

interface eth1
      ipv6 ospf6 cost 1
      ipv6 ospf6 hello-interval 10
      ipv6 ospf6 dead-interval 40
      ipv6 ospf6 retransmit-interval 5
      ipv6 ospf6 priority 0
      ipv6 ospf6 transmit-delay 1
      ipv6 ospf6 instance-id 0
router ospf6
      router id 255.1.1.1
      interface eth0 area 0.0.0.0
      interface eth1 area 0.0.0.0
...
```

Listing 4:Quagga-Konfiguration

Alle Router erhalten die oben angegebene Konfiguration. Die Router haben somit direkten Anschluss an die Backbone-Area (0.0.0.0) und die Kosten aller Interfaces sind gleich (eins). Einzig die Router-ID ist auf jedem Router eindeutig. Alle zehn Sekunden werden Hello-Pakete versendet, um zu überprüfen, ob die Verbindung zum nächsten Router noch existiert. Das Dead-Interval, also die Zeit, nach der die Verbindung als nicht existent betrachtet wird, beträgt 40 Sekunden. Diese Werte entsprechen dem Standard von OSPFv3. Die Reduzierung dieser Werte kann eine höhere Konvergenz des Netzes bei Ausfällen bewirken. Andererseits erhöhen sich dadurch auch der Netzverkehr und die Wahrscheinlichkeit von fälschlicherweise abgebauten Nachbarschaften [Coltun, et al., 2008].

Nach Konfiguration und Neustart des Quagga-Dienstes werden auf den Interfaces eth0 und eth1 die Routinginformationen ausgetauscht und die Netztopologie ermittelt.

Server-Konfiguration

Den DNS-Servern wird die IPv6-Anycast-Adresse 2001:638:408:100::53 zugeteilt. Sie muss im Betriebssystem anschließend als Anycast-Adresse festgelegt werden. Dazu wird sie unter Windows Server 2008 R2 bzw. Ubuntu Server 12.04 mit folgenden Befehlen hinzugefügt:

Windows

```
netsh interface ipv6 add address interface=<index> address=<IPv6-Adresse>
type=anycast
```

Linux

```
ip addr add <IPv6-Adresse> anycast <IPv6-Adresse> dev <interface>
```

Damit die Router die DNS-Pakete an die Anycast-Adresse zustellen können, müssen die Server logischerweise verfügbar sein. Allerdings verfügt Router 1 über keine Möglichkeit, vor dem Senden des Paktes den Ausfall eines Servers festzustellen. Das Netz (2001:638:408:100::50/124), in dem sich die Anycast-Adresse befindet, wird von den Routern 2 und 3 weiterhin propagiert. Da die beiden Router direkt an dieses Netz angebunden sind, können sie es auch immer noch erreichen. Folglich wird Router 1 immer versuchen, die Pakete über Router 2 an den DNS-Server zu senden. Unabhängig, ob der DNS-Server 1 verfügbar ist oder nicht.

Es ist daher sinnvoll, eine Verknüpfung zwischen der Route und der Verfügbarkeit der DNS-Server herzustellen. Die Route zum Server soll nur propagiert werden, wenn der DNS-Server auch zur Verarbeitung der Pakete in der Lage, also verfügbar, ist. Daher wird ebenfalls auf den DNS-Servern ein Routing-Dienst (Quagga) installiert. Für den DNS-Dienst bzw. die Anycast-Adresse wird eine eigene Route propagiert. Die Verfügbarkeit des DNS-Dienstes wird daher durch die Bekanntgabe der Route signalisiert. Fällt beispielweise der DNS-Server aus, würde auch die Route durch den ebenfalls auf dem DNS-Server installierten Routing-Dienst nicht mehr propagiert.

Auf den DNS-Servern wird auf dem Loopback-Interface die IPv6-Anycast-Adresse gebunden. Der DNS-Dienst verwendet die Anycast-Adresse als Listening-Adresse. Mithilfe von Quagga wird die Route zu der Anycast-Adresse an die anderen Router im Netz verteilt. Anhand der Routing-Tabelle von Router 2 ist ersichtlich, dass die DNS-Pakete an die Anycast-Adresse über den DNS-Server geroutet werden (siehe Next-Hop).

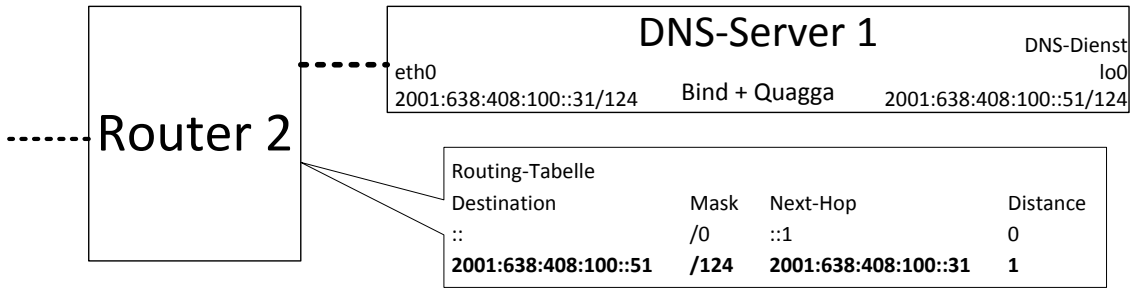

Abbildung 8: Kopplung von DNS-Dienst und Route bei Anycast

Aus Sicht des Routers 1 ergibt sich somit nachfolgende Routing-Tabelle (Tabelle 5). Ist beispielsweise DNS-Server 1 nicht mehr verfügbar, wird die Route für die Anycast-Adresse von dem DNS-Server nicht mehr propagiert. Dadurch entfällt der vorletzte Eintrag in der Routing-Tabelle und Router 1 leitet die Pakete an die Anycast-Adresse automatisch an Router 3 weiter (letzter Eintrag).

Ziel	Gateway	If
Client-Netz (2001:638:408:100::10/124)	::1	eth0
Core-Netz (2001:638:408:100::20/124)	::1	eth1
Mgmt-Netz1 (2001:638:408:100::30/124)	Router2(fe80::250:56ff:fea7:6369)	eth1
Mgmt-Netz2 (2001:638:408:100::40/124)	Router3(fe80::250:56ff:fea7:173c)	eth1
Anycast-DNS(2001:638:408:100::51/128)	Router2(fe80::250:56ff:fea7:6369)	eth1
Anycast-DNS(2001:638:408:100::51/128)	Router3(fe80::250:56ff:fea7:173c)	eth1

Tabelle 5: Routing-Tabelle von Router 1 bei Anycast

Der DNS-Dienst (Bind) soll anschließend auf der IPv6-Anycast-Adresse lauschen und darüber DNS-Querys empfangen. Die Bind-Konfiguration dafür sieht wie folgt aus (/etc/bind/named.conf.options):

```
options {
      directory "/var/cache/bind";
      dnssec-validation auto;
      auth-nxdomain no;       #conform to RFC 1035
      listen-on { none; };
      listen-on-v6 port 53 { 2001:638:408:100::53; };
};
```

Es entstehen durch die Verwendung der IPv6-Anycast-Adresse keine Probleme, da sie syntaktisch nicht von einer Unicast-Adresse zu unterscheiden ist.

3.3 Multicast-Konfiguration

Die DNS-Server sind in der IPv6-Multicast-Umgebung Mitglied in einer IPv6-Multicast-Gruppe und die Clients kommunizieren für eine Namensauflösung per IPv6-Multicast mit den DNS-Servern.

Client-Konfiguration

Die IPv6-Adresse für die Clients wird wieder automatisch bezogen. In der Praxis beschränkt sich die Verteilung der DNS-Server IP-Adressen normalerweise auf IPv6-Unicast-Adressen. Die Standards von Router-Advertisement (RFC 6106) und DHCPv6 (RFC 3646) sehen aber für die Verteilung von Multicast-Adressen keine Einschränkungen vor. Ein praktischer Test bestätigt das. Die DNS-Server Adresse (FF08::114) kann daher über DHCPv6 bzw. RDNSS an die Clients zu verteilen

Router-Konfiguration

Die Router erhalten wieder wie bei Unicast und Anycast die gleichen IPv6-Adressen.

Router 1
Interface 1 (eth0): 2001:638:408:100::11
Interface 2 (eth1): 2001:638:408:100::21

Router 2
Interface 1 (eth0): 2001:638:408:100::22
Interface 2 (eth1): 2001:638:408:100::31

Router 3
Interface 1 (eth0): 2001:638:408:100::23
Interface 2 (eth1): 2001:638:408:100::41

Die Antwort von den DNS-Servern an die Clients erfolgt per Unicast. Dadurch werden folgende statische Routen konfiguriert, um die DNS-Antwort-Pakete per Unicast zurück an die Clients leiten zu können:

Ziel	Gateway	Interface
Client (2001:638:408:100::10/124)	Router 1 (2001:638:408:100::21)	eth0

Tabelle 6: Routing-Tabelle von Router 2 (Multicast)

Ziel	Gateway	Interface
Client (2001:638:408:100::10/124)	Router 1 (2001:638:408:100::21)	eth0

Tabelle 7: Routing-Tabelle von Router 3 (Multicast)

Als IPv6-Multicast Routing-Dienst wird SMCROUTE verwendet und folgende Weiterleitungsregeln definiert, damit die Pakete an die Multicast-Adresse FF08::114 an die beiden DNS-Server geleitet werden:

Router 1:
Alle Multicast-Pakete an die Adresse FF08::114 werden von Router 1 an die Schnittstelle eth1 weitergeleitet.

Befehl: `smcroute -a eth0 2001:638:408:100::20 ff08::114 eth1`

Router 2:
Alle Multicast-Pakete an die Adresse FF08::114 werden von Router 2 an die Schnittstelle eth1 weitergeleitet.

Befehl: `smcroute -a eth0 2001:638:408:100::21 ff08::114 eth1`

Router 3:
Alle Multicast-Pakete an die Adresse FF08::114 werden von Router 3 an die Schnittstelle eth1 weitergeleitet.

Befehl: `smcroute -a eth0 2001:638:408:100::21 ff08::114 eth1`

Jedes IPv6-Multicast-Paket besitzt, falls nicht anders angegeben, ein Hop-Limit von eins. Dieser Wert ist auch unabhängig von dem Scope-Wert der verwendeten IPv6-Multicast-Adresse [Gilligan, et al., 2003].

Da die Clients dieses Limit für einen DNS-Query auf dem Standardwert belassen, ist es auf jedem Router notwendig, dieses Limit zu erhöhen, da das Paket ansonsten verworfen wird. Dazu wird auf jedem Router mittels IPv6Table das Hop-Limit vor dem Routen des Paketes um eins erhöht. Dies ist mit folgendem Befehl möglich:

```
ip6tables -t mangle -A PREROUTING -d FF08::114 -j HL --hl-set 2
```

Server-Konfiguration
Die DNS-Server sollen Mitglied in der IPv6-Multicast-Adresse FF08::114 werden und darüber DNS-Anfragen entgegennehmen. In der Bind-Konfigurations-Datei wird daher versucht, folgende Einstellung zu verwenden:

```
options {
      directory "/var/cache/bind";
      dnssec-validation auto;
      auth-nxdomain no;       #conform to RFC 1035
      listen-on { none; };
      listen-on-v6 port 53 { ff08::114; };
};
```

Listing 5: Multicast-Konfiguration von Bind

Die anschließende Überprüfung der Zugehörigkeit zu einer IPv6-Multicast-Gruppe ergibt aber, dass der Server nicht der Gruppe FF08::114 beitritt.

Daher wird ein manuelles Joinen dieser Gruppe mithilfe des Routing-Dienstes und des Befehls `smcroute -j ff08::114 eth0` durchgeführt. Der anschließende Test ergibt, dass nun z. B. ICMPv6-Requests an die IPv6-Multicast-Adresse des Servers geschickt werden können. Der Server kann aber weiterhin keine DNS-Requests empfangen und verarbeiten, da die Pakete nicht an die entsprechende Anwendung weitergeleitet werden. Um diese Einschränkung zu umgehen, wird ein Multicast-Proxy mithilfe des Python-Scripts auf dem Server integriert. Das Script wurde im Rahmen eines IPv6-Multicast-Vortrages auf dem Heise IPv6-Kongresses 2012 vorgestellt [Strotmann, 2012].

Nach dem Starten des Proxys joint dieser die IPv6-Multicast-Gruppe FF08::114. Alle an diese Multicast-Gruppe gesendeten Pakete mit dem Zielport 53 (De-

fault-Port für DNS) nimmt der Proxy nun entgegen und sendet sie weiter an eine Unicast-Adresse. In diesem Fall werden die Loopback-Adresse des DNS-Servers und der Port 5353 verwendet.

Durch den vorgeschalteten Proxy muss nun die Bind-Konfiguration entsprechend angepasst werden. Der DNS-Dienst muss nun auf die Loopback-Adresse hören und der Port vom Standardport abweichen. Dies ist notwendig, da schon der Proxy auf Port 53 für alle IPv6-Adressen hört und kein zweiter Dienst diesen Port belegen kann. Die entsprechende Konfigurationsdatei sieht damit folgendermaßen aus:

```
options {
      directory "/var/cache/bind";
      dnssec-validation auto;
      auth-nxdomain no;       #conform to RFC 1035
      listen-on { none; };
      listen-on-v6 port 5353 { ::1; };
};
```

Listing 6: Endgültige Multicast-Konfiguration von Bind

Der darauffolgende Test eines DNS-Requests an die Multicast-Adresse FF08::114 über einen Router hinweg ist nun erfolgreich. Der entsprechende Kommunikationsablauf ist aus Abbildung 9 ersichtlich.

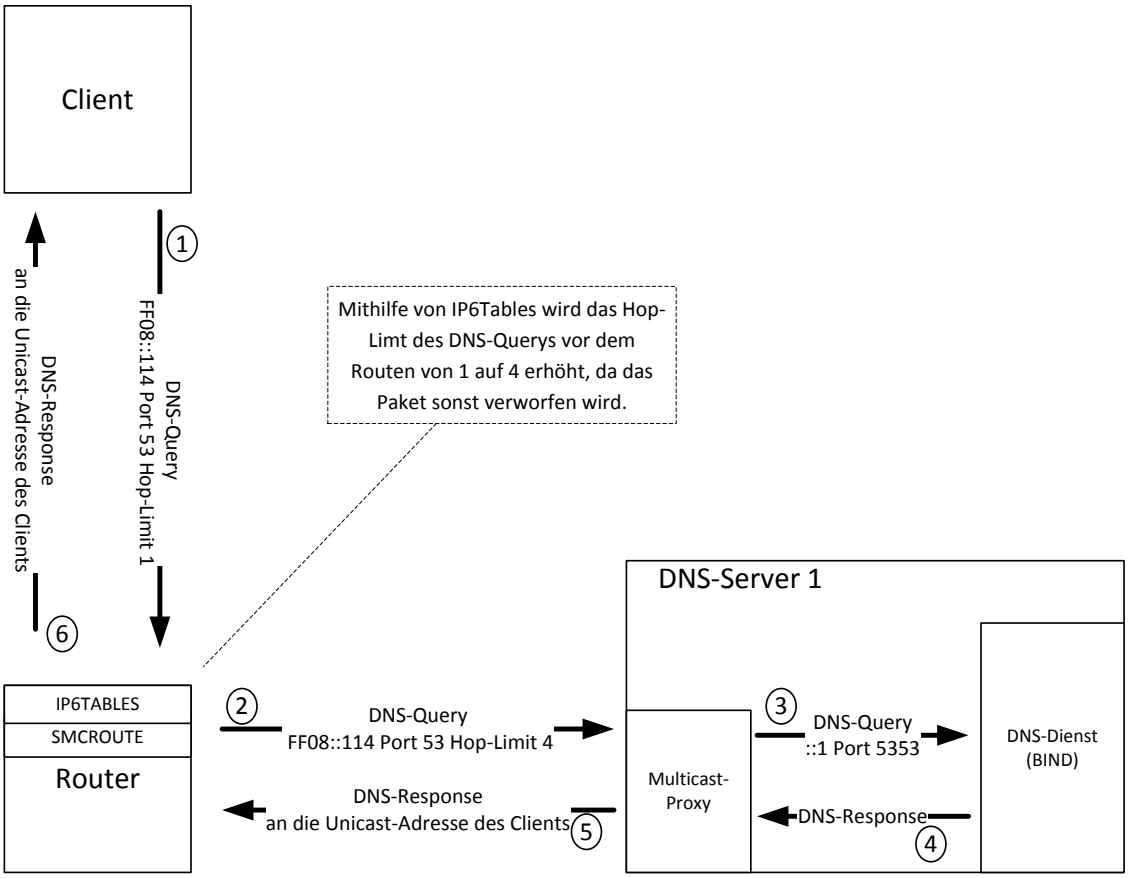

Abbildung 9: DNS-Query über Multicast

3.4 Clientsysteme

Da die Resolver der einzelnen Betriebssysteme unterschiedlich agieren und deren Verhalten Auswirkungen auf die Testergebnisse hat, werden in diesem Abschnitt die Architektur und die Eigenschaften der Resolver beschrieben. Untersucht werden die Resolver der gängigsten Desktop- und Mobile-Betriebssysteme. Die große Auswahl der Clientsysteme stellt sicher, dass die mögliche DNS-Kommunikation mittels IPv6-Anycast und IPv6-Multicast in der Praxis von ausreichend vielen Clients unterstützt wird.

Windows XP, Windows 7 und Windows 8

In Windows werden die DNS-Server-Adressen auf dem Client pro Interface gesetzt. Eine Namensauflösung läuft wie folgt ab [Microsoft Corporation, 2012]:

1. Zunächst wird das bevorzugte Interface ausgewählt (vom Benutzer konfigurierbar) und dem dort eingetragenen ersten DNS-Server ein DNS-Query gesendet. Der Timeout des ersten Querys beträgt eine Sekunde.
2. Erhält der Resolver keine Antwort, wird im nächsten Schritt für jedes Interface eine Anfrage an den nächsten DNS-Server in seiner Liste verschickt und bis zu einer Sekunde auf eine Antwort gewartet.
3. Ist immer noch keine Antwort eingetroffen, wird ein DNS-Query an alle DNS-Server über alle Interfaces verschickt und bis zu zwei Sekunden gewartet.
4. Hat der Resolver immer noch keine Antwort erhalten, sendet er erneut allen DNS-Servern-Adressen ein DNS-Query und wartet bis zu vier Sekunden.
5. Falls der Resolver wiederum keine Antwort erhält, sendet er abschließend wieder an alle DNS-Server eine Anfrage und wartet bis zu vier (bei Windows XP sieben) Sekunden.

Ist nach den vier Sekunden keine Antwort eingetroffen, bricht der Resolver die Namensauflösung ab und gibt einen Negativeintrag zurück. Maximal ergibt sich somit eine Zeit von 12 Sekunden (1+1+2+4+4=12).

Beim Einsatz von zwei DNS-Servern und einem Client mit nur einer Netzschnittstelle ergibt sich damit folgender Ablauf:

1. DNS-Server 1 (Eine Sekunde Timeout)
2. DNS-Server 2 (Eine Sekunde Timeout)
3. DNS-Server 1 (Zwei Sekunden Timeout)
4. Beide DNS-Server (Vier Sekunden Timeout)
5. Beide DNS-Server (Vier bzw. sieben Sekunden Timeout)

Eine Anpassung der Query-Timeouts kann über die Windows-Registry erfolgen. Dazu muss der Eintrag `DNSQueryTimeouts` entsprechend geändert werden.

Windows Phone 8

Dem Windows Phone kann für IPv6 nur per DHCPv6 die Adressen der DNS-Server übermittelt werden. Die Namensauflösung findet bei dem Test mit zwei eingetragenen DNS-Servern nach folgender Prozedur statt:

1. Primärer DNS-Server wird angefragt mit einem Timeout von bis zu drei Sekunden
2. Anfrage an den sekundären DNS-Server. Timeout bis zu acht Sekunden

Diese beiden Schritte werden zweimal durchlaufen. Damit ergibt sich eine maximale Wartezeit von 22 Sekunden (3+8+3+8=22).

Ubuntu 10.04, Ubuntu 11.10 und Ubuntu 12.04

Die Nameserver werden beim Client zentral in der Datei resolv.conf eingetragen und werden nach ihrer Priorität bei einer Namensauflösung verwendet (oberster Eintrag zuerst). Maximal können bis zu drei Nameserver verwendet werden. Bei einer Konfiguration mittels Network-Manager verwenden die neueren Versionen von Ubuntu zudem einen internen DNS-Server, der vom System-Resolver über das Loopback-Interface angesprochen wird. Für die Tests wird eine statische Konfiguration verwendet, um mögliche Fehlerquellen zu minimieren. Daher kommt der interne DNS-Server nicht zum Einsatz und es wird nur der Algorithmus des System-Resolvers betrachtet.

1. Der Resolver versucht bei einer Namensauflösung den ersten Server in der Liste (resolv.conf) zu erreichen.
2. Erhält der Resolver nach fünf Sekunde keine Antwort, wird der nächste Nameserver angefragt, bis alle Server in der Liste abgearbeitet sind.
3. Ist dann immer noch keine Antwort vorhanden, wartet der Resolver fünf Sekunden und beginnt die Prozedur wieder mit dem ersten Server. Insgesamt wird die Prozedur zweimal durchlaufen, bevor der Resolver dann endgültig eine Fehlermeldung an die Anwendung zurückgibt.

Bei zwei eingetragenen DNS-Servern ergibt sich somit ein maximales Timeout von 20 Sekunden (5+5+5+5=20).

Eine Anpassung der Wiederholungen und Timeout kann mittels Einträgen in der Datei /etc/resolv.conf vorgenommen werden.

Android 4.2 und Mac OS X 10.8

Die Prozedur der Resolver von Android und Mac OS X entspricht dem von Ubuntu.

iOS 6.1

Die Nameserver-IPv6-Adressen können iOS per DHCPv6 oder Router-Advertisements (RDNSS-Option) mitgeteilt werden.

1. Der erste DNS-Server wird vom Resolver angefragt.
2. Erhält der Resolver innerhalb einer Sekunde keine Antwort, wird erneut dem ersten Server eine Anfrage gesendet. Der Resolver wartet bis zu drei Sekunden auf eine Antwort.
3. Anschließend wird der sekundäre DNS-Server anfragt. Folglich wird nach insgesamt vier Sekunden Wartezeit versucht, den sekundären Server zu erreichen.
4. Abschließend wartet der Resolver bis zu drei weitere Sekunden, bevor er einen Negativeintrag zurückliefert.

Insgesamt ergibt das eine Wartezeit von maximal sieben Sekunden (1+3+3=7).

Auf allen Systemen ist es möglich, mindestens einen alternativen DNS-Server einzutragen. Antwortet der primäre DNS-Server nicht schnell genug, fragen alle Resolver automatisch weitere Server an. Die Zeit variiert je nach Resolver. Einige Resolver fahren eine "aggressivere" Strategie als die anderen und senden schon nach einer Sekunde eine weitere DNS-Anfrage. Das führt ggf. zu einer schnelleren Namensauflösung, aber auch gleichzeitig zu mehr Datenverkehr, was besonders bei den mobilen Endgeräten relevant ist. Die maximale Zeit, bis der der Resolver einen Negativeintrag zurückliefert, schwankt je nach System stark.

4 Kriterien und Testszenarien

Anhand der Kriterien werden die einzelnen Testszenarien definiert und erstellt. Die Kriterien wurden zuvor durch Literaturrecherche ermittelt und mithilfe einer Unternehmensbefragung validiert. Die Testszenarien zeigen die Eigenschaften der verschiedenen Kommunikationsmodelle auf und geben wieder, ob die zu untersuchenden Kriterien von dem jeweiligen Kommunikationsmodell unterstützt werden. Nachfolgend eine Auflistung aller untersuchten Kriterien, die in den kommenden Abschnitten zusammen mit den Testszenarien beschrieben werden:

- Systemanforderungen
- Verfügbarkeit
- Response-Time
- Aufwand bei Änderungen
- Netz-Last
- Netz-Management
- Sicherheit
- Aktualität der Antworten

Die Testergebnisse dazu folgen in Kapitel 5. Um eine breite Auswahl an unterstützten Systemen sicherzustellen, kommen dabei die folgenden Client-Betriebssysteme zum Einsatz: Ubuntu 10.04, 11.10 und 12.04, Windows XP, 7 und 8, iOS 6.1, OS X 10.8 sowie Android 4.2.

4.1 Systemanforderungen

Um überhaupt eine Kommunikation mittels IPv6-Anycast oder -Multicast zu ermöglichen, müssen die einzelnen Netzkomponenten (Client, Router, Server) bestimmte Eigenschaften erfüllen, wie z. B. das Joinen von Multicast-Gruppen. Diese Voraussetzungen werden mithilfe der nachfolgenden Szenarien überprüft.

Szenario 1.a (Multicastanforderungen):

Im ersten Szenario wird analysiert, ob diverse Netzkomponenten eine DNS-Kommunikation (DNS-Query & DNS-Reply) über IPv6-Multicast vollständig und korrekt durchführen können. Hierfür werden zuerst die entsprechenden Anforderungen für eine solche Kommunikation aufgestellt.

Client:
Damit ein Client mit einem DNS-Server kommunizieren kann, benötigt er die IP-Adresse des Servers. Die Verteilung der Adresse kann in IPv6 über drei verschiedene Wege geschehen. Diese werden inzwischen von den meisten Clients vollständig unterstützt und lauten wie folgt[1]:

- Manuelle Eintragung der Adresse auf dem Client
- Verteilung der Adresse mittels Router-Advertisements (SLAAC)
- Verteilung der Adresse per DHCPv6

Ob bei den automatischen Verfahren auch eine Verteilung von IPv6-Multicast-Adressen möglich ist, muss geklärt werden. Hierfür verteilt ein DHCPv6-Server bzw. ein Router eine IPv6-Multicast-Adresse als DNS-Server-Adresse. Anschließend wird auf den Clients geprüft, ob diese Multicast-Adresse vom Resolver verwendet wird.

Server:
Für den Server existieren vier Anforderungen für die Bereitstellung des DNS-Dienstes über IPv6-Multicast.

Anforderung 1:
Der Server muss einer IPv6-Multicast-Gruppe beitreten (joinen) können. Für das Joinen einer Gruppe stehen verschiedene Tools oder Skripts/Befehle zur Verfügung. In den Tests werden die IPv6-Mulitcast Adressen FF02::114 für linklokale Tests und die Adresse FF08::114 für subnetzübergreifende Tests verwendet. Diese Adressen sind von der IANA für private Versuche reserviert und für jeden frei verwendbar [IANA, 2013]. Ob der Server der gewünschten Gruppe beigetreten ist, kann mithilfe des Befehls `ip -6 maddr` bei Linux-Systemen und `netsh interface ipv6 show joins` bei Windows überprüft werden.

Anforderung 2:
Ist der Server Mitglied einer IPv6-Multicast Gruppe (in diesem Fall der FF02::114), sollte er von anderen Nodes aus dem selben Subnetz zu erreichen sein. Um dies zu testen, sendet ein Node einen Echo-Request an den DNS-Server 1 mit der Multicast-Adresse FF02::114. Erhält er ein Echo-Reply, ist der Test erfolgreich.

Anforderung 3:
Damit der Server DNS-Requests, die an eine IPv6-Multicast-Adresse gesendet werden, verarbeiten kann, muss die DNS-Anwendung (z. B. Bind) diese entgegen nehmen können. Dazu muss es in der Anwendung möglich sein, eine IPv6-Multicast-Adresse als Listening-Adresse einzugeben, wodurch der Server dann diese Multicast-Gruppe joinen sollte.

[1] Für weitere Informationen bzgl. SLAAC und DHCPv6, sowie die genaue Auflistung der unterstützten Client-Funktionen wird auf den Heise iX-Artikel "IPv6-Autokonfiguration für Clients" verwiesen.

Anforderung 4:
Laut RFC 2181 "Clarifications to the DNS Specification" erwarten einige Clients, dass die Source-Adresse des Reply-Paketes die gleiche ist, wie die Destination-Adresse des DNS-Request. Eine Multicast-Adresse darf aber laut RFC 4291 nicht als Source-Adresse verwendet werden, wodurch der Server gezwungen ist, eine andere IPv6-Adresse für die Source-Adresse von DNS-Reply Nachrichten zu verwenden. In dem Test wird somit überprüft, welche Source-Adresse der Server wählt und ob dies zu Problemen bei der Client-Server Kommunikation führt.

Router:
Damit eine Multicast-Kommunikation vom Client über einen Router hin zum Server funktioniert, müssen die Router in der Lage, sein IPv6-Multicast-Pakete weiterzuleiten. Die Routing-Informationen hierzu werden nicht in der Standard Routing-Tabelle gespeichert, sondern in einer eigenen Mulitcast Routing-Tabelle abgelegt.

Somit ist es notwendig, dass das Multicast-Forwarding auf dem Router (bzw. im OS des Routers) aktiviert werden kann. Zudem muss die Multicast Routing-Tabelle entsprechend konfigurierbar sein. Der Test ist erfolgreich, wenn anschließend ein Client einen DNS-Query über den Router an die IPv6-Multicast-Adresse FF08::114 sendet und eine positive Antwort erhält.

Szenario 1.b (Anycastanforderungen):

Client:
Da sich eine IPv6-Anycast-Adresse nicht von anderen globalen IPv6-Adressen unterscheiden lässt, ist die DNS-Kommunikation für den Client vollkommen transparent. Es ändert sich aus seiner Sicht nichts, wodurch auch keine neuen Anforderungen entstehen. Die Anforderungen sind dieselben wie bei einer Unicast-Kommunikation.

Server:
Als grundlegende Anforderung bei der Konfiguration des DNS-Dienstes ist es notwendig, die IPv6-Anycast-Adresse als Listening-Adresse einzutragen, damit der Dienst auf dieser Adresse auf Client-Anfragen reagiert.

Zudem muss laut RFC 4291 "IPv6 Addressing Architecture" eine IPv6-Adresse, die durch eine Verwendung auf mehreren Schnittstellen zu einer IPv6-Anycast-Adresse wird, explizit als solche festgelegt werden. Das Deklarieren der Anycast-Adresse ist erforderlich, damit beim Neighbor Discovery Protocol das O-Flag (Override Flag) nicht gesetzt wird.

Im Test wird überprüft, welche IPv6-Adresse der Server standardmäßig bei DNS-Reply-Nachrichten verwendet und ob es möglich ist, hierfür die IPv6-Anycast-Adresse zu konfigurieren. Zudem wird kontrolliert, ob die IPv6-Anycast-Adresse als solche deklariert werden kann. Damit wäre die Konfiguration stan-

dardkonform und es könnten, falls gewünscht, zwei DNS-Server in einem Subnetz optimal betrieben werden.

Router:
Auch für den Router entstehen keine weiteren Anforderungen. Nur dynamisches Routing sollte möglich sein.

4.2 Verfügbarkeit

Bei den nächsten Testszenarien handelt es sich um eine Reihe von Tests zur Verfügbarkeit des DNS-Services. Die Verfügbarkeit wird laut International Telecommunication Union (ITU) wie folgt definiert:

> *"Die Fähigkeit einer Einheit, eine angeforderte Funktion zu jedem beliebigen Zeitpunkt innerhalb eines vorgegebenen Zeitintervalls erfolgreich durchzuführen, unter der Voraussetzung, dass externe Ressourcen, soweit gefordert, vorhanden sind." [ITU, 2007]*

Es wird überprüft, ob der DNS-Dienst noch erreichbar ist und Ergebnisse liefert. Der Versuchsaufbau bzw. die Infrastruktur bleiben gleich. Vor dem eigentlichen Test wird jedoch eine Netzkomponente deaktiviert bzw. unter hohe Last gesetzt. Diese Tests zeigen, wie die einzelnen Kommunikationsmodelle auf Ausfälle oder Beeinträchtigungen reagieren und wie hoch die jeweilige Verfügbarkeit ist.

Szenario 2.a (Verfügbarkeit bei Ausfall des DNS-Servers): Bei diesem Test ist der primäre DNS-Server, genauer gesagt das Netz-Interface des Servers, deaktiviert und nicht erreichbar. Ein DNS-Query wird vom Client verschickt und es wird protokolliert, ob dieser eine entsprechende Antwort erhält. Die benötigte Response-Time und der interne Ablauf im Betriebssystem, um eine Antwort zu erhalten, sind hier nicht relevant.

Szenario 2.b (Verfügbarkeit bei hoher Server-Last): In diesem Szenario ist der oben genannte Server wieder erreichbar. Diesmal wird der primäre Server einer hohen Last ausgesetzt. Simuliert wird dies mit einem Programm auf dem DNS-Server, das die Verarbeitung des DNS-Paketes zeitlich verzögert.

Für den Test führt der Client wieder jeweils über Anycast, Multicast und Unicast eine Namensauflösung durch. Hierbei ist nicht die Response Time wichtig. Entscheidend ist nur, ob der Service verfügbar ist und somit ein Antwort-Paket verschickt.

Szenario 2.c (Verfügbarkeit bei Ausfall des DNS-Dienstes): In diesem Szenario ist der DNS-Dienst (z.B. bind-daemon) des primären DNS-Servers ausgefallen. Das heißt, der Dienst antwortet nicht mehr auf DNS-Querys, weil er z. B. eine Fehlfunktion aufweist.

Für den Test wird der DNS-Dienst des primären Servers beendet und der Client schickt je Kommunikationsmodell eine DNS-Anfrage an die Server. Auch hier ist nur die Verfügbarkeit und nicht die Response Time von Bedeutung.

Szenario 2.d (Verfügbarkeit bei Ausfall eines Routers): In Szenario 2.d ist der Router R2 ausgefallen. Dazu wird dieser vor dem Test heruntergefahren oder es werden alle seine Schnittstellen deaktiviert. Danach folgt, wie in den Szenarien davor, eine DNS-Anfrage je Kommunikationsmodell vom Client an die DNS-Server.

4.3 Response-Time

Die Response-Time, auch Antwortzeit genannt, kann wie folgt definiert werden:

> *"Die Antwortzeit ist die Zeitspanne zwischen der ausgelösten Aktion und der korrelierenden Reaktion. Sie beinhaltet alle Laufzeiten, die Latenzzeiten des Netzwerks, die Berechnungs-, Bearbeitungs- und Reaktionszeiten." [DATACOM Buchverlag GmbH]*

Bei DNS kann es vorkommen, dass ein Antwort-Paket den Client nicht rechtzeitig erreicht oder gar nicht versendet wird. Da einige nachfolgende Tests genau diese Problematik verursachen, wird die Response-Time, wie in Abbildung 10 dargestellt, gemessen. Die Zeitaufnahme beginnt in dem Moment (T_{start}), bei dem das erste UDP-Paket zu einem DNS-Request vom Client versendet wird. Sie endet zu dem Zeitpunkt, an dem das erste Antwortpaket den Client erreicht (T_{end}). Aus der Differenz von T_{end} und T_{start} ergibt sich die Response-Time. Die Latenz bei der Verarbeitung des Paketes durch den Resolver des Clients (beim senden und empfangen) ist bei jedem Kommunikationsmodell gleich und wird daher bei dieser Messung nicht berücksichtigt. Erhält der Resolver keine Antwort, liefert er nach einer gewissen Zeit einen Negativeintrag an die Anwendung zurück. In diesem Fall wird dieser Zeitpunkt als T_{end} gewertet. Die Messung bzw. Aufzeichnung der Pakete findet auf dem Client mithilfe von Tcpdump statt.

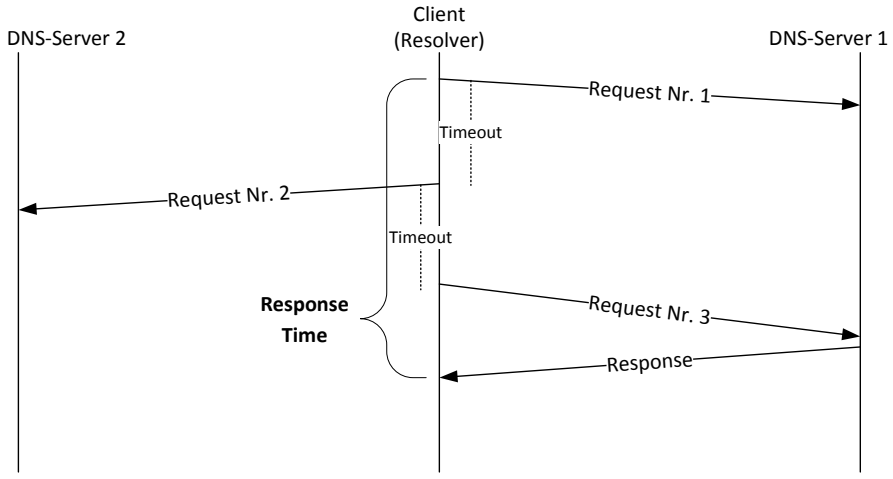

Abbildung 10: Messung der Response-Time

Eine geringe Response-Time bei der Namensauflösung ist in vielerlei Hinsicht wichtig. Wird die Namensauflösung erheblich verzögert, können einige Pro-

grammabläufe zeitlich gestört werden oder sogar in einen Timeout laufen. Der interne Resolver von Windows 7 gibt beispielsweise nach spätestens 12 Sekunden ohne Antwort vom DNS-Server einen negativen DNS-Eintrag an die Anwendung zurück, obwohl der Eintrag auf dem DNS-Server vorhanden wäre.

Aber auch geringere Antwortzeiten sollten vermieden werden. Studien haben gezeigt, dass Besucher von Webseiten nach 2-4 Sekunden Ladezeit das Interesse an einer Webseite verlieren und diese wieder verlassen [Nah, 2004] & [Jupiter Research, 2006]. Da die Namensauflösung nur einen kleinen Teil der Ladezeit einer Webseite ausmacht, sollte die Antwortzeit bei der Namensauflösung deutlich unter 2 Sekunden liegen.

Zudem schreibt ein älterer Standard (RFC 2010) für den Betrieb von Root Namensservern eine maximale Latenz von 5ms vor. Dies ist zwar für einen Namensserver innerhalb eines Unternehmens nicht bindend, kann aber als Referenz herangezogen werden.

Der Resolver-Cache auf den Clients ist, falls nicht anders angegeben, bei allen Szenarien deaktiviert. Dadurch können die Namensauflösungen unabhängig voneinander betrachtet werden.

Szenario 3.a (Response-Time ohne Last): Zuerst findet eine Überprüfung der Antwortzeit ohne Last statt. Dazu sendet der Client zehn DNS-Querys an den Server und die benötigte Zeit der einzelnen Querys (vom Client zum Server und wieder zurück) wird gemessen. Aus diesen zehn Werten ergibt sich die Durchschnittszeit. Diese Berechnung erfolgt jeweils für das Anycast, Multicast und Unicast Szenario.

Szenario 3.b (Response-Time mit Last auf dem DNS-Server): Hier findet die gleiche Prozedur statt. Der Unterschied besteht darin, dass der primäre DNS-Server unter hoher Last steht. Mittels Last-Tools, die auf dem DNS-Server selbst installiert werden und diesen auslasten, wird dabei jeweils eine Latenz bei der Namensauflösung von 0,5/1/2 Sekunden erzeugt.

Szenario 3.c (Response-Time bei Ausfall des DNS-Servers): Im letzten Szenario bzgl. der Response-Time-Überprüfung ist der primäre DNS-Server komplett nicht zu erreichen. Es können also keine DNS-Querys von ihm verarbeitet werden. Ansonsten entspricht es Szenario 2.a. Der Client vollführt wieder zehn Namensauflösungen, aus denen anschließend eine Durchschnittszeit ermittelt wird.

4.4 Änderungsaufwand

In diesen Szenarien wird der Aufwand bei Veränderungen der Konfiguration oder der Umgebung ermittelt. Änderungen an Clients, Routern oder Servern werden abhängig von ihrer Komplexität und ihrem zeitlichen Umfang den entsprechenden Kategorien zugeordnet. Diese reichen von "sehr niedrig" über "mittel" bis "extrem hoch".

Komplexität \ Zeit	< 10min	≥ 10min und < 1std	≥ 1std und < 4std	> 4std
gering	sehr niedrig	niedrig	mittel	sehr hoch
hoch	niedrig	mittel	hoch	extrem hoch

Tabelle 8: Einordnung des Änderungsaufwandes

Tabelle 8 gibt den Änderungsaufwand abhängig von dem geschätzten Zeitumfang für die Änderung und deren Komplexität wieder.

Szenario 4.a (Ersetzen eines DNS-Servers): In diesem Szenario wird der Aufwand analysiert, der bei Austausch und Neuinstallation eines DNS-Servers entsteht (z. B. auf Grund eines Hardwaredefekts).

Szenario 4.b (Änderung der DNS-Server IP-Adresse): Aus technischen oder organisatorischen Gründen kann es notwendig sein, einen DNS-Server mit einer anderen IP-Adresse zu versehen oder in ein anderes Netz zu verschieben. Die hieraus resultierenden Veränderungen an der ganzen Umgebung (z.B. Konfigurationsänderungen) werden hier für jedes Kommunikationsmodell evaluiert.

Szenario 4.c (Erweiterung um einen DNS-Server): Um die Verfügbarkeit zu erhöhen, kann die Inbetriebnahme eines weiteren DNS-Servers notwendig werden. Es wird die Höhe des Aufwandes analysiert, wenn der weitere DNS-Server zur Erhöhung der Verfügbarkeit zum Einsatz kommt.

4.5 Netzlast

Durch die Veränderung des Kommunikationsmodells ändert sich auch die Anzahl der Pakete, die eine Netzkomponente versendet oder eine andere verarbeiten muss. Wichtig ist diese Überlegung bei der Dimensionierung der Netzkomponenten und sollte daher vor einem möglichen Wechsel des Kommunikationsmodells miteinbezogen werden. Das nachfolgende Szenario beschreibt den entsprechenden Test.

Szenario 5.a (Netzlast): In diesem Szenario wird die Last für eine Namensauflösung je Kommunikationsmodell erfasst. Über Router 1 läuft jeglicher Datenverkehr der Clients bei einer Namensauflösung. Daher werden auf diesem die übertragenen Pakete bei 1000 Namensauflösung und die zu übertragene Datenmenge mittels eines Netz-Sniffers ermittelt.

4.6 Netzmanagement

Netzmanagement bezeichnet „die Summe aller Aktivitäten und Maßnahmen zur Sicherstellung des effektiven und effizienten Einsatzes der Netzressourcen im Sinne des Unternehmens bzw. des Kunden“ [Leischner, 2012].

Bei einer Unicast Kommunikation sind die Anforderungen und Umsetzungen dafür eindeutig und bekannt. Dies ändert sich jedoch bei der Verwendung von Anycast und Multicast. Hier ist nicht jeder Host eindeutig mittels einer IPv6-Adresse identifizierbar. Zudem ist die korrekte und vollständige Funktionalität des Services und aller beteiligten Komponenten komplexer. Die nachfolgenden Szenarien beschreiben diese Probleme genauer und geben konkrete Testfälle wieder.

Der Standard „Requirements for Management of Name Servers for the DNS“ (RFC 6168) beschreibt, welche Managementanforderungen bei dem Betrieb eines DNS-Servers gestellt werden. Die notwendigen Aufgaben sind in vier Kategorien dargestellt und müssen auch mit Anycast oder Multicast sinnvoll durchgeführt werden können:

- Kontrolle
- Konfiguration
- Überwachung
- Alarm

Für die Tests wird jeweils aus den beiden Kategorien "Kontrolle" und "Konfiguration", sowie "Überwachung" und "Alarm" ein Testszenario gebildet. In den beiden Szenarien wird überprüft, ob die Aufgaben (nach RFC 6168) generell möglich sind und welche Anforderungen sie mit sich bringen.

Szenario 6.a (Kontrolle und Konfiguration):

Unter Kontrollaufgaben wird hier die Möglichkeit verstanden, jeden einzelnen DNS-Server neu zu starten oder die Zonendatei neu zu laden. Für die Konfiguration eines Systems muss es zudem möglich sein, auf jedem DNS-Server z. B. einzelne DNS-Einträge zu bearbeiten oder andere DNS-spezifische Einstellungen vorzunehmen.

Szenario 6.b (Überwachung und Alarm):

In die Kategorie "Überwachung" fallen Aufgaben, die das Monitoring des Service-Status und das Auslesen von Performance-Countern beinhalten. Zu der Kategorie "Alarm" gehören Managementfunktionen, die eine automatische Benachrichtigung bei Fehlern und außergewöhnlichen Ereignissen ermöglichen. Ein Beispiel hierfür wäre der Ausfall eines Systems oder des Services.

4.7 Sicherheit

In diesen Szenarien wird überprüft, wie die Kommunikationsmodelle Unicast, Anycast und Multicast auf bestimmte Bedrohungen reagieren und wie anfällig sie sind. Laut IT-Grundschutz sind Vertraulichkeit, Integrität und Verfügbarkeit bei dem Einsatz von DNS-Services besonders wichtig [BSI, 2012]. Die ITU beschreibt diese wie folgt:

Vertraulichkeit:

> *"Die Eigenschaft, dass Informationen nicht für unbefugte Personen, Unternehmen und Prozesse verfügbar oder einsehbar sind." [ITU, 1991]*

Integrität:

> *"Die Eigenschaft, dass Daten nicht in einer unberechtigten Weise verändert wurden."[ITU, 1991]*

Verfügbarkeit:

> *"Die Fähigkeit einer Einheit, eine angeforderte Funktion zu jedem beliebigen Zeitpunkt innerhalb eines vorgegebenen Zeitintervalls erfolgreich durchzuführen, unter der Voraussetzung, dass externe Ressourcen, soweit gefordert, vorhanden sind." [ITU, 2007]*

Die nachfolgenden Tests enthalten unter anderem einige der im IT-Grundschutz aufgelisteten, sowie speziell für die Kommunikationsmodelle relevanten Gefährdungen. Nicht überprüft und beschrieben werden hier grundlegende Sicherheitsaspekte für den Einsatz von DNS, die z.B. das (Betriebs-)System, die Planung oder ähnliches betreffen. Hier sei auf den IT-Grundschutz verwiesen.

Szenario 7.a (Sicherheit bei Denial-of-Service-Angriff):

Denial-of-Service-Angriffe haben das Ziel, legitime Benutzer eines IT-Systems an der Nutzung des Systems zu hindern. Bei DNS-Servern wird dies durch viele Anfragen an den DNS-Server erreicht, wodurch die Netzverbindung zum DNS-Server oder der DNS-Server selbst überlastet ist. Dieser ist nun nicht mehr in der Lage, alle Anfragen zu verarbeiten. Die Verfügbarkeit des DNS-Services sinkt bzw. ist nicht mehr vorhanden.

Testszenario:
Die Verfügbarkeit und die Response-Time des DNS-Dienstes werden mithilfe eines Monitoring-Tools auf einem Client im Client-Netz gemessen. Dieses sendet jede Sekunde eine DNS-Anfrage und protokolliert das Ergebnis. Weitere Clients versuchen durch eine massive Menge an DNS-Anfragen, den DNS-Dienst hinsichtlich Verfügbarkeit und Respones-Time negativ zu beeinflussen.

Szenario 7.b (Sicherheit bei DNS-Hijacking):

DNS-Hijacking ist ein Angriff, bei dem der DNS-Verkehr zwischen Resolver und DNS-Server über das System des Angreifers umgeleitet wird. Der Angreifer hat somit die Möglichkeit,

- die Kommunikation abzuhören und aufzuzeichnen,
- Pakete zu modifizieren und weiterzuleiten oder
- eigene Antwortpakete zu versenden.

Bedroht sind hierbei die Vertraulichkeit und die Integrität.

Testszenario:
In diesem Szenario wird analysiert, ob und wie der DNS-Verkehr in den drei Kommunikationsmodellen über einen Server des Angreifers geleitet werden kann. Dies soll Aufschluss darüber geben, ob die Kommunikationsmodelle anfällig auf DNS-Hijacking-Angriffe sind.

Szenario 7.c (Sicherheit bei DNS-Spoofing):

DNS-Spoofing bezeichnet einen Angriff mit dem Ziel, die Zuordnung zwischen IP-Adressen und Domainnamen zu verfälschen. Dazu wird versucht, eine DNS-Reply-Nachricht zu fälschen und dem Client manipulierte Daten unterzuschieben. Auch in diesem Szenario sind die Integrität und die Vertraulichkeit gefährdet.

DNS-Replies werden akzeptiert, wenn folgende Punkte erfüllt sind:

- Die ID (Zufallszahl) des Querys und des Replys müssen übereinstimmen.
- Die Reply-Nachricht muss an die korrekte IP-Adresse und den Port des Clients gerichtet sein. (Übereinstimmung mit der entsprechenden Query)

Testszenario:
Dieser Test soll zeigen, ob es mithilfe der Unicast-, Anycast- oder Multicast-Kommunikation möglich ist, eine DNS-Antwort zu fälschen bzw. DNS-Spoofing zu betreiben.

Szenario 7.d (Sicherheit bei DNS Information Leakage):

Als DNS Information Leakage wird das Ausspähen von internen Domain-Informationen bezeichnet. Der Gewinn dieser Informationen stellt zwar keinen direkten Schaden dar, kann aber zur Vorbereitung späterer Angriffe verwendet werden. Interne Domain-Informationen können zum Beispiel die Funktion oder der Standort gewisser IT-Komponenten sein, wodurch der Angreifer sich einen Überblick über das Netz und die lohnenden Ziele verschafft. Je mehr Informationen der Angreifer über das Angriffsziel besitzt, desto größer ist die Wahrscheinlichkeit für einen erfolgreichen Angriff [BSI, 2012]. Da hier keine Daten verändert werden, ist nur die Vertraulichkeit der Daten gefährdet.

Testszenario:
In diesem Szenario wird versucht, an Informationen (DNS-Records) des DNS-Servers zu gelangen.

Szenario 7.e (Sicherheit: Einsatz von DNSSec):

Neben dem Verhalten bei akuten Bedrohungen ist es wichtig, sich im Vorhinein durch entsprechende Maßnahmen davor zu schützen. Bei DNS bietet die Erweiterung DNSSEC (RFC 4033, 4034, 4035) die Möglichkeit, die Integrität und die Authentizität der Nachrichten sicherzustellen.

Testszenario:
Es wird geprüft, ob eine der Kommunikationsmodelle den Einsatz von DNSSec erschwert oder ganz verhindert.

4.8 Aktualität der Antworten

Werden Änderungen an der Zone vorgenommen, ist es wichtig, dass diese von allen Clients und Servern wahrgenommen und schnell übernommen werden. Daher ist es notwendig, die Update-Zyklen und Prozeduren der beteiligten Systeme sowie deren Möglichkeit zum Cachen der DNS-Einträge zu betrachten.

Testszenario 8.a (Aktualität der DNS-Antworten): In diesem Szenario wird analysiert, ob der geänderte DNS-Eintrag auf dem Server auch bei einer DNS-Anfrage auf dem Client erscheint bzw. wie lange noch der alte Wert verwendet wird.

Dazu wird auf dem primären DNS-Server der AAAA-Record für die Hostnamen mail.test-dns.netlab.inf.h-brs.de von 2001:db8::500 auf 2001:db8::600 geändert. Unverzüglich danach fragt ein Client diesen Record ab, indem er einen Ping-Request auf die mail.test-dns.netlab.inf.h-brs.de durchführt.

Dieser Test findet für jede Kommunikationsmodelle statt. Der Zonentransfer wird einmal mit und einmal ohne Notify-Update des DNS-Servers durchgeführt. Der Update-Zyklus beträgt 60 Minuten. Somit ergeben sich insgesamt sechs Tests.

5 Testergebnisse

In diesem Kapitel sind die Testergebnisse der in Kapitel 4 vorgestellten Testszenarien aufgeführt. Die Reihenfolge entspricht der Abfolge des vorigen Kapitels. Jedes Testszenario bildet jeweils ein Unterkapitel. Als erstes sind die Ergebnisse der Systemanforderung dargestellt.

5.1 Systemanforderungen

Zunächst werden die Anforderungen an die Multicast-Umgebung untersucht. Anschließend folgt die Überprüfung der Anycast-Umgebung.

5.1.1 Multicastanforderungen – Szenario 1.a

Im ersten Szenario werden die Anforderungen an die Clients, DNS-Server und Router für eine IPv6-Multicast-Kommunikation überprüft. Ein Merkmal der DNS-Multicast-Kommunikation ist es, dass beide DNS-Server gleichzeitig abgefragt werden. Dazu wird vom Resolver ein DNS-Query an eine IPv6-Multicast-Adresse gesendet. Beide DNS-Server sind Mitglieder dieser Gruppe. Mittels des Multicast-Routings wird der DNS-Query an alle Mitglieder geleitet. Beide Server antworten auf den Query, wodurch der Resolver zwei Antwort-Pakete auf seine Namensauflösung erhält.

Client:
Die automatische Verteilung der IPv6-Multicast-Adresse des DNS-Dienstes ist möglich. Die getesteten Router-Advertisement- und DHCPv6-Dienste konnten die IPv6-Multicast-Adresse problemlos an die Clients verteilen. Die Standardkonfiguration der Dienste ist daher ausreichend. Die Resolver auf den Clients sind in der Lage, die erhaltenen Multicast-Adressen für ihre Namensauflösung zu verwenden.

Server:
Die vier Anforderungen an die DNS-Server werden anfänglich nur teilweise erfüllt. Wie in Kapitel 3.3 zur Konfiguration der Multicast-Umgebung beschrieben, muss ein Multicast-Proxy verwendet werden, damit die Server die Multicast-Gruppe FF08::114 betreten und der DNS-Dienst Anfragen an diese Multicast-Adresse verarbeiten können.

Anforderung 1:
Die DNS-Server können erfolgreich den IPv6-Multicast-Gruppen beitreten.

Anforderung 2:
Ein ICMPv6-Echo-Request aus dem lokalen Netz an die DNS-Server-Multicast-Adresse FF02::114 ist erfolgreich. Die Anforderung des Sendens und Empfangens von IPv6-Multicast-Paketen ist gegeben.

Anforderung 3:
Die IPv6-Multicast-Adresse kann zwar ohne Schwierigkeiten in der Bind-Konfiguration eingetragen werden, aber es folgt durch Bind kein Joinen der entsprechenden Multicast-Gruppe. Ein manuelles Beitreten dieser Gruppe ist auch nicht erfolgreich, da die DNS-Pakete an die Multicast-Adresse weiterhin nicht an die DNS-Anwendung übergeben werden. Erst der oben erwähnte Multicast-Proxy ermöglicht es, DNS-Anfragen an die IPv6-Multicast-Adresse zu senden.

Anforderung 4:
Bei den Tests mit allen Clientsystemen führt die Ungleichheit zwischen Ziel-Adresse der Anfrage und Quell-Adresse des Antwort-Paketes zu keinen Problemen. Der Client verarbeitet das zuerst ankommende Paket korrekt. Das DNS-Antwort-Paket des zweiten DNS-Servers verwirft der Client.

Router:
Mithilfe des Multicast-Routings sind die Router in der Lage, die DNS-Querys mit der Ziel-Adresse FF08::114 an die beiden DNS-Server zu routen. Die DNS-Antworten der Server sind an die IPv6-Unicast-Adresse der Clients gerichtet und besitzen als Quell-Adresse die IPv6-Unicast-Adresse der DNS-Server. Mittels Unicast-Routing werden die Pakete zurück an die Clients geleitet.

5.1.2 Anycastanforderungen – Szenario 1.b

Die Besonderheit der Anycast-Kommunikation ist es, dass der Resolver nur eine IPv6-Anycast-Adresse für seine Namensauflösungen besitzt. Über diese Adresse sind beide DNS-Server erreichbar. Das bedeutet dass, an beide DNS-Server die Anycast-Adresse gebunden ist. Bei einer Namensauflösung wird das DNS-Paket allerdings nur an einen der Server geleitet. Der Resolver besitzt keinen Einfluss auf diese Wahl, sondern das Routing-System bestimmt das Zielsystem (DNS-Server). In der Regel ist das der dem Client nächstgelegene Server.

Da für den Client keine Anforderungen bzgl. Anycast bestehen und die Router nur dynamisches Routing unterstützen müssen, werden in diesem Abschnitt nur die DNS-Server Voraussetzungen getestet.

Server:
Wie in Kapitel 3.2 beschrieben, ist die Verwendung der IPv6-Anycast-Adresse in der Bind-Konfiguration auf den beiden DNS-Servern problemlos möglich. Die IPv6-Anycast-Adresse kann zudem als Quell-Adresse der Antwortpakete konfiguriert werden.

Die an die DNS-Server gebundene IPv6-Adresse kann als IPv6-Anycast-Adresse konfiguriert werden. Allerdings ist bei Ubuntu das O-Flag im NDP nicht korrekt gesetzt. Auf den weiteren Testverlauf hat das aber keine Auswirkungen, da sich beide DNS-Server nicht im selben Subnetz befinden.

	Multicast	Anycast
Clientanforderungen	Erfüllt	Keine Vorhanden
Serveranforderungen	Erfüllt*	Erfüllt
Routeranforderungen	Erfüllt	Keine Vorhanden
DNS-Kommunikation möglich	Ja*	Ja

* Mittels Workaround (Multicast-Proxy) möglich

Tabelle 9: Ergebnisübersicht der erfüllten Anforderungen

5.2 Verfügbarkeit

Überprüft wird, ob der Resolver für jedes Kommunikationsmodell ein Antwort-Paket auf seine Namensauflösung erhält. Das Antwort-Paket muss dabei innerhalb der maximalen Wartezeit des Resolvers eintreffen, da der Resolver ansonsten einen Negativeintrag zurückliefert. Die maximale Wartezeit der Resolver ist in Kapitel 3.4 beschrieben.

5.2.1 Verfügbarkeit bei Ausfall des DNS-Servers – Szenario 2.a

In diesem Test wird die Verfügbarkeit des DNS-Dienstes bei einem Ausfall des primären DNS-Servers (DNS-Server 1) getestet.

Unicast:
Der Client schickt eine DNS-Anfrage an die IPv6-Adresse des primären Servers. Aufgrund des Ausfalls des Servers erhält der Resolver keine Antwort, wodurch er nach einer gewissen Zeit (abhängig vom Client-Betriebssystem) den zweiten DNS-Server anfragt. Der sekundäre Server liefert erfolgreich eine Antwort an den Resolver.

Multicast:
Der Resolver sendet eine Anfrage an die IPv6-Multicast-Adresse FF08::114, womit sie an beide Server geleitet wird. Da der primäre Server nicht erreichbar ist, erhält der Resolver nur eine Antwort vom sekundären Server.

Anycast:
Eine DNS-Anfrage wird vom Resolver an die eingetragene IPv6-Anycast-Adresse gesendet. Durch den Ausfall des Servers ist dieser nicht mehr in der Lage, OSPF-Hello-Pakete zu verteilen. In der Folge führt nach spätestens 40 Sekunden (DeadInterval) keine Route mehr zu ihm. Router 1 wird daher die DNS-Anfrage über Router 3 an den sekundären Server leiten. Der Resolver erhält von DNS-Server 2 eine Antwort.

5.2.2 Verfügbarkeit bei hoher Server-Last – Szenario 2.b

In diesem Szenario befindet sich der DNS-Server 1 unter hoher Last.

Unicast:
Der Resolver des Clients sendet zur Namensauflösung ein DNS-Query an den primären DNS-Server. Durch die Überlast reagiert er nicht schnell genug auf die Anfrage, sodass der Resolver einen weiteren DNS-Query sendet. Wie lange der Resolver wartet bzw. welchen DNS-Server er als nächstes anfragt, unterscheidet sich je nach Betriebssystem (siehe Kapitel 3.4). Nach einer gewissen Zeit fragt der Resolver jedoch den zweiten DNS-Server an und erhält eine Antwort.

Multicast:
Für die Namensauflösung sendet der Resolver einen DNS-Query an die Multicast-Adresse FF08::114. Das Paket wird von beiden Routern (Router 2 und 3) an die beiden DNS-Server weitergeleitet. Da der primäre DNS-Server unter hoher Last steht, kann dieser nicht rechtzeitig antworten. Der sekundäre Server antwortet aber innerhalb einiger Millisekunden.

Anycast:
Bei Anycast kann es vorkommen, dass die Verfügbarkeit des DNS-Dienstes nicht gegeben ist. Es ist möglich, dass der DNS-Server aus Sicht des Routers 2 und des OSPF-Protokolls noch erreichbar ist, da dem Server hierfür noch genügend Ressourcen zur Verfügung stehen und er innerhalb des RouterDead-Intervals von 40 Sekunden ein OSPF-Hello-Paket versendet. Dadurch wird die Route zwischen Router 2 und DNS-Server 1 weiterhin propagiert. Andererseits ist die Last des Servers so hoch, dass er nicht mehr rechtzeitig auf DNS-Querys antworten kann. Der Resolver ist durch die Verwendung der IPv6-Anycast-Ardesse nicht in der Lage, mit einem alternativen DNS-Server zu kommunizieren. Die DNS-Querys werden von den Routern weiterhin an den primären DNS-Server geleitet. Dadurch liefert der Resolver irgendwann einen Negativeintrag zurück.

Abbildung 11 zeigt am Beispiel der Standardkonfiguration (Timeouts) des Ubuntu-Resolvers und der Quagga-OSPF-Konfiguration, bei welcher Konstellation es trotz Anycast zu einem Ausfall des DNS-Dienstes kommen kann.

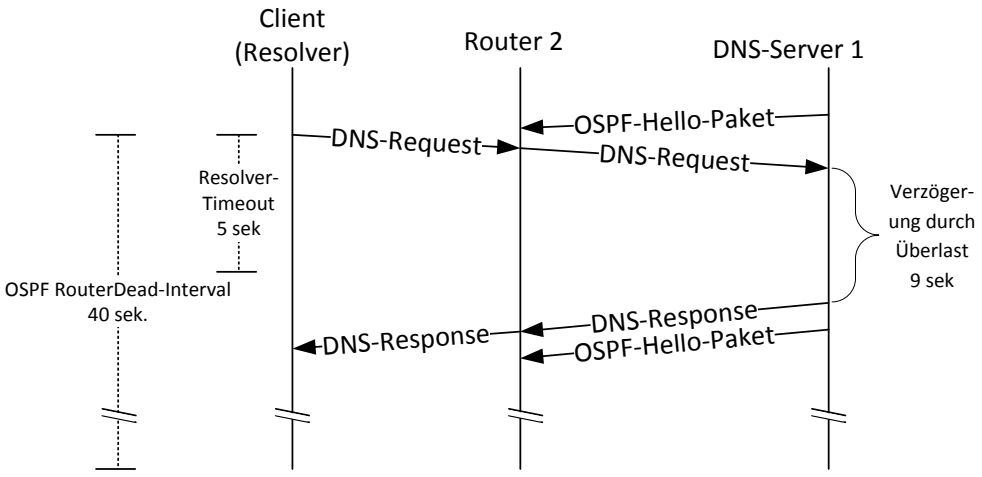

Abbildung 11: Verfügbarkeit bei Anycast ist nicht immer gegeben

Um dem entgegenzuwirken, ist es sinnvoll, den Timeout des Resolvers und das RouterDead-Interval exakter aufeinander abzustimmen. Dazu sollte das RouterDead-Interval verkürzt werden. Bei einer Überlast des primären DNS-Servers können dann die DNS-Pakete rechtzeitig auf den sekundären Server geleitet werden. Problematisch bleibt jedoch weiterhin die alleinige Überlastung des DNS-Dienstes. Dann antwortet dieser nicht rechtzeitig, aber die OSPF-Hello-Pakete werden ohne Verzögerung versendet.

5.2.3 Verfügbarkeit bei Ausfall des DNS-Dienstes – Szenario 2.c

Im Unterschied zum ersten Test bzgl. der Verfügbarkeit ist in diesem Szenario nicht der gesamte DNS-Server 1 ausgefallen, sondern nur der DNS-Dienst auf diesem Server.

Unicast:
Der Client schickt eine Anfrage an den primären Server. Durch den Ausfall des DNS-Dienstes auf diesem Server erhält der Resolver keine Antwort, wodurch der Resolver des Clients nach einer gewissen Zeit den zweiten DNS-Server anfragt.

Multicast:
Der Resolver sendet eine Anfrage an die Multicast-Adresse (FF08::114), wodurch die Anfrage an beide Server geleitet wird. Da der DNS-Dienst auf dem primären Server nicht erreichbar ist, erhält der Resolver nur eine Antwort vom sekundären DNS-Server.

Anycast:
Eine DNS-Anfrage wird vom Resolver an die eingetragene Anycast-Adresse gesendet. Da der primäre Server an sich noch erreichbar ist (aus der Sicht des Routers), wird das Paket von den Routern an den primären Server geleitet. Dort kann es jedoch nicht verarbeitet werden, da der DNS-Dienst nicht verfügbar ist. Der Resolver versucht nun, weitere DNS-Server nach seiner vorgegebenen Prozedur zu erreichen (siehe Kapitel 3.4). In seiner Konfiguration sind aber keine weiteren DNS-Server eingetragen, wodurch jegliche Versuche auf allen vorhandenen Interfaces fehlschlagen.

5.2.4 Verfügbarkeit bei Ausfall eines Routers – Szenario 2.d

Die Testergebnisse in diesem Abschnitt geben Aufschluss über die Verfügbarkeit bei Ausfall von Router 2. Hierfür wird der Router heruntergefahren.

Unicast:
Durch den Ausfall des Routers 2 kann der Resolver des Clients nicht mehr den primären DNS-Server erreichen. Alle getesteten Clients versuchen daher, den zweiten DNS-Server anzufragen. Dieser ist verfügbar und kann den DNS-Query des Clients beantworten.

Multicast:
Beide DNS-Server sind Mitglied der Multicast-Gruppe FF08::114, wodurch ein

DNS-Query des Clients in der Regel von beiden Servern empfangen wird. Da Router 2 ausgefallen ist, können von ihm keine Pakete an den DNS-Server 1 weitergeleitet werden. Über Router 3 erhält der sekundäre DNS-Server aber weiterhin DNS-Querys, die an die IPv6-Multicast-Adresse gesendet werden.

Anycast:
Wenn die Namensauflösung des Clients in dem Moment gestartet wird, in dem Router 2 ausfällt, tritt folgendes ein:

Die DNS-Pakete können dem primären DNS-Server nicht zugestellt werden. Der Ausfall des Routers ist durch den RouterDead-Interval von 40 Sekunden noch nicht bemerkt worden. Der Resolver versucht, nach einer festen Prozedur (wie in Kapitel 3.4 beschrieben) weitere DNS-Querys an die IPv6-Anycast-Adresse der DNS-Server zu senden. Solange aber der Ausfall des Routers 2 noch nicht bemerkt wurde, wird weiterhin versucht, die Pakete an den DNS-Server 1 zu leiten. Dies kann bis zu 40 Sekunden dauern. Die Resolver tolerieren jedoch nur einen Ausfall von maximal 7 bis 22 Sekunden, bevor sie einen Negativeintrag zurückgeben.

Sendet der Resolver hingegen mindestens 40 Sekunden nach dem Ausfall einen DNS-Query, wird dieses Paket direkt an den sekundären DNS-Server geleitet.

Tabelle 10 zeigt einen zusammenfassenden Überblick über die Testergebnisse zur Verfügbarkeit.

Verfügbarkeit	**Unicast**	**Multicast**	**Anycast**
bei Ausfall des primären DNS-Servers	Vorhanden	Vorhanden	Vorhanden
bei hoher Server-Last	Vorhanden	Vorhanden	Teils vorhanden
bei Ausfall des DNS-Dienstes	Vorhanden	Vorhanden	Nicht vorhanden
bei Ausfall eines Routers	Vorhanden	Vorhanden	Vorhanden

Tabelle 10: Testergebnisse der Verfügbarkeit

5.3 Response-Time

In diesem Abschnitt wird die Response-Time (Antwortzeit) des DNS-Services ermittelt. Die Antwortzeit gibt Aufschluss darüber, wie lange der Client je nach Kommunikationsmodell auf eine DNS-Antwort warten muss.

5.3.1 Response-Time ohne Last – Szenario 3.a

Die durchschnittliche Antwortzeit bei DNS-Anfragen ist in allen drei Kommunikationsmodellen gleich und liegt bei sechs Millisekunden. Keines der Kommunikationsmodelle verursacht eine höhere Latenz beim Routing oder bei der Verarbeitung der Pakete auf den DNS-Servern.

5.3.2 Response-Time bei Last auf dem DNS-Server – Szenario 3.b

Bei einer für diesen Test erzeugten Latenz von durchschnittlich einer halben Sekunde auf dem primären DNS-Server ergeben sich folgende mittlere Antwortzeiten für eine Namensauflösung:

	Unicast	Multicast	Anycast
Ubuntu Client	500ms	6ms	500ms
Windows Client	500ms	6ms	500ms
Mac OS X Client	500ms	6ms	500ms
Windows Phone	500ms	6ms	500ms
Android	500ms	6ms	500ms
iOS	500ms	6ms	500ms

Tabelle 11: Antwortzeiten bei einer Latenz von 0,5 Sekunden

Da bei Multicast gleichzeitig der sekundäre DNS-Server angefragt wird, ist dort die Antwortzeit am geringsten. Die Ergebnisse der Resolver der einzelnen Betriebssysteme unterscheiden sich dabei nicht. Auch bei den Kommunikationsmodellen Unicast und Anycast sind die Antwortzeiten der einzelnen Resolver gleich. Es wird immer der primäre DNS-Server angefragt. Dadurch entsteht eine Antwortzeit, die der Latenz des primären DNS-Servers entspricht.

Steigt die Latenz für die Verarbeitung der DNS-Pakete auf dem primären DNS-Server auf eine Sekunde, zeigen sich folgende Antwortzeiten:

	Unicast	Multicast	Anycast
Ubuntu Client	1000ms	6ms	1000ms
Windows Client	1000ms	6ms	1000ms
Mac OS X Client	1000ms	6ms	1000ms
Windows Phone	1000ms	6ms	1000ms
Android	1000ms	6ms	1000ms
iOS	1000ms	6ms	1000ms

Tabelle 12: Antwortzeiten bei einer Latenz von einer Sekunde

Auch hier liefert Multicast die geringsten Antwortzeiten. Die Werte von Unicast und Anycast sind weiterhin gleich. Bei Unicast ist auf allen getesteten Resolvern standardmäßig ein Timeout größer gleich eine Sekunde konfiguriert. Bei Anycast werden alle Pakete noch an den primären DNS-Server geroutet. Daher wird sowohl bei Unicast, als auch bei Anycast nur der primäre DNS-Server angefragt.

Steigt die Latenz auf dem DNS-Server 1 insgesamt auf zwei Sekunden, ergeben sich folgende Antwortzeiten:

	Unicast	Multicast	Anycast
Ubuntu Client	2000ms	6ms	2000ms
Windows Client	1000ms	6ms	2000ms
Mac OS X Client	2000ms	6ms	2000ms
Windows Phone	2000ms	6ms	2000ms
Android	2000ms	6ms	2000ms
iOS	2000ms	6ms	2000ms

Tabelle 13: Antwortzeiten bei einer Latenz von zwei Sekunden

Da der Timeout des Resolvers von Windows bei einer Sekunde liegt, fragt dieser nach einer Sekunde den sekundären Server an. Alle anderen warten länger auf eine mögliche Antwort. Daher wird nicht der sekundäre Server angefragt und die Antwortzeit entspricht der Latenz des primären Servers. Bei Anycast bleibt die Antwortzeit für alle Clients bei zwei Sekunden, da die Resolver dort nur über die Anycast-Adresse der DNS-Server verfügen. Der Client kann somit keinen sekundären Server anfragen und das Routing-System wird die Pakete weiterhin an den primären DNS-Server senden, da dieser grundsätzlich noch erreichbar ist.

5.3.3 Response-Time bei Ausfall des DNS-Servers – Szenario 3.c

Ist der primäre DNS-Server in dem Moment des DNS-Querys ausgefallen, ergeben sich die nachfolgenden Antwortzeiten:

	Unicast	Multicast	Anycast
Ubuntu Client	5000ms	6ms	20000ms
Windows Client	1000ms	6ms	12000ms
Mac OS X Client	5000ms	6ms	20000ms
Windows Phone	3000ms	6ms	22000ms
Android	5000ms	6ms	20000ms
iOS	2000ms	6ms	7000ms

Tabelle 14: Antwortzeiten bei Ausfall des primären DNS-Servers (Teil 1)

Bei Unicast entspricht die Antwortzeit der Zeitspanne, bis der Resolver den sekundären DNS-Server anfragt. Da bei Anycast der Ausfall des Servers noch nicht erkannt wurde, werden alle Anfragen des Resolvers weiterhin an den DNS-Server 1 geleitet. Daher erhält der Resolver keine Antwort und liefert nach seiner maximalen Wartezeit von 7 bis 22 Sekunden je nach Betriebssystem einen Negativeintrag zurück. Bei Multicast beträgt die Antwortzeit hingegen nur 6 ms.

Ist der Server jedoch schon vor mindestens 40 Sekunden ausgefallen, ergeben sich folgende Werte:

	Unicast	Multicast	Anycast
Ubuntu Client	5000ms	6ms	6ms
Windows Client	1000ms	6ms	6ms
Mac OS X Client	5000ms	6ms	6ms
Windows Phone	3000ms	6ms	6ms
Android	5000ms	6ms	6ms
iOS	2000ms	6ms	6ms

Tabelle 15: Antwortzeiten bei Ausfall des primären DNS-Servers (Teil 2)

Im Fall von Unicast und Multicast bleiben die Antwortzeiten unverändert. Bei Anycast verkürzen sich die Zeiten deutlich. Nach spätestens 40 Sekunden wird durch das Ausbleiben der Hello-Pakete von OSPF der Ausfall des Servers erkannt und die Anfragen werden direkt an den sekundären Server (DNS-Server 2) geroutet.

In Tabelle 16 sind die Testergebnisse bzgl. der Response-Time als Übersicht zusammengefasst.

Response-Time	Unicast	Multicast	Anycast
Ohne Last	6ms	6ms	6ms
Mit Last (Latenz 0,5/1/2 Sek.)	500ms / 1000ms / 1000 bis 2000ms[I]	6ms	500ms / 1000ms / 2000ms
Bei Ausfall eines Servers	1000 bis 5000ms[II]	6ms	7 - 22 Sek.[III]

[I] Bei Windows Phone und iOS zwei Sek. Windows eine Sek. Alle anderen System zwei Sekunden
[II] Windows Phone drei, iOS zwei und Windows eine Sekunde. Alle anderen Systeme fünf Sekunden
[III] Je nach Betriebssystem

Tabelle 16: Ergebnisübersicht der Antwortzeiten

5.4 Änderungsaufwand

Die folgenden Ergebnisse geben den Arbeitsaufwand wieder, der bei Änderungen an der DNS-Infrastruktur entsteht. Der Aufwand wird, wie in Kapitel 4.4 beschrieben, entsprechend der folgenden Tabelle eingeordnet.

Zeit / Komplexität	< 10min	≥ 10min und < 1std	≥ 1std und < 4std	> 4std
gering	sehr niedrig	niedrig	mittel	sehr hoch
hoch	niedrig	mittel	hoch	extrem hoch

Tabelle 17 : Einordnung des Änderungsaufwandes

5.4.1 Aufwand: Ersetzen eines DNS-Servers – Szenario 4.a

Ein ausgefallener DNS-Server wird durch einen neuen Server ersetzt. Die nachfolgen Resultate geben die Aufwandsabschätzung wieder.

Unicast:
Außer der Standard-Systemkonfiguration (IP-Adressen etc.) des DNS-Servers und der typischen Konfiguration des DNS-Dienstes sind keine weiteren Maßnahmen durchzuführen. Der Aufwand für das Ersetzen eines DNS-Servers kann als *niedrig* eingestuft werden.

Multicast:
Der Aufwand beim Ersetzen eines DNS-Servers im Multicast-Szenario ist höher als im Unicast-Szenario. Hier kommt zusätzlich ein Multicast-Proxy zum Einsatz, der die DNS-Multicast-Pakete an den DNS-Dienst weiterleitet (siehe dazu die Multicast-Konfiguration in Kapitel 3.3). Die Bereitstellung und Konfiguration des Proxys benötigt zusätzliche Zeit. Zudem ist eine angepasste Konfiguration des DNS-Dienstes zu beachten. Der Aufwand wird auf *mittel* festgelegt.

Anycast:
Im Anycast-Szenario muss eine Reihe von Konfigurationsschritten durchlaufen werden, um einen DNS-Server zu ersetzen. Der Aufwand ist in diesem Fall *hoch*. So muss eine weitere IPv6-Adresse an das Loopback-Interface gebunden und die Konfiguration des DNS-Dienstes an diese IPv6-Adresse angepasst werden. Zudem sind die Installation und die Konfiguration des Routing-Dienstes (Quagga) sowie die anschließende Überprüfung der dynamisch verteilten Routen notwendig.

5.4.2 Aufwand: Änderung der DNS-Server IP-Adresse – Szenario 4.b

Welcher Aufwand entsteht, wenn der DNS-Server eine andere IPv6-Adresse erhält, ist in diesem Kapitel dargestellt.

Unicast:
Auf dem Server selbst muss im Betriebssystem und in der Konfiguration des DNS-Dienstes die neue IPv6-Adresse eingetragen werden. Für die automatische Verteilung der neuen DNS-Server-Adressen an die Clients ist zudem eine Anpassung in den DHCPv6- bzw. Router-Advertisement-Diensten notwendig. Der Aufwand ist *niedrig*.

Multicast:
Ändert sich nur die globale IPv6-Adresse der DNS-Server, ist der Aufwand *sehr gering*. Die globale IPv6-Adresse der Server kann geändert werden, ohne dass es Auswirkungen auf die Multicast-Adresse hat. Die beiden Adressen sind komplett unabhängig voneinander. Daher muss die neue IPv6-Adresse der Server den Clients nicht mitgeteilt werden. Die Clients erreichen die DNS-Server weiterhin über die IPv6-Multicast-Adresse FF08::114. Auf dem DNS-Server selbst muss die neue IPv6-Adresse nur in der Systemkonfiguration angepasst werden.

Anycast:
Ähnlich verhält es sich bei Anycast: Die IPv6-Adresse, die für den DNS-Dienst verwendet wird, ist unabhängig von der eigentlichen Server IPv6-Adresse. Daher kann der DNS-Server mit einem *sehr geringen* Aufwand eine neue IP-Adresse erhalten.

5.4.3 Aufwand: Erweiterung um einen DNS-Server – Szenario 4.c

In diesem Szenario wird die Umgebung um einen DNS-Server erweitert und der dadurch entstandene Aufwand abgeschätzt.

Unicast:
Abgesehen von den Aufgaben für die Inbetriebnahme eines neuen DNS-Servers (siehe Kapitel 5.4.1), muss die IP-Adresse des weiteren DNS-Servers noch an die Clients verteilt werden. Wird dafür eine automatische Verteilung verwendet, ist der Aufwand *gering*.

Multicast:
Bei der Erweiterung der DNS-Infrastruktur um einen Server entstehen bei Multicast keine weiteren Arbeitsschritte. Nur der in Kapitel 5.4.1 beschriebene Aufwand für die Einrichtung eines neuen Servers ist notwendig. Der neue Server ist ein weiteres Mitglied in der Multicast-Gruppe FF08::114. Diese Gruppe ist allen Clients schon bekannt, wodurch keine Änderungen vorgenommen werden müssen. Der Aufwand ist somit unverändert *mittel*.

Anycast:
Neben dem in Kapitel 5.4.1 ermittelten Aufwand fallen keine weiteren Tätigkeiten an, da den Clients die Anycast-Adresse aller DNS-Server schon bekannt ist. Der Aufwand bleibt daher *hoch*, aber steigt nicht weiter.

Die nachfolgende Tabelle zeigt eine Übersicht über die Testergebnisse hinsichtlich des Änderungsaufwandes aus diesem Kapitel.

Aufwand bei Änderungen	Unicast	Multicast	Anycast
Ersetzen eines DNS-Servers	Niedrig	Mittel	Hoch
Ändern der IPv6-Adresse	Niedrig	Sehr niedrig	Sehr niedrig
Erweitern um einen Server	Niedrig	Mittel	Hoch

Tabelle 18: Testergebnisse bzgl. des Aufwands bei Änderungen

5.5 Netzlast – Szenario 5.a

Die nachfolgende Analyse gibt die Anzahl der übertragenen Pakete auf Router 1 bei einer Namensauflösung wieder. Die Ermittlung findet unter der Voraussetzung statt, dass alle Netzkomponenten einander bekannt sind.

Unicast:
Bei einer Unicast-Kommunikation entstehen zwei zu übertragene Pakete (Query und Response) für eine Namensauflösung. Bei angenommenen 1000 DNS-Querys von je 512 Byte ergeben sich für Router 1 insgesamt 2000 zu übertragene Pakete und eine Datenmenge von 1 MByte. Wird die DNS-Erweiterung EDNS verwendet, sind Pakete bis zu einer größer von 4096Byte möglich. Dadurch ergibt sich eine maximale Datenmenge von 8 MByte für Router 1.

Multicast:
Durch die Verwendung von Multicast wird der DNS-Server 2 bei jedem DNS-Query mitangefragt. Über das Core-Netz und das Client-Netz werden daher zwei Antwortpakete übertragen: Ein Paket von DNS-Server 1 und ein weiteres von DNS-Server 2. Im Mgmt-Netz 2 werden sowohl ein Query-, als auch ein Response-Paket übertragen. Bei Unicast werden dort zum Vergleich keine Pakete übertragen. Im Mgmt-Netz 1 bleibt es bei jeweils einem Query- und einem Response-Paket. Bei angenommenen 1000 DNS-Querys von je 512 Byte ergeben sich für Router 1 insgesamt 3000 zu übertragene Pakete und eine Datenmenge von 1,5 MByte. Für jeden weiteren DNS-Server, der unter der Multicast-Adresse erreichbar ist, erhöht sich die Anzahl der Pakte um 1000 und die Datenmenge um 512 KByte. Mit EDNS ergibt sich eine maximale Datenmenge von 12 MByte für Router 1.

Beachtet werden müssen noch die Pakete, die für die Verwaltung und Organisation von Multicast-Gruppen entstehen (MLD-Pakete siehe Kapitel 2.3.3 für weitere Informationen). Diese erzeugen zwar zusätzliche Netzlast, ihre Anzahl ist aber konstant und unabhängig von der Anzahl der DNS-Querys. Bei der Erhöhung der Anzahl von DNS-Querys tritt daher keine weitere Last auf.

Anycast:
Wie im Unicast-Szenario werden auch bei Anycast nur ein Query- und ein Response-Paket übertragen. Die Netzlast hat sich im Vergleich zur Unicast-Kommunikation nicht verändert. Hinzu kommen noch die Pakete für die Verwaltung des dynamischen Routings. Die Anzahl der Pakete ist aber unabhängig von der Menge der DNS-Anfragen.

Tabelle 19 stellt zusammenfassend für jedes Kommunikationsmodell die Anzahl der übertragenen Pakete und die Datenmenge für Router 1 bei 1000 Requests dar.

	Unicast	**Multicast**	**Anycast**
1000 Requests	2000 Pakete	3000 Pakete	2000 Pakete
Max. 512 Byte	1 MByte	1,5 MByte	1 MByte
Max. 4096 Byte	8 MByte	12 MByte	8 MByte

Tabelle 19: Anzahl übertragener Pakete und Datenmenge

5.6 Netzmanagement

In den folgenden Unterkapiteln sind die Ergebnisse der einzelnen Managementanforderungen an eine DNS-Umgebung dargestellt.

5.6.1 Netzmanagement (Kontrolle und Konfiguration) – Szenario 6.a

In diesem Abschnitt wird geprüft, ob sich bestimmte Kontroll- und Konfigurationsfunktionen wie beispielsweise das Neustarten eines Servers oder das Bearbeiten von DNS-Records durchführen lassen.

Unicast:
Jeder Server ist über seine IPv6-Adresse eindeutig identifizierbar. Dies ermöglicht ein zentrales Management jedes einzelnen Servers, um beispielsweise eine Zonendatei einzulesen, den Server neuzustarten oder einen DNS-Record zu editieren.

Multicast:
Neben der IPv6-Multicast-Adresse müssen die Server in jedem Fall eine eindeutige IPv6-Adresse besitzen. Nur mittels dieser Adresse ist zentrales Management von Kontrollfunktionen auf den einzelnen Servern möglich.

Eine Management-Kommunikation über die Multicast-Adresse der Server ist keine Alternative, da dadurch keine eindeutige Zuordnung zu den einzelnen Servern gegeben ist. Ein separater Neustart der Server ist damit nicht möglich.

Anycast:
Auch in dem Anycast Szenario benötigt der Server für das Management eine eindeutige IPv6-Adresse. Werden die Server hingegen über die Anycast-Adresse angesprochen, kann eine Zuordnung zu einem der Server nicht garantiert werden.

5.6.2 Netzmanagement (Überwachung und Alarm) – Szenario 6.b

Geprüft wird hier, ob und wie eine Überwachung der einzelnen DNS-Server und des DNS-Dienstes im Allgemeinen möglich ist. Zudem wird analysiert, wie gut je nach Kommunikationsmodell auf Fehler reagiert werden kann.

Unicast:
Von einem zentralen Punkt im Netz aus ist es möglich, die einzelnen DNS-Server mittels ihrer IPv6-Unicast-Adresse zu überwachen. Somit können von jedem Server diverse Performance-Counter bzgl. des DNS-Zustandes ermittelt werden. Daraus ist z. B. die aktuelle Auslastung (Anzahl an Requests) jedes DNS-Servers ersichtlich.

Treten Fehler bei der Namensauflösung auf, ist deren Analyse unkompliziert. In der Regel wird vom Client immer der primäre DNS-Server angefragt, unabhängig vom Standort des Clients im Netz. Erhält ein Client beispielsweise falsche Daten auf seine DNS-Anfrage oder nur eine verzögerte Antwort, lässt sich die Ursache leicht ermitteln.

Multicast:
Über die Multicast-Adresse ist nur eine Überwachung des DNS-Dienstes selbst möglich. Performance-Counter eines einzelnen DNS-Servers lassen sich damit nicht ermitteln. Hierfür muss jeder Server über seine eindeutige globale IPv6-Adresse überwacht werden. Dann ist ein zentrales Management der einzelnen DNS-Server möglich.

Bei Multicast kann eine genaue Fehlersuche problematisch werden. Es werden grundsätzlich beide DNS-Server gleichzeitig angefragt. Daher ist nicht immer eindeutig, welcher Server im konkreten Fall zuerst antwortet.

Anycast:
Die Überwachung ist bei Anycast deutlich komplexer als bei Unicast, da es sich hierbei um einen verteilten Dienst unter einer IP-Adresse handelt. Die Verfügbarkeit des Dienstes ist im Allgemeinen abhängig vom Standpunkt des Clients. Daher ist es notwendig, den Dienst von verschiedenen Punkten im Netz aus zu monitoren. Zudem ist es bei einer komplexeren Umgebung sinnvoll, das Routing zu überwachen.

Durch das dynamische Routing ist nicht immer fest definiert, an welchen Server die DNS-Anfragen geroutet werden. Dies hängt unter anderem vom Zustand des Netzes und vom Standpunkt des Clients ab. Tritt ein Problem auf, ist es bei Anycast schwieriger, die Fehlerursache zu analysieren. Durch das dynamische Routing ist nicht immer eindeutig, mit welchem DNS-Server kommuniziert wurde und wer den Fehler verursacht hat. Somit ist auch eine Rekonstruktion durch den Administrator komplexer. Es könnten sich inzwischen die Gegebenheiten geändert haben. Außerdem kann sich der Testrechner für die Rekonstruktion in einem anderen Netzsegment befinden. Dann ist nicht sichergestellt, dass gegen den gleichen DNS-Server getestet wird.

Zusammenfassend ergeben sich folgende Ergebnisse hinsichtlich des Netzmanagements.

Netzmanagement	**Unicast**	**Multicast**	**Anycast**
Kontrolle, Konfiguration	Möglich	Per globaler IPv6-Adresse möglich	Per globaler IPv6-Adresse möglich
Überwachung, Alarm	Normal	Erschwert	Erschwert

Tabelle 20: Testergebnisse bzgl. Netzmanagement

5.7 Sicherheit

In den nachfolgenden Abschnitten werden die Ergebnisse zu den einzelnen Tests (DoS-Angriff, DNS-Hijacking, DNSSec etc.) bezüglich der Sicherheit dargestellt.

5.7.1 Sicherheit bei Denial-of-Service-Angriff – Szenario 7.a

Die Anfälligkeit des DNS-Dienstes für Denial-of-Service-Angriffe (DoS-Angriffe) wird in diesem Kapitel für jedes Kommunikationsmodell dargestellt.

Unicast:
Für die wirkungsvolle Beeinträchtigung eines DNS-Dienstes, der aus mehreren Servern besteht und unter unterschiedlichen IPv6-Adressen per Unicast zu erreichen ist, muss der Angreifer jeden Server mit Anfragen überfluten. Dazu benötigt der Angreifer eine ausreichend schnelle Anbindung an jeden der Server. Zudem muss er genügend Ressourcen bereitstellen, um so viele Anfragen zu erzeugen, dass die Server nicht mehr erreichbar sind.

Dass solche Angriffe auf den DNS-Dienst kein Problem darstellen, zeigen diverse Meldungen [Ermert, 2006], [Mills, 2012], [Ihlenfeld, 2008]. In der Testumgebung ist ein DoS-Angriff mithilfe von zwei Nodes produzierbar. Durch je ca. 500 Anfragen pro Sekunde ist es möglich, auf dem DNS-Server eine Überlast zu erzeugen. Eine Antwort der Server erfolgt nicht mehr innerhalb von 22 Sekunden, wodurch die Resolver einen Negativeintrag zurückliefern.

Multicast:
Bei dem Einsatz von Multicast besteht das Problem, dass die DNS-Server einfacher mit großen Mengen von Anfragen überflutet werden können. Der Grund hierfür liegt darin, dass die Multicast-Pakete von den Routern an alle DNS-Server geschickt werden.

Angenommen die DNS-Server hätten eine maximale Verarbeitungskapazität von je 1.000 Requests pro Sekunde, bevor sie nicht mehr zeitnah antworten. Dann müsste der Angreifer im Unicast-Szenario 1.000 Requests/s an den DNS-Server 1 und weitere 1.000 Requests/s an den DNS-Server 2 senden. Er müsste also insgesamt 2.000 Requests erzeugen. Bei Multicast wären nur 1.000 Requests/s vom Angreifer notwendig, da die Router 2 und 3 die Requests an Server 1 und Server 2 leiten, also "verdoppeln". Dadurch muss der Angreifer weniger Ressourcen bereitstellen, um ausreichend Requests zu erzeugen. Dies übernimmt in diesem Fall der Router für ihn (siehe nachfolgende Abbildung).

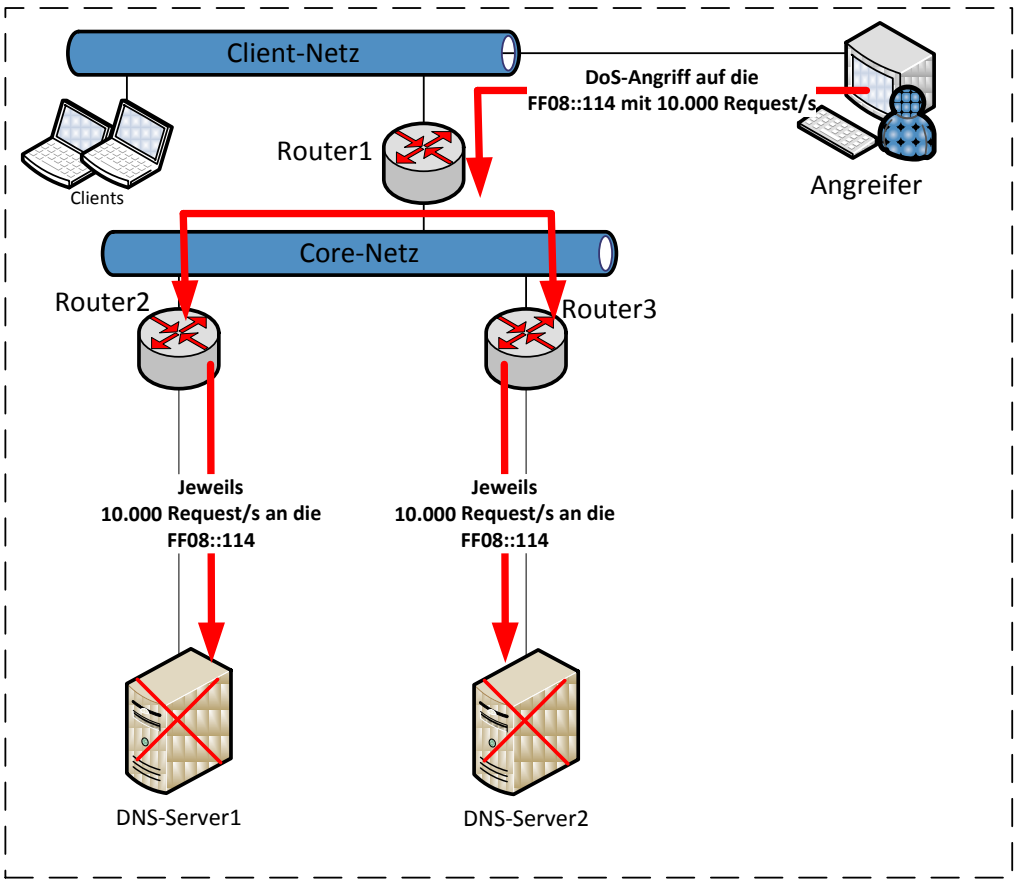

Abbildung 12: DoS-Angriff bei Multicast

Möglicherweise sind dem Angreifer auch nicht alle DNS-Server bekannt, wodurch er einen Server bei seinem Angriff auslassen könnte. Im Multicast-Szenario muss er nur die IPv6-Multicast-Adresse für den DNS-Service kennen. Die Verteilung der DNS-Requests auf alle DNS-Server übernehmen die Router für ihn.

Anycast:
Bei dem Anycast-Szenario ist nur die IPv6-Anycast-Adresse der DNS-Server bekannt. Dem Angreifer ist es nicht möglich, beide Server gleichzeitig mit Anfragen zu überfluten. Die DNS-Requests werden immer an den aus Sicht des Angreifers nächsten DNS-Server geleitet. Angenommen dies sei in diesem Fall der DNS-Server 2, so wäre nur dieser von dem DoS-Angriff betroffen. DNS-Server 1 würde ohne Beeinträchtigung weiterarbeiten. Clients aus dem Client-Netz wären also nicht betroffen und könnten weiterhin Namensauflösungen durchführen. Die Anycast-Kommunikation verhindert, dass nicht der ganze DNS-Service ausfällt (siehe Abbildung 13).

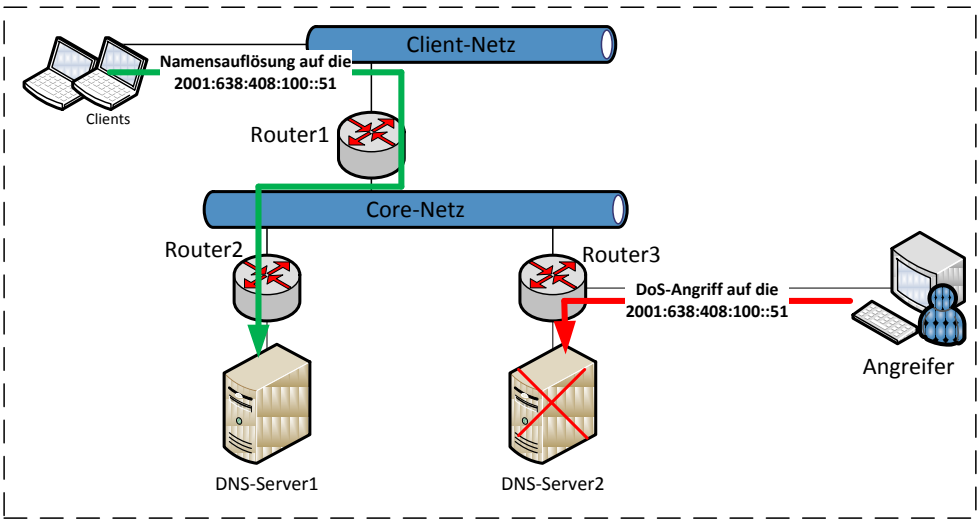

Abbildung 13: DoS-Angriff bei Anycast

5.7.2 Sicherheit bei DNS-Hijacking – Szenario 7.b

DNS-Hijacking bezeichnet das Umleiten des DNS-Verkehrs über den Angreifer. Dargestellt wird, ob und wie ein solcher Angriff in jedem Kommunikationsmodell möglich ist.

Unicast:
Bei Unicast bestehen diverse Möglichkeiten, DNS-Anfragen über das System des Angreifers zu leiten. So kann durch Angriffe auf den Client der DNS-Server des Angreifers eingetragen werden. Eine weitere Möglichkeit ist das Fälschen des Neighbor-Advertisements, ähnlich wie beim ARP-Spoofing aus IPv4.

Multicast:
Bei Multicast kommt noch folgende, für den Angreifer unkomplizierte Möglichkeit hinzu: Durch das Konzept von Multicast kann jeder, der der Gruppe FF08::114 beitritt, die DNS-Requests empfangen. Tritt also ein Client aus dem Client-Netz dieser Gruppe bei, kann er alle Requests der anderen Clients in diesem Netz abhören und aufzeichnen. Dies ermöglicht es unter anderem, einzelne Clients auszuspionieren. Diese Informationen können für weitere Angriffe, Erpressungen oder Social Engineering verwendet werden.

Anycast:
Da im Anycast-Szenario ein dynamisches Routing-Protokoll verwendet wird, kann durch gefälschte OSPF-Pakete der Datenverkehr umgeleitet werden. Datenverkehr an die IPv6-Anycast-Adresse kann damit über den Server des Angreifers geleitet werden, indem der Angreifer angibt die kürzeste Route zum Ziel zu besitzen. Ansonsten sind für Anycast keine Änderungen gegenüber Unicast hinsichtlich DNS-Hijacking zu erkennen.

5.7.3 Sicherheit bei DNS-Spoofing – Szenario 7.c

DNS-Spoofing bezeichnet das Fälschen von DNS-Zuordnungen. In diesem Kapitel sind die entsprechenden Ergebnisse zu DNS-Spoofing je nach Kommunikationsmodell dargestellt.

Unicast:
Für das Fälschen einer DNS-Antwort benötigt der Angreifer die ID des DNS-Querys, die IPv6-Adresse des Clients und den Quell-Port. Diese Informationen erhält er am einfachsten, wenn es ihm gelingt, den DNS-Verkehr mitzuschneiden. Wie im vorherigen Kapitel erwähnt, existieren für das Umleiten des Datenverkehrs einige Möglichkeiten. Andernfalls besteht für den Angreifer nur die Option, diese Informationen zu erraten. Da für die ID und den Source-Port jeweils 16Bit vorgesehen sind, ergibt sich hierfür eine Wahrscheinlichkeit von 1 zu 4,2 Milliarden.

Multicast:
Bei Multicast besteht das gleiche Problem wie schon bei DNS-Hijacking: Jeder Rechner kann sich als DNS-Server ausgeben. Daher werden die DNS-Requests auch an den Rechner des Angreifers geschickt. Dieser kann daraufhin eine gefälschte Antwort an den anfragenden Client schicken. Er besitzt alle Informationen (ID des Requests, Source Port und IP-Adresse des Clients), um ein Antwort-Paket zu fälschen. Befindet sich der Angreifer im gleichen Netz wie der anfragende Client, wird mit hoher Wahrscheinlichkeit sein manipuliertes Antwort-Paket vor der Antwort der richtigen DNS-Server ankommen. Daher wird das gefälschte Paket des Angreifers vom Client angenommen und die Pakete von den DNS-Servern verworfen. Der Angreifer hat dem Client eine manipulierte Zuordnung von Name und IP-Adresse untergeschoben. Weiteren Datenverkehr, der diese Zuordnung benutzt, kann der Angreifer anschließend über seinen Rechner leiten.

Anycast:
Wie im vorigen Kapitel sind bei Anycast keine weiteren Schwachstellen zu erkennen, die nicht schon bei Unicast existieren. Durch die Verwendung von OSPF ist aber auch hier eine Umleitung des Verkehrs durch die Verwendung von gefälschten OSPF-Paketen möglich.

5.7.4 Sicherheit bei DNS Information Leakage – Szenario 7.d

In diesem Abschnitt wird anhand der Testergebnisse beschrieben, inwieweit es möglich ist, an Informationen des DNS-Servers zu gelangen.

Unicast:
Bei einer Unicast-Kommunikation zwischen Resolver und DNS-Server benötigt der Angreifer Zugriff auf die Netzkomponenten, damit er Informationen abgreifen kann. Dies ist beispielsweise durch das Umleiten des DNS-Datenverkehrs oder durch die Einrichtung von Mirror-Ports auf den Switchen möglich. Andern-

falls bleibt dem Angreifer nur das Kompromittieren des DNS-Servers, um an DNS-Informationen zu gelangen.

Multicast:
Da die Clients alle ihre Anfragen zur Namensauflösung an eine Multicast-Adresse schicken, ist es für einen Angreifer sehr einfach, Informationen zu sammeln. Er muss nur der entsprechenden Multicast-Gruppe beitreten. Dann erfährt er anhand der diversen Namensauflösungen der Clients, z. B. die IP-Adresse und den Hostname des Mailservers des Unternehmens.

Anycast:
Für Anycast existieren die gleichen Angriffsmethoden wie bei Unicast. Außerdem besteht die Möglichkeit, durch gefälschte OPSF-Pakete den Datenverkehr umzuleiten.

5.7.5 Sicherheit: Einsatz von DNSSec – Szenario 7.e

In diesem Abschnitt werden die Ergebnisse der Frage dargestellt, ob jedes Kommunikationsmodell den Einsatz von DNSSec ermöglicht.

Wie in Kapitel 2.4.2 beschrieben, basiert DNSSec weiterhin auf einer UDP-Kommunikation und auf der Abfrage von Records beim DNS-Server. Im Vergleich zu einem "normalen" DNS-Query bestehen nur zwei Unterschiede (siehe Abbildung 14):

1. Zusätzlich zum angefragten Record wird noch die entsprechende Signatur (RRSIG) im gleichen Paket mitgeschickt.
2. Mithilfe eines weiteren Querys wird der öffentliche Schlüssel (DNSKEY) des Servers angefragt.

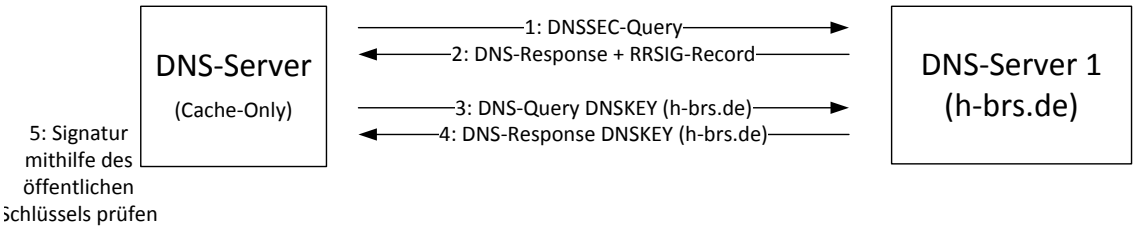

Abbildung 14: DNSSec-Kommunikation (Ausschnitt)

Unicast:
Da DNSSec für eine Unicast-Kommunikation designt wurde, ist es im Unicast-Szenario zum Schutz der Integrität und Authentizität voll einsetzbar.

Anycast:
Die oben beschriebenen Unterschiede zum "normalen" DNS schränken eine Verwendung von DNSSec mit Anycast nicht ein. Beide DNS-Server besitzen zudem den gleichen Datenbestand, welcher mit demselben privaten Schlüssel signiert wurde. Daher ist es sogar unproblematisch, wenn der Resolver die Sig-

natur (RRSIG) von DNS-Server 1 und den öffentlichen Schlüssel (DNSKEY) von DNS-Server 2 erhält. Die Routing-Pfade und Ziele der DNS-Pakete (Schritt 1 und 3) können sich daher ohne Einschränkung der DNSSec-Funktionalität jederzeit ändern (siehe Abbildung 15).

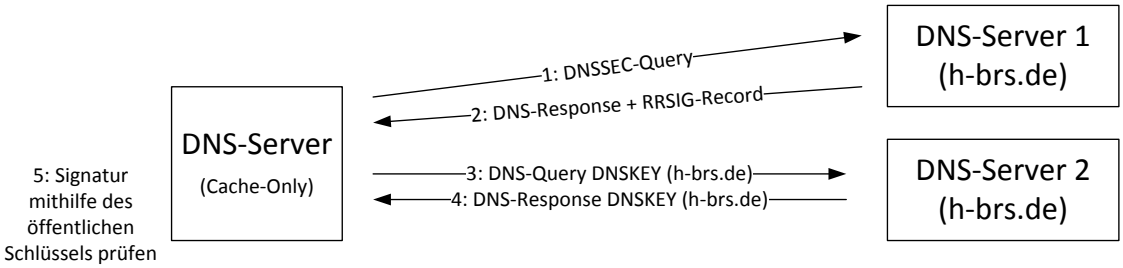

Abbildung 15: Mögliche DNSSec-Kommunikation mit Anycast

Multicast:
Im Multicast-Szenario ist die Verwendung von DNSSec ebenfalls möglich. Wie schon im Anycast-Szenario beschrieben, sind die beiden Anfragen (Schritt 1 und 3) unabhängig voneinander zu betrachten. Es ist unerheblich, von welchem DNS-Server sie beantwortet werden.

Bei Multicast werden beide DNS-Server gleichzeitig angefragt. Es ist irrelevant, von welchem Server die Pakete beim Resolver zuerst ankommen. So verwendet der Resolver im nachfolgenden Beispiel (Abbildung 16) das Antwortpaket 2 von DNS-Server 1 und das Antwortpaket 4 von DNS-Server 2 und kann dennoch die Signatur erfolgreich prüfen.

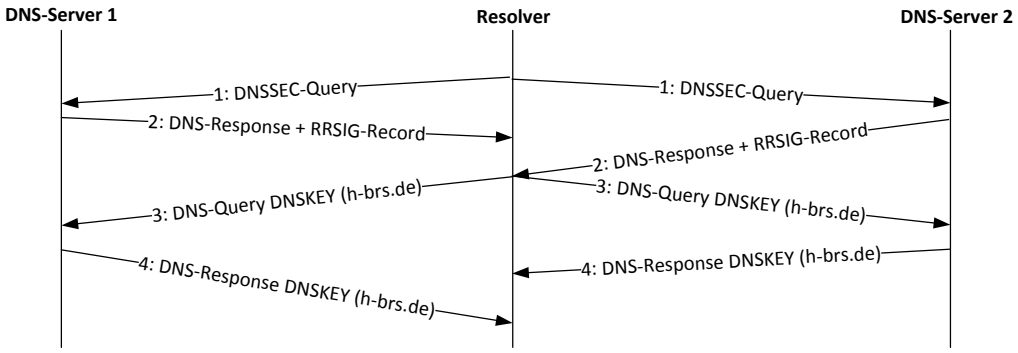

Abbildung 16: Mögliche DNSSec-Kommunikation mit Multicast

Tabelle 21 zeigt eine Zusammenfassung der Testergebnisse zum Thema Sicherheit. Der Einsatz von DNSSec ist mit allen Kommunikationsmodellen möglich. Die Verwendung von Multicast macht die DNS-Kommunikation für Angriffe anfälliger. Anycast erschwert DoS-Angriffe.

Sicherheit	Unicast	Multicast	Anycast
DoS-Angriff	Anfällig	Hochanfällig	Weniger anfällig
DNS-Hijacking	Anfällig	Hochanfällig	Anfällig
DNS-Spoofing	Anfällig	Hochanfällig	Anfällig
DNS Information Leakage	Anfällig	Hochanfällig	Anfällig
Einsatz von DNSSec	Möglich	Möglich	Möglich

Tabelle 21: Testergebnisse bzgl. Sicherheit

5.8 Aktualität der Antworten – Szenario 8.a

Die Ergebnisse in diesem Kapitel zeigen, wie lange ein Client im ungünstigsten Fall auf einen aktualisierten Record bei einer Namensauflösung warten muss. Die Resultate sind aus Tabelle 22 ersichtlich.

	Unicast	Multicast	Anycast
Ohne Notify-Update	0 Minuten	Bis zu 60 Minuten	Bis zu 60 Minuten
Mit Notify-Update	0 Minuten	0 Minuten	0 Minuten

Tabelle 22: Maximale Wartezeit auf einen aktualisierten Record

Normalerweise wird die Änderung eines DNS-Records auf dem primären DNS-Server durchgeführt. Bei Anycast und Multicast existiert aber ein solcher Server in der klassischen Form nicht mehr. Das bedeutet, dass nicht immer zuerst vom Resolver versucht wird, den primären Server anzufragen. Im Fall von Anycast liegt das daran, dass abhängig vom Zustand des Netzes und der Position des Clients die DNS-Anfrage an den nächstgelegenen DNS-Server geleitet wird. Bei Multicast werden jederzeit beide DNS-Server angefragt und das Antwort-Paket vom Resolver verwendet, das beim Client zuerst eintrifft. Dies kann von Anfrage zu Anfrage variieren.

Standardmäßig ist bei Bind die Funktion Notify-Update eingeschaltet. Daher erhalten alle DNS-Server eine Benachrichtigung über eine Änderung und beziehen diese direkt. Ist die Funktion allerdings abgeschaltet oder nicht in der DNS-Software vorhanden, kann sich die Aktualisierung verzögern. Bei einer zyklischen Übertragung (ohne Notify-Update) der Zonendatei erhalten die anderen DNS-Server erst nach bis zu 60 Minuten den neuen Eintrag für mail.test-dns.netlab.inf.h-brs.de. Die Clients bei Anycast und Multicast fragen im ungünstigsten Fall immer den sekundären Server an bzw. dessen Antwort-Paket trifft am schnellsten bei ihnen ein. Daher empfangen sie noch bis zu 60 Minuten die "alte" IPv6-Adresse für den Domain-Name mail.test-dns.netlab.inf.h-brs.de.

6 Auswertung der Ergebnisse

In diesem Kapitel werden die Ergebnisse aus dem vorigen Kapitel ausgewertet und anschließend in Tabelle 23 zusammenfassend dargestellt.

6.1 Unicast

Da DNS für eine Unicast-Kommunikation konzipiert ist, existieren für das Unicast-Szenario keine Anforderungen, die über das Vorhandensein eines DNS-Dienstes und einer IP-Kommunikation hinausgehen. Eine Namensauflösung der verschiedenen Clients ist daher problemlos möglich.

Die Verfügbarkeit des DNS-Dienstes bei einer Namensauflösung per Unicast ist in den Tests immer gegeben. Trotz des Ausfalls diverse Komponenten (Server, Router etc.) in den einzelnen Tests kann eine Namensauflösung immer durchgeführt werden. Alle Resolver verfügen über eine Failover-Strategie. Dabei wird nach einer gewissen Zeit versucht, einen alternativen DNS-Server zu erreichen. Die Zeitspanne bis zur Anfrage eines weiteren Servers variiert aber je nach Betriebssystem von eins bis fünf Sekunden. Dementsprechend variieren auch die Antwortzeiten. Im Fehlerfall fragen einige Resolver schneller einen zweiten DNS-Server an als andere. Dadurch können Antwortzeiten von bis zu fünf Sekunden für eine Namensauflösung entstehen.

Im Gegensatz zu Anycast oder Multicast müssen bei Unicast keine weiteren Konfigurationen vorgenommen werden. Die Bereitstellung von DNS per Unicast ist daher unkompliziert und schnell möglich. Auch Änderungen an der Umgebung sind einfach durchzuführen. Managementkonzepte sowie Erfahrungen in der Fehleranalyse sind für Unicast-Umgebungen grundsätzlich vorhanden, wodurch hierbei keine neuen Probleme entstehen. Diese Einfachheit und die Verwendung einer bewährten Kommunikation sind die Vorteile der Unicast-Kommunikation.

Im Bezug auf die Sicherheit existieren diverse Angriffsmöglichkeiten für DNS. Diese sind aber nicht auf Unicast beschränkt, sondern betreffen auch die anderen Kommunikationsmodellen. Allein bei DoS-Angriffen zeigt Anycast ein besseres Ergebnis als Unicast. Zur Absicherung des DNS-Verkehrs lässt sich bei Unicast die DNS-Erweiterung DNSSec nutzen.

Aufgrund der Verwendung von mehreren DNS-Servern durch den Resolver ist der DNS-Dienst bei Unicast für die meisten Anforderungen ausreichend gerüstet. Ein Wechsel von Unicast nach Anycast oder Multicast sollte ausreichend begründet sein, da die anderen Kommunikationsmodelle wiederum einige Nachteile besitzen.

6.2 Anycast

Die Tests zeigen, dass der Betrieb einer DNS-Umgebung mittels IPv6-Anycast-Kommunikation ordnungsgemäß funktioniert. Die Nutzung von DNS über Anycast ist mit allen getesteten Clients möglich. Auf den DNS-Servern kann die IPv6-Anycast-Adresse für den Betrieb des DNS-Dienstes verwendet werden. Bei Ubuntu ist es jedoch nicht möglich, beide DNS-Server sinnvoll in einem Subnetz zu betreiben, da die Anycast-Adresse nicht als solche im Betriebssystem deklariert werden kann (siehe O-Flag).

Eine hohe Verfügbarkeit ist einer der Vorteile bei der Verwendung von Anycast [Abley, et al., 2006]. In den durchgeführten Test zeigt sich dies im Zusammenhang mit DNS jedoch nicht. DNS über IPv6-Anycast weist in einigen Tests sogar eine schlechtere Verfügbarkeit auf als Unicast oder Multicast. Dafür gibt es mehrere Gründe:

Ein Grund ist die nicht ausreichende Verknüpfung von der Verfügbarkeit des Dienstes mit der Bekanntgabe der entsprechenden Route. In einigen Tests ist die Route zum primären DNS-Server immer noch vorhanden, obwohl der DNS-Dienst nicht mehr verfügbar ist. Um diesen Fehler zu verhindern, muss der Ausfall des DNS-Servers von OSPF schneller erkannt werden. Das ist Beispielsweise mit der Verkürzung des DeadIntervals auf eine Sekunde möglich. Dann wird ein Ausfall eines Servers innerhalb einer Sekunde (gegenüber standardmäßig 40 Sekunden) erkannt und die DNS-Pakete an den sekundären DNS-Server geleitet. Diese Veränderung bewirkt aber auch, dass innerhalb einer Sekunde mehrere OSPF-Hello-Pakete versendet werden müssen, was wiederum das Netz belastet.

Da hierbei im Endeffekt aber nur die Verfügbarkeit des Servers mit dem Vorhandensein der Route verknüpft wird, bietet sich noch eine zusätzliche Möglichkeit der Überprüfung an. Mithilfe einer ständigen Überwachung des DNS-Dienstes auf den Servern könnte dessen Kopplung mit der Bekanntgabe der Route sichergestellt werden. Dazu müsste beispielsweise ein Skript auf dem DNS-Server die Verfügbarkeit des Dienstes ständig kontrollieren (mittels DNS-Query) und das Versenden der OSPF-Hello-Pakete unterbinden, wenn sie nicht gegeben ist. Dadurch wären Verfügbarkeit und niedrige Antwortzeiten sichergestellt.

Der andere Grund für die besseren Ergebnisse von Unicast und Multicast gegenüber Anycast besteht darin, dass die Resolver automatisch versuchen einen alternativen Server zu kontaktieren. Im Fall von Multicast werden bei jedem Request beide Server angefragt. Bei Unicast existiert durch die Verwendung von zwei DNS-Server-Adressen auf dem Client eine Failover-Strategie. Ohne eine solche Failover-Strategie hätte Unicast bei dem Ausfall des primären DNS-Servers eine deutlich schlechtere Verfügbarkeit.

Im Fall von Anycast sind die Antwortzeiten am höchsten (bis zu 22 Sekunden). Der Grund hierfür ist die oben bereits erwähnte nicht optimale Verknüpfung von DNS-Dienst und Route.

Durch den Einsatz des dynamischen Routing-Protokolls ist der Änderungsaufwand bei Anycast größer als bei Unicast. Denn neben der Konfiguration werden auch das anschließende Management der Umgebung und die Fehleranalyse aufwändiger. Zudem existiert bei Anycast kein primärer Server mehr im klassischen Sinne. Die Resolver können von Anfrage zu Anfrage mit einem anderen DNS-Server kommunizieren. Das verursacht neben der schwierigeren Fehleranalyse auch eine mögliche Verzögerung bei der Aktualisierung von DNS-Records (siehe Kapitel 5.8)

Die Tests zeigen im Bezug auf die Sicherheit, abgesehen von der potenziellen Manipulation des dynamischen Routing Protokolls, bei Anycast keine neuen Sicherheitsrisiken. Durch die Anbindung der DNS-Server über Anycast ist der DNS-Dienst sogar weniger anfällig für DoS-Angriffe. Viele DNS-Root-Server verwenden Anycast und sind daher vor DoS-Angriffen besser geschützt [ICANN, 2007]. Die Verwendung von DNSSec zur Absicherung des DNS-Verkehrs bzgl. Authentizität und Integrität ist bei Anycast möglich.

6.3 Multicast

Der Betrieb einer DNS-Umgebung mittels IPv6-Multicast-Kommunikation funktioniert ordnungsgemäß. Die Nutzung von DNS über Multicast ist mit allen getesteten Clients möglich. Dabei kann die Verteilung der IPv6-Adressen der DNS-Server an alle Clients per Router Advertisements bzw. DHCPv6 durchgeführt werden.

Bei der Bereitstellung des DNS-Dienstes über IPv6-Multicast muss allerdings ein Workaround durchgeführt werden. Hier besteht das Problem, dass die getesteten DNS-Dienste nicht die direkte Verwendung einer IPv6-Multicast-Adresse unterstützen. Daher muss ein Multicast-Proxy verwendet werden, der die DNS-Pakete entgegennimmt und an den DNS-Dienst weiterleitet. In einer produktiven Umgebung führt dieser Umstand zu einem erhöhten Konfigurationsaufwand und zu einer zusätzlichen Fehlerquelle. Funktioniert der Proxy aufgrund einer Fehlfunktion nicht mehr ordnungsgemäß, kann keine Namensauflösung auf dem Server durchgeführt werden. Daher ist eine direkte Verwendung von IPv6-Multicast durch die DNS-Anwendung in Zukunft sinnvoll. Auch eine Reservierung einer eigenen IPv6-Multicast-Adresse für DNS durch die IANA, wie z. B. bei NTP [IANA, 2013], würde die Verwendung von DNS per IPv6-Multicast unterstützen.

Bei den Tests bzgl. der Response-Time ist die Kommunikation mittels IPv6-Multicast diejenige mit den geringsten Antwortzeiten. Da gleichzeitig immer beide DNS-Server angefragt werden, erhalten die Resolver bei diesem Kommunikationsmodell eine Antwort nach durchschnittlich 6 ms. Durch den Einsatz des

Multicast-Proxys ist der Änderungsaufwand beim Einsatz von Multicast höher als bei Unicast. Außerdem bringt die Verwendung von Multicast eine höhere Komplexität mit sich. Dies begründet sich neben dem höheren Konfigurationsaufwand auch darin, dass das anschließende Management der Umgebung und die Fehleranalyse aufwändiger werden. Wie schon bei Anycast existiert auch bei Multicast kein primärer DNS-Server im klassischen Sinne mehr.

Dadurch, dass bei Multicast immer alle DNS-Server gleichzeitig angefragt werden, entsteht eine höhere Netzlast (50% mehr). Grundsätzlich sollte das bei DNS durch die geringe Datenmenge kein Problem darstellen. Beim Einsatz in einer großen Umgebung mit vielen Clients ist eine entsprechende Dimensionierung des Netzes erforderlich.

Der Einsatz von IPv6-Multicast in Verbindung mit DNS birgt zudem einige Sicherheitsrisiken, wie die Testergebnisse in Kapitel 5.7 zeigten. Es ist dem Angreifer unkompliziert möglich, sich durch einen Beitritt der Multicast-Gruppe FF08::114 als DNS-Server auszugeben. Allgemein ist es bei Multicast einfacher, an DNS-Informationen zu gelangen und DNS-Pakete zu fälschen. Außerdem ist eine DNS-Kommunikation per Multicast anfälliger für DoS-Angriffe. Zur Minimierung der Risiken sollte IPv6-Multicast nur in Verbindung geeigneter Sicherheitsvorkehrungen (Intrusion Detection System oder Switch-Policy) eingesetzt werden.

Daher ist die Verwendung von DNS über Multicast nur in vertrauenswürdigen Netzen sinnvoll. Ein Einsatz dieses Kommunikationsmodells für DNS im Internet ist nicht zu empfehlen. Die Verwendung von DNSSec zur Absicherung des DNS-Verkehrs bzgl. Authentizität und Integrität ist auch bei IPv6-Multicast möglich.

	Unicast	Multicast	Anycast
Systemvoraussetzungen	Erfüllt	Erfüllt[1]	Erfüllt
Verfügbarkeit			
Ausfall des primären DNS-Servers	Vorhanden	Vorhanden	Vorhanden
Server-Last	Vorhanden	Vorhanden	Teils vorhanden
Ausfall DNS-Dienst	Vorhanden	Vorhanden	Nicht vorhanden
Ausfall Router	Vorhanden	Vorhanden	Vorhanden
Response-Time			
Ohne Last	6ms	6ms	6ms
Mit Last (Latenz 0,5/1/2 Sekunden)	500ms / 1000ms / 1000 - 2000ms	6ms	500ms / 1000ms / 2000ms
Bei Ausfall eines Servers	1000 - 5000ms	6ms	7 - 22 Sekunden
Aufwand bei Änderungen			
Ersetzen eines DNS-Servers	Niedrig	Mittel	Hoch
Ändern der IPv6-Adresse	Niedrig	Sehr niedrig	Sehr niedrig
Erweiterung um einen Server	Niedrig	Mittel	Hoch
Netzlast	Standard	Erhöht + Gruppen-Mgmt-Pakete	Standard + dynamisches Routing
Netzmanagement			
Kontrolle, Konfiguration	Möglich	Per globaler IPv6-Adresse möglich	Per globaler IPv6-Adresse möglich
Überwachung, Alarme & Ereignisse	Normal	Erschwert	Erschwert
Sicherheit			
DoS-Angriff	Anfällig	Hochanfällig	Weniger anfällig
DNS-Hijacking	Anfällig	Hochanfällig	Anfällig
DNS-Spoofing	Anfällig	Hochanfällig	Anfällig
DNS Information Leakage	Anfällig	Hochanfällig	Anfällig
Einsatz von DNSSec	Möglich	Möglich	Möglich
Aktualität der Antworten			
Mit / Ohne Notifiy-Update	0 / 0 min	0 / bis 60 min	0 / bis 60 min

[1] Mittels Workaround (Multicast-Proxy) möglich

Tabelle 23: Ergebnisübersicht

6.4 Überführung der Ergebnisse auf andere Services

Nach der Evaluierung der Kriterien für DNS stellt sich die Frage, inwieweit die Ergebnisse auf andere Dienste übertragbar sind. Um eine Überführung grundsätzlich zu ermöglichen, sind die untersuchten Kriterien möglichst allgemein gehalten. Die Response-Time ist beispielsweise auch für andere Dienste von großer Bedeutung. Beim Network Time Protocol (NTP) ist eine geringe Antwortzeit besonders relevant, um die Zeitberechnung möglichst genau zu halten. Andere Kriterien, wie z. B. die Systemanforderungen für Multicast, muss letztendlich jeder Dienst unterstützen, um eine Anbindung per Multicast zu gewährleisten.

Die aus den Kriterien erarbeiteten Szenarien sind speziell für den DNS-Dienst entwickelt worden. Für eine genauere Überführung der Ergebnisse auf andere Dienste müssen daher die einzelnen Testszenarien an diese angepasst werden.

Die zuvor festgestellte Diskrepanz zwischen den Testergebnissen bezüglich DNS in dieser Arbeit und der allgemeinen Literatur, insbesondere hinsichtlich der Verfügbarkeit von Anycast, ermöglicht eine Überführung oder Verallgemeinerung der Ergebnisse nur bedingt. Zudem ist eine Übertragung auf eine TCP-Kommunikation nicht möglich, da über Multicast keine TCP-Verbindung aufgebaut werden kann. Häufig genutzte Protokolle, wie beispielsweise FTP, können daher nicht nativ mit IPv6-Multicast verwendet werden. Um dennoch FTP per Multicast zu ermöglichen, können aber Programme wie UFTP genutzt werden [Bush, 2013]. Bei diesem FTP-Dienst kommt ein eigenes FTP-Protokoll zum Einsatz, welches auf UDP basiert. Dadurch sind Übertragungen mit IPv6-Mulitcast möglich. Durch die nicht standardkonforme Implementierung von FTP ist es mit UFTP nicht möglich, mit anderen FTP-Clients oder -Servern zu kommunizieren. Alle Clients und Server, die an einer FTP-Kommunikation per IPv6-Multicast beteiligt sein sollen, benötigen somit dieses Programm.

Das Ergebnis, dass die Multicast-Kommunikation (bei DNS) nur mittels Proxy realisierbar ist, kann auch nicht auf alle UDP-Dienste übertragen werden. Beispielsweise ermöglicht der NTP-Dienst laut Standard eine direkte IPv6-Multicast-Kommunikation. Daher unterstützen die NTP-Implementierungen diese Kommunikation nativ. Obwohl sich das Ergebnis nicht auf einen anderen Dienst übertragen lässt, muss das Kriterium (Systemanforderung für Multicast) auch bei anderen Diensten für eine Multicast-Kommunikation erfüllt sein. Andere Kriterien (Verfügbarkeit, Änderungsaufwand, Sicherheit) sind ebenfalls für viele weitere Dienste/Protokolle relevant. Die Methodik und die Kriterien dieser Arbeit lassen sich daher auch für andere Dienste verwenden.

Für die Überprüfung von anderen Protokollen ist es zunächst sinnvoll, die Kriterien zu ermitteln. Dabei können die in dieser Arbeit festgelegten Kriterien übernommen werden. Gegebenenfalls können sie noch angepasst und durch protokollspezifische Kriterien erweitert werden. Durch die Erstellung der Testszenarien erfolgt ihre Abstimmung auf das zu untersuchende Protokoll. Die Testergeb-

nisse ermöglichen dadurch klare Aussagen über die Auswirkungen der Verwendung von Anycast, Multicast und Unicast auf den Dienst. Der Vergleich der Ergebnisse zwischen den Kommunikationsmodellen ermöglicht eine Übersicht über die Vor- und Nachteile sowie über die speziellen Eigenschaften der Kommunikationsmodelle. Für die Praxis kann anschließend abgewogen werden, ob die Vorteile eines Kommunikationsmodells überwiegen und ob sich der Wechsel zu dem Modell lohnt.

7 Zusammenfassung und Ausblick

Der Einsatz von IPv6-Anycast und IPv6-Multicast in Verbindung mit DNS ist grundsätzlich möglich. Die Umsetzung von DNS per IPv6-Anycast ist unproblematisch. Bei der Verwendung von IPv6-Multicast ist ein Workaround in Form eines Multicast-Proxys notwendig. Für eine zukünftige unkompliziertere Anwendung von DNS per Multicast ist eine von der IANA festdefinierte IPv6-Multicast-Adresse für DNS sinnvoll. Sie ermöglicht DNS-Implementierungen eine native Unterstützung von Multicast, wodurch kein Multicast-Proxy mehr notwendig ist.

Die Nutzung von IPv6-Multicast bringt Vorteile bei der Response-Time und der Verfügbarkeit des DNS-Dienstes mit sich. Bei Multicast werden immer mehrere Server gleichzeitig angefragt. Dadurch wird in vielen Testszenarien eine geringere Antwortzeit als bei Anycast oder Unicast erreicht.

Bei Anycast hingegen besteht die Schwierigkeit, den Ausfall einer Teilkomponente des DNS-Dienstes rechtzeitig zu erkennen. Das kann beispielsweise der Ausfall eines Routers oder eines DNS-Dienstes eines der Server sein. Nur dadurch können im Fehlerfall eine geringe Antwortzeit und eine hohe Verfügbarkeit garantiert werden. Mithilfe eines dynamischen Routing-Dienstes, der auch auf dem DNS-Server installiert wird, kann die Erkennung eines Ausfalls für einige aber nicht alle Testszenarien realisiert werden.

Da es auf allen Resolvern möglich ist, mehrere DNS-Server einzutragen, ist die Verfügbarkeit auch bei Unicast immer gegeben. Alle Resolver verfügen über eine Strategie, um bei Nichterreichen des primären Servers weitere Server zu kontaktieren. Allein die Response-Time schwankt bei den verschiedenen Clients, da ihre Timeouts variieren.

Allgemein bewirken Anycast und Multicast eine höhere Komplexität der Umgebung, was sich insbesondere beim Management und bei der Konfiguration der DNS-Infrastruktur bemerkbar macht. Zudem birgt der Einsatz von IPv6-Multicast neue Sicherheitsrisiken. Diese Risiken beschränken sich zwar nicht nur auf DNS, müssen aber beherrscht werden, um eine sichere Umgebung zu gewährleisten.

Durch die Nachteile ist eine grundsätzliche Empfehlung für den produktiven Einsatz schwierig. Für eine vorhandene oder geplante Umgebung sollte individuell analysiert werden, ob die Vorteile von Anycast oder Multicast gegenüber Unicast überwiegen. In welcher konkreten Umgebung (Anzahl von Clients, Kommunikationsmedium, Serverteilung etc.) dies der Fall ist, kann mithilfe weiterer Untersuchungen überprüft werden.

Die in dieser Arbeit ermittelten Kriterien sind auch für andere Dienste sinnvoll und geben einen Überblick, welche Anforderungen zu erfüllen sind und welche grundsätzlichen Vorteile oder Nachteile die Verwendung von Anycast oder

Multicast mit sich bringt. Die einzelnen Ergebnisse sind aber letztendlich aufgrund der Testszenarien individuell auf DNS abgestimmt. Die Ergebnisse lassen sich nicht exakt auf einen anderen Dienst anwenden, sondern müssen einzeln geprüft werden. Dazu können aus den Kriterien eigene Szenarien für den zu untersuchenden Dienst erstellt und anschließend getestet werden. Mithilfe dieser Arbeit kann die Verwendung von Anycast oder Multicast bei anderen Diensten daher leichter überprüft werden. Gegebenenfalls lassen sich durch eine Vielzahl von Untersuchungen Gemeinsamkeiten in den Ergebnissen finden, wodurch eine einfache und exakte Überführung der Ergebnisse auf andere Dienste möglich ist.

8 Literaturverzeichnis

Abley, J. und Lindqvist, K. 2006. Operation of Anycast Services. *RFC 4786.* s.l. : Internet Engineering Task Force, 2006.

Arends, R., et al. 2005. DNS Security Introduction and Requirements. *RFC 4033.* s.l. : Internet Engineering Task Force, 2005.

Arends, R., et al. 2005. Protocol Modifications for the DNS Security Extensions. *RFC 4035.* s.l. : Internet Engineering Task Force, 2005.

Arends, R., et al. 2005. Resource Records for the DNS Security Extensions. *RFC 4034.* s.l. : Internet Engineering Task Force, 2005.

BSI. 2012. IT-Grundschutz-Kataloge, 12. Ergänzungslieferung. [Online] September 2012. [Zitat vom: 20. März 2013.] https://gsb.download.bva.bund.de/BSI/ITGSK12EL/IT-Grundschutz-Kataloge-12-EL.pdf.

Bush, D. 2013. UFTP - Encrypted UDP based FTP with multicast. [Online] 27. April 2013. [Zitat vom: 1. Juni 2013.] http://www.tcnj.edu/~bush/uftp.html.

Cheshire, S. und Krochmal, M. 2001. Multicast DNS: draft-cheshire-dnsext-multicastdns-15. *Internet-Draft.* s.l. : Internet Engineering Task Force, 2001.

Coltun, R., et al. 2008. OSPF for IPv6. *RFC 5340.* s.l. : The Internet Engineering Task Force, 2008.

Crawford, M. 1998. Transmission of IPv6 Packets over Ethernet Networks. *RFC 2464.* s.l. : Internet Engineering Task Force, 1998.

DATACOM Buchverlag GmbH. Antwortzeit. [Online] [Zitat vom: 22. April 2013.] http://www.itwissen.info/definition/lexikon/Antwortzeit-response-time.html.

Davies, J. 2012. *Understanding IPv6: Covers Windows 8 and Windows 2012 Server.* 3. Auflage. Sebastopol : O'Reilly, 2012. 978-0-7356-5914-8.

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

Deering, S. und Hinden, R. 2006. IP Version 6 Addressing Architecture. *RFC 4291.* s.l. : Internet Engineering Task Force, 2006.

Deering, S. und Johnso, D. 1999. Reserved IPv6 Subnet Anycast Addresses. *RFC 2526.* s.l. : Internet Engineering Task Force, 1999.

Deering, S., Fenner, W und Haberman, B. 1999. Multicast Listener Discovery (MLD) for IPv6. *RFC 2710.* s.l. : Network Working Group, 1999.

Droms, R. 2003. DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6). *RFC 3646.* s.l. : Internet Engineering Task Force, 2003.

Durand, Jerome. 2004. M6Bone - IPv6 multicast network. [Online] 2004. [Zitat vom: 14. März 2013.] http://www.m6bone.net.

Elz, R. und Bush, R. 1997. Clarifications to the DNS Specification. *RFC 2181.* s.l. : Internet Engineering Task Force, 1997.

Ermert, M. 2006. DoS-Angriffe beunruhigen Betreiber von DNS-Nameservern. [Online] 6. April 2006. [Zitat vom: 22. April 2013.] http://heise.de/-115677.

Gilligan, R., et al. 2003. Basic Socket Interface Extensions for IPv6. *RFC 3493.* s.l. : Internet Engineering Task Force, 2003.

Haberman, B. und Thaler, D. 2002. Unicast-Prefix-based IPv6 Multicast Addresses. *RFC 3306.* s.l. : Internet Engineering Task Force, 2002.

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

Hagen, Silvia. 2009. *IPv6: Grundlagen - Funktionalität - Integration.* 2. Auflage. Maur : Sunny Edition, 2009. ISBN: 978-3-9522942-2-2.

Hardaker, W. 2011. Requirements for Management of Name Servers for the DNS. *RFC 6168.* s.l. : Internet Engineering Task Force, 2011.

Hardie, T. 2002. Distributing Authoritative Name Servers via Shared Unicast Addresses. *RFC 3258.* s.l. : Internet Engineering Task Force, 2002.

Hinden, R. und Deering, S. 1995. IP Version 6 Addressing Architecture. *RFC 1884.* s.l. : Internet Engineering Task Force, 1995.

Hinden, R. und Haberman, B. 2005. Unique Local IPv6 Unicast Addresses. *RFC 4193.* s.l. : Internet Engineering Task Force, 2005.

IANA. 2013. IPv6 Multicast Address Space Registry. [Online] 28. Februar 2013. [Zitat vom: 19. März 2013.] http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xml.

ICANN. 2007. Internet Corporation for Assigned Names and Numbers. [Online] 1. März 2007. [Zitat vom: 14. März 2013.] http://www.icann.org/announcements/factsheet-dns-attack-08mar07_v1.1.pdf.

Ihlenfeld, J. 2008. DDoS-Angriff auf DNS-Provider. [Online] 21. November 2008. [Zitat vom: 22. April 2013.] http://www.golem.de/0811/63704.html.

Internet Systems Consortium. 2012. *BIND 9 Administrator Reference.* [Online] 2012. [Zitat vom: 27. Juni 2012.] https://kb.isc.org/getAttach/31/AA-00659/Bv9ARM.pdf.

ITU. 2007. *Framework and methodologies for the determination and application of QoS parameters. (ITU-T Recommendation E.802).* s.l. : International Telecommunications Union, 2007.

ITU 1991. *Security architecture for Open Systems Interconnection for CCITT applications (ITU-T Recommendation X.800).* s.l. : International Telecommunications Union, 1991.

Jankiewicz, E., Loughney, J. und Narten, T. 2011. RFC 6434. *IPv6 Node Requirements.* s.l. : Internet Engineering Task Force, 2011.

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

Jupiter Research. 2006. Retail Web Site Performance. [Online] 1. Juni 2006. [Zitat vom: 22. April 2013.] http://www.akamai.com/4seconds.

Leischner, M. 2012. Management vernetzter Systeme - Aufgabe und Einordnung. [Online] 3. April 2012. [Zitat vom: 14. März 2013.] http://www.leischner.inf.fh-bonn-rhein-sieg.de/lehre/alt/12ss/nmgt1/nm01-aufgabe.pdf.

Liu, Cricket. 2011. *DNS and BIND on IPv6.* Sebastopol : O'Reilly Media, 2011. ISBN: 978-1-449-30519-2.

Liu, Cricket und Albitz, Paul. 2006. *DNS and BIND.* 5. Auflage. Sebastopol : O'Reilly Media, 2006. ISBN: 978-0-596-10572-3.

Manning, B. und Vixie, P. 1996. Operational Criteria for Root Name Servers. *RFC 2010.* s.l. : Internet Engineering Task Force, 1996.

McPherson, D., et al. 2013. *Architectural Considerations of IP Anycast; <draft-iab-anycast-arch-implications-06>.* s.l. : Internet Engineering Task Force, 2013.

Mendez, T., Milliken, W. und Partridge, C. 1993. Host Anycasting Service. *RFC 1546.* s.l. : Internet Engineering Task Force, 1993.

Microsoft Corporation. 2012. DNS Processes and Interactions. [Online] 25. November 2012. [Zitat vom: 22. April 2013.] http://technet.microsoft.com/en-us/library/dd197552(v=ws.10).aspx.

Mills, E. 2012. Go Daddy-serviced Web sites go down; hacker takes credit. [Online] 10. September 2012. [Zitat vom: 22. April 2013.] http://news.cnet.com/8301-1009_3-57509753-83/go-daddy-serviced-web-sites-go-down-hacker-takes-credit/.

Mockapetris, P. 1983. DOMAIN NAMES - CONCEPTS and FACILITIES. *RFC 882.* s.l. : Internet Engineering Task Force, 1983.

Mockapetris, P. 1987. DOMAIN NAMES - CONCEPTS AND FACILITIES. *RFC 1034.* s.l. : Internet Engineering Task Force, 1987.

Mockapetris, P. 1983. DOMAIN NAMES - IMPLEMENTATION and SPECIFICATION. *RFC 883.* s.l. : Internet Engineering Task Force, 1983.

Mockapetris, P. 1987. DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION. *RFC 1035.* s.l. : Internet Engineering Task Force, 1987.

Nah, F. 2004. A study on tolerable waiting time: how long are Web users willing to wait? *Behaviour & Information Technology.* 2004, Bd. 23, 3, S. 153-163.

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

Quinn, et al. 2001. IP Multicast Applications: Challenges and Solutions. *RFC 3170.* s.l. : Internet Engineering Task Force, 2001.

RIPE NCC. 2013. K-root Homepage. [Online] 2013. [Zitat vom: 14. März 2013.] http://k.root-servers.org.

Savola, P. und Haberman, B. 2004. Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address. *RFC 3956.* s.l. : Internet Engineering Task Force, 2004.

Strotmann, Carsten. 2007. *New DNS Technologies in the LAN.* [Online] Oktober 2007. [Zitat vom: 27. Juni 2012.] http://meetings.ripe.net/ripe-55/presentations/strotmann-mdns.pdf.

Strotmann, Carsten 2012. *IPv6 Multicast - mehr als nur link-local.* [PDF] s.l. : Men & Mice, 2012.

Strotmann, Carsten. 2012. *Multicast Proxy for DNS (mcdnsProxy).* [Online] April 2012. [Zitat vom: 27. Juni 2012.] https://github.com/dnsworkshop/mcdnsProxy.

Thomson, et al. 2003. DNS Extensions to Support IP Version 6. *RFC 3596.* s.l. : Internet Engineering Task Force, 2003.

Vida, R. und Costa, L. 2004. Multicast Listener Discovery Version 2 (MLDv2) for IPv6. *RFC 3810.* s.l. : Internet Engineering Task Force, 2004.

Vixie, P. 1999. Extension Mechanisms for DNS (EDNS0). *RFC 2671.* s.l. : Internet Engineering Task Force, 1999.