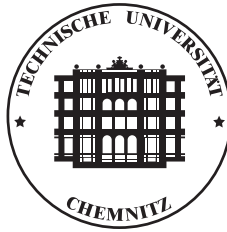


Analyse dynamischer Hostkonfiguration auf Basis von IPv6

Diplomarbeit



TU Chemnitz
Fakultät für Informatik

eingereicht von Ralph Meyer
am Lehrstuhl für Rechnernetze und verteilte Systeme

Betreuender Hochschullehrer: Prof. Dr.-Ing. habil. Uwe Hübner
Chemnitz, den 1. Oktober 2001

Thema: Gegenüberstellung der bisher gebräuchlichen Techniken zur dynamischer Hostkonfiguration gekoppelt an Autorisierungs,- Authentifizierungs,- Abrechnungsmechanismen im IPv4-Umfeld und den vorgeschlagenen Techniken/ Erweiterungen der Internet Protokollversion 6. Herausarbeitung von Vorteilen, Nachteilen und Problemen für eine Migrationsentscheidung des URZ im dynamischen Hostkonfigurationsumfeld vom bestehenden System auf IPv6. Implementierung eines DHCPv6-Servers und -Klienten (Plattform Linux)

Hinweis: Bezeichnungen von Erzeugnissen, die zugleich eingetragene Warenzeichen sind, wurden nicht als solche kenntlich gemacht. Aus dem Fehlen der Markierung ® bzw. ™ kann nicht geschlossen werden, dass die Bezeichnung ein freier Warenname ist. Ebenso wenig wird auf Patente oder Gebrauchsmusterschutz hingewiesen.

Inhaltsverzeichnis

| | | |
|----------|--|----------|
| 1 | Einleitung | 7 |
| 2 | Einführung in Authentifizierung, Autorisation und Accounting von Netzwerkressourcen | 9 |
| 2.1 | Authentifizierung, Autorisation, Accounting | 9 |
| 2.1.1 | Authentifizierung | 10 |
| 2.1.1.1 | Host zu Host Authentifizierung | 10 |
| 2.1.1.2 | Nutzerauthentifizierung | 10 |
| 2.1.2 | Autorisation | 13 |
| 2.1.2.1 | Verwaltung der Autorisationsregeln | 13 |
| 2.1.2.2 | Autorisationsmanagement über Ticketsysteme | 14 |
| 2.1.2.3 | Regelbasierter Netzwerkverkehr | 14 |
| 2.1.3 | Accounting | 15 |
| 2.1.3.1 | Erfassung der Accountingdaten | 16 |
| 2.1.4 | Bildung eines zentralen AAA- Servers | 16 |
| 2.2 | Wahl einer geeigneten Netzwerkprotokollschicht für die Authentifizierung | 17 |
| 2.2.1 | Authentifizierung in der Subnetzschicht | 17 |
| 2.2.2 | Authentifizierung in der Internetschicht | 18 |
| 2.2.3 | Authentifizierung in der Transportschicht | 18 |
| 2.2.4 | Authentifizierung auf Anwendungsebene | 18 |
| 2.2.5 | Authentifizierung am Kommunikationsstartpunkt | 18 |
| 2.3 | Aufbau einer AAA- Infrastruktur | 19 |
| 2.3.1 | Motivation für eine globale AAA- Infrastruktur | 19 |
| 2.3.2 | Ein Identifikator für jeden Nutzer | 19 |
| 2.3.2.1 | Eine IPv6 Adresse als ID | 19 |
| 2.3.2.2 | ID im Mobilfunk / IMSI, IMEI | 20 |
| 2.3.2.3 | Der Network Access Identifikator (NAI) | 22 |
| 2.3.2.4 | Fazit | 22 |
| 2.3.3 | Eine globale AAA- Infrastruktur | 23 |
| 2.3.3.1 | Globale AAA- Infrastrukturen für spezielle Einsatzgebiete | 24 |
| 2.4 | Anforderungen an eine AAA- Infrastruktur | 24 |
| 2.4.1 | Verbindung zwischen Nutzer und AAA- Agent | 26 |
| 2.4.2 | Verbindung zwischen AAA- Agent und AAA- Server | 26 |

| | | |
|----------|---|-----------|
| 2.5 | Vorgeschlagene Protokolle für eine AAA- Infrastruktur | 27 |
| 2.5.1 | Das Diameter Protokoll | 27 |
| 2.5.1.1 | Protokollaufbau | 28 |
| 2.5.2 | Protokolle für die Nutzer zu AAA- Agentenverbindung | 29 |
| 2.5.2.1 | Das Extensible Authentication Protokoll (EAP) | 29 |
| 2.5.2.2 | “Port Based Network Access Control” IEEE 802.1X | 29 |
| 2.5.2.3 | PPP | 31 |
| 2.5.2.4 | Mobile IPv6 | 31 |
| 2.5.2.5 | DHCPv6 Erweiterung | 31 |
| 2.5.2.6 | DHCPv6 Relay | 31 |
| 2.6 | Mechanismen zum Schutz der Privatsphäre des Nutzers | 32 |
| 2.6.1 | Automatische Veränderung der eigenen IPv6 Adresse | 32 |
| 3 | Vergleich bestehender AAA- Infrastrukturen mit geplantem neuen Infrastrukturaufbau | 33 |
| 3.1 | Bestehende AAA- Strukturen am Beispiel TU- Chemnitz | 33 |
| 3.1.1 | Die AAA- Zentrale | 34 |
| 3.1.2 | Druckdienst | 34 |
| 3.1.3 | Chemnitzer Studenten Netz - CSN | 34 |
| 3.1.4 | Unterstützung von mobilen Nutzern | 34 |
| 3.1.4.1 | Das Portmanagersystem | 35 |
| 3.1.4.2 | Wireless LAN Infrastruktur | 35 |
| 3.1.4.3 | UNI@HOME Einwahlzugang | 36 |
| 3.1.4.4 | DFN@HOME Einwahlzugang | 36 |
| 3.2 | Die neue AAA Infrastruktur | 37 |
| 3.2.1 | Der Druckdienst | 37 |
| 3.2.2 | Nutzerauthentifizierung | 38 |
| 3.2.3 | Unterstützung von mobilen Nutzern | 38 |
| 3.2.3.1 | Verwaltung öffentlicher Ports | 39 |
| 3.2.3.2 | DFN@HOME | 39 |
| 3.2.3.3 | Wireless LAN | 39 |
| 3.3 | Fazit | 39 |
| 4 | Adresskonfigurationsmechanismen in IPv6 | 41 |
| 4.1 | IPv6 Neuerungen | 41 |
| 4.1.1 | IPv6 Renumbering | 41 |
| 4.1.1.1 | Einführung einer Adressgültigkeit zur Renumberingunterstützung | 42 |
| 4.1.2 | Mehrere IP-Adressen pro Interface | 42 |
| 4.2 | IPv6 Adressaufbau | 43 |
| 4.2.1 | Spezielle Adresstypen | 43 |
| 4.2.2 | Unicast Adressen | 44 |
| 4.2.2.1 | Globale Unicast Adressen | 44 |
| 4.2.2.2 | Site- local Adressen | 45 |
| 4.2.2.3 | Link- local Adressen | 45 |

| | | |
|----------|--|-----------|
| 4.2.3 | Multicast Adressen | 45 |
| 4.2.4 | Anycast Adressen | 46 |
| 4.3 | Zustandslose Adresskonfiguration von IPv6- Hosts | 46 |
| 4.3.1 | Prinzip des Ablaufs der zustandslosen Adresskonfiguration | 47 |
| 4.3.2 | Entdeckung benachbarter Hosts (Neighbor Discovery) | 47 |
| 4.3.2.1 | Konfigurationsverbreitung über lokale Router (Router Advertise- ment) | 48 |
| 4.4 | Zustandsgebundene dynamische Adresskonfiguration von IPv6 Hosts | 49 |
| 4.4.1 | Entwicklung | 49 |
| 4.4.2 | Das DHCPv6 Protokoll | 50 |
| 4.4.2.1 | Nachrichtentypen | 50 |
| 4.4.2.2 | Der eindeutige DHCP Identifikator (DUID) | 51 |
| 4.4.2.3 | Klient- Adressen Beziehung (Identity association) | 51 |
| 4.4.2.4 | DHCP Optionen | 51 |
| 4.4.2.5 | Ein typischer Protokollablauf | 53 |
| 4.4.2.6 | Protokollablauf über ein DHCP- Relay | 54 |
| 4.4.2.7 | Verzicht auf Relay durch Adress Autoconfiguration | 55 |
| 4.4.2.8 | Klientenzustände | 55 |
| 4.4.3 | DHCP und ein dynamisches DNS | 55 |
| 4.4.3.1 | Unterstützung von sicheren DNS Updates | 56 |
| 4.5 | Nutzen eines zusätzlichen zustandsgebundenen Adresskonfigurationsprotokolls | 57 |
| 4.6 | Sicherheitsüberlegungen | 58 |
| 4.6.1 | DHCP Authentifizierung | 59 |
| 4.6.2 | DHCP Authentifizierung mit AAA- Server | 59 |
| 4.6.3 | Unterstützung von IPv4 Geräten durch den DSTM Mechanismus | 60 |
| 4.7 | Einordnung von DHCPv6 in die AAA- Infrastruktur | 62 |
| 4.8 | Fazit | 62 |
| 5 | DHCPv6 Implementierung | 63 |
| 5.1 | Server | 63 |
| 5.1.1 | ISC DHCP 3.0 Serveraufbau | 64 |
| 5.1.1.1 | Programmaufbau im Detail | 64 |
| 5.1.1.2 | Der Berkley Packet Filter | 65 |
| 5.1.2 | Die DHCPv6 Erweiterung | 66 |
| 5.1.2.1 | Programmaufbau | 66 |
| 5.2 | Der DHCPv6 Klient | 66 |
| 5.2.1 | Duplicate Address Detection | 68 |
| 5.2.2 | Verbindung zwischen Adresskonfigurationssystem im Systemkern und DHCPv6- Klient | 69 |
| 5.3 | Testumgebung | 70 |
| 5.3.1 | IPv6 Konfiguration | 70 |
| 5.3.2 | Besonderheiten der IPv6 Implementation in Linux | 70 |
| 5.4 | Test des DHCPv6 Prototyps | 71 |

| | |
|---|-----------|
| <i>INHALTSVERZEICHNIS</i> | 6 |
| 5.4.1 DHCPv6 Server Konfiguration | 71 |
| Abkürzungsverzeichnis | 72 |
| Abbildungsverzeichnis | 75 |
| Literaturverzeichnis | 77 |

Kapitel 1

Einleitung

Im vergangenen Jahr stieg das Interesse am Einsatz mobiler Netzwerke, z.B. Wireless LANs, rasant an. Ein Grund dafür sind preiswertere Hardwarekomponenten aber vor allem die zunehmende Erschliessung öffentlicher und privater Räume für den Funknetzzugang. Das geschieht durch Errichtung von Funknetzzellen sog. hot spots auf Flughäfen und in Innenstadtbereichen. Bisher haben diese Projekte allerdings noch Versuchscharakter. Internet Service Provider (ISP) stellen für ihre Versuche Dienste kostenfrei zur Verfügung und testen auf diese Weise Technik und Akzeptanz bei den Nutzern. Die Akzeptanz der Dienste hängt zum grossen Teil von den entstehenden Kosten, einem unproblematischen Zugang und einfacher Bedienung ab. Um genau das den Nutzern bieten zu können, ist seitens der ISP noch viel zu leisten. So muss eine Infrastruktur geschaffen werden, die es erlaubt:

- automatisch Endgeräte mit entsprechenden Einstellungen für den Netzzugang zu konfigurieren,
- einen beliebigen fremden Nutzer zu authentifizieren,
- bestimmten Nutzern (z.B. 1. Klasse Passagieren, Seminarteilnehmern) spezielle Dienste anzubieten oder zu verwehren,
- zu prüfen, ob der Nutzer kreditwürdig ist und
- Rechnungen über die angefallene Ressourcennutzung zu stellen.

Die vorangehend aufgezählten Abläufe müssen dabei für die Dienstanutzer möglichst transparent und unkompliziert erfolgen. Deshalb sind an die Protokolle der Authentifizierungs-, Autorisations- und der Accountingschicht (AAA) hohe Anforderungen zu stellen. Durch ihren Einsatz muss es möglich sein, eine ISP übergreifende AAA- Infrastruktur aufzubauen und zur Abstimmung des Austauschs der erforderlichen Daten (z.B. Nutzeridentifikatoren) mit den Hostkonfigurationsprotokollen zusammenzuarbeiten.

Die vorliegende Arbeit soll einen Einblick in die Problematik von Authentifizierung, Autorisation und Accounting geben und den zwischen AAA und Hostkonfigurationsprotokollen (z.B. DHCP) bestehenden Zusammenhang aufzeigen.

Im nachfolgenden 2. Kapitel erfolgt eine allgemeine Einführung in die AAA- Problematik und es werden die einzelnen Anforderungen an eine AAA- Infrastruktur dargestellt. Im 3. Kapitel wird eine bestehende AAA- Infrastruktur, die des URZ- Netzwerkes der TU- Chemnitz, mit einer noch fiktiven, zukünftigen AAA- Infrastruktur des URZ verglichen. Anschliessend soll in Kapitel 4 auf Hostkonfigurationsmechanismen unter IPv6 eingegangen und deren Schnittpunkte mit der AAA- Infrastruktur aufgezeigt werden. Am Ende der Arbeit in Kapitel 5 wird anhand einer Implementation des DHC- Protokoll Version 6 ein Beispiel für einen im 4. Kapitel erläuterten Hostkonfigurationsmechanismus vorgestellt.

Kapitel 2

Einführung in Authentifizierung, Autorisation und Accounting von Netzwerkressourcen

2.1 Authentifizierung, Autorisation, Accounting

Netzwerke, wie das Internet, stellen im weitesten Sinne den losen Zusammenschluss einer grossen Zahl von Dienstleistungsanbietern dar. Solche Dienste sind z.B.:

- als Kernfunktion: die Bereitstellung von physikalischen Leitungsnetzen durch Internet Service Provider,
- die Bereitstellung einer Funknetzinfrastruktur durch Mobilfunkprovider,
- Druckdienste,
- Informationsangebote,
- Datenbanken im Unternehmensnetzwerk oder
- Netzwerkdateisysteme.

Natürlich werden diese Dienste nicht jedem Nutzer frei zur Verfügung gestellt, sondern es fallen für deren Nutzung Kosten an, in welcher Weise auch immer, oder ihre Nutzung wird an bestimmte Voraussetzungen gebunden.

Solche Voraussetzungen sind u.a.:

- Zugehörigkeit zu einer bestimmten Nutzergruppe bzw. Organisation (Firma, Universität),
- ein Nutzer ist bekannt und kreditwürdig oder
- ein Nutzer ist Kunde einer bestimmten Firma.

Der Begriff “Nutzer” ist auch auf Geräte anwendbar, denn diese müssen ebenfalls zur Nutzung bestimmter Dienste, z.B. einer Netzwerkinfrastruktur, berechtigt sein.

Die angebotenen Dienste sind im allgemeinen nicht atomar, sondern bauen aufeinander auf, wie folgendes Beispiel verdeutlicht:

Nutzer N möchte ein Foto bei einem Druckdienst drucken lassen. Dazu sind mindestens zwei Dienstleister notwendig, ein ISP und der Druckdienstleister (DDL). Der ISP stellt seine Netzinfrastruktur zur Verfügung, um das Foto zum DDL zu übertragen. Der DDL wiederum stellt seine Drucker zur Verfügung. N muss sich für das Übertragen des Fotos beim ISP authentifizieren und nochmals beim DDL. Anschliessend werden beide Dienste getrennt voneinander abgerechnet.

Einfacher wäre es, die zweimalige Authentifizierung zu vermeiden und eine Möglichkeit für den Rückgriff des DDL auf die Authentifizierungsdaten des ISP und gegebenenfalls auch eine direkte Abrechnung über den ISP zu schaffen.

Dafür ist jedoch eine Infrastruktur notwendig, die es auch über Dienstleistungsgrenzen hinweg erlaubt, die Identität des Nutzers zu prüfen (Authentifizierung), eine Entscheidung anhand des authentifizierten Nutzers darüber zu treffen, ob ein bestimmter Dienst bereitgestellt wird (Autorisation), Nutzungsdaten zu sammeln und eine Abrechnung durchzuführen (Accounting).

2.1.1 Authentifizierung

Authentifizieren ist das Beweisen der Identität eines Nutzers. Da an die Authentifizierung jeweils verschiedene Ansprüche gestellt werden, ist eine Unterscheidung danach notwendig, wer sich welchem Kommunikationspartner gegenüber authentifiziert. In dieser Diplomarbeit werden lediglich die Host- zu- Host und die Benutzerauthentifizierung thematisiert. Die Details der verwendeten kryptographischen Protokolle sollen dagegen nicht behandelt werden und sind stattdessen bei [TRAPP 1999] nachzulesen.

2.1.1.1 Host zu Host Authentifizierung

Die Host- zu- Host Authentifizierung ermöglicht die Feststellung der Identität bzw. Authentizität der Hosts untereinander. Durch dieses Authentifizierungsverfahren wird sichergestellt, dass die mit einer bestimmten IP- Adresse verbundene Identität eines Hosts nicht verändert wurde, z.B. durch einen Austausch.

2.1.1.2 Nutzerauthentifizierung

Eine grundlegende Rolle in einer AAA- Infrastruktur kommt der Authentizität des Nutzers bzw. des Klienten zu. Denn ohne sichergestellte und geprüfte Identität des Nutzers sind Autorisation und nachfolgendes Accounting nicht durchführbar.

Um einen Nutzer zu authentifizieren, benötigt dieser einen eindeutigen Identifikator (z.B. UNIX- Nutzerkennzeichen), über den er mit dem Authentifizierungsmerkmal verknüpft wird. Für die Authentifizierung eines Nutzers stehen grundsätzlich drei Möglichkeiten zur Verfügung:

- Authentifizierung über ein geteiltes Geheimnis zwischen Nutzer und Authentifizierungseinheit,

- über ein global eindeutiges, unfälschbares Merkmal des Nutzers, welches der Authentifizierungseinheit bekannt ist bzw.
- über ein Zertifikat des Nutzers, das von einer vertrauenswürdigen Zertifizierungsstelle ausgegeben und mit dem öffentlichen Schlüssel des Nutzers authentifiziert wird.

Authentifizierung über ein Passwort

Die einfachste Methode der Authentifizierung ist die über ein Passwort. Zu diesem Zweck werden an die Nutzer Identifikatoren (z.B. Benutzername) und dazugehörige Passwörter vergeben. In diesem Verfahren stellt das Passwort das gemeinsame Geheimnis zwischen Authentifizierungsstelle und Nutzer dar, denn nur ihnen ist es bekannt.

Authentifizierung über ein Einmal- Passwort/ Token System

Bei diesem Verfahren erhält der Nutzer Benutzername, Passwort (meist als PIN) und einen Token. Der Token ist eine Spezialhardware und stellt einen Teil des Geheimnisses dar.

Im Falle von SecurID (RSA) hat der Token Scheckkartengrösse, ein numerisches Display und ein Eingabefeld. Der Nutzer gibt dort seine PIN ein. Mit Hilfe des enthaltenen Zufallsgenerators und der aktuellen Zeit wird alle 60 Sekunden ein neues gültiges Einmalpasswort für die Authentifizierung gegenüber dem Dienstleister erzeugt. Beim Dienstleister hingegen wird mit dem gleichen Zufallszahlengenerator und der Zeit das eingegebene Passwort überprüft. Bei Verlust des Tokens ist durch die PIN eine nicht autorisierte Nutzung ausgeschlossen.



Abbildung 2.1: SecurID Token

Authentifizierung über ein Challenge Response Verfahren

Bei Challenge Response Verfahren wird nicht das Geheimnis als solches übertragen, es muss vielmehr durch den Nutzer der Nachweis über die Kenntnis des Geheimnisses erbracht werden.

Dieses Verfahren findet u.a. bei Chipkarten Verwendung. Dabei wird der Chipkarte eine Challenge (Bitfolge) übergeben, aus der mit einem bestimmtem, nicht umkehrbaren Algorithmus (z.B. MD5) und der auf dem Chip gespeicherten geheimen Bitfolge eine Response errechnet wird. Diese wird anschliessend an den Authentifikationspartner übertragen, der die beschriebenen Schritte

nachvollzieht und sein Resultat mit dem empfangenen vergleicht. Kommt er auf das gleiche Resultat, wurde folglich die Kenntnis des Geheimnisses nachgewiesen und die Karte authentifiziert.

Authentifizierung durch ein eindeutiges Merkmal

Bei dieser Form der Authentifikation wird ein unfälschbares, eindeutiges Merkmal eingesetzt. Somit kommen für eine wirklich sichere Identifizierung nur biometrische Merkmale eines Menschen in Frage. Biometrische Merkmale lassen sich nach physiologischen (statischen) Merkmalen, wie dem menschlichen Fingerabdruck und nach verhaltensbasierten (dynamischen) Merkmalen, z.B. der Tippdynamik an der Tastatur, unterscheiden.

In der Host- zu- Host Authentifizierung würde das einer eindeutigen, unveränderbaren Seriennummer (z.B. Prozessor ID) oder einer aus der gesamten Konfiguration (z.B. verschiedene Seriennummern, Versionskennzahlen) errechneten Zahl entsprechen.

Authentifizierung anhand von Nutzerzertifikaten

Die Nutzung einer Zertifikats- und Schlüsselinfrastruktur ist eine Authentifizierungsmöglichkeit ohne den vorangehenden Geheimnisaustausch. Für deren Durchführung müssen folgende Voraussetzungen erfüllt sein:

- ein Nutzer besitzt einen öffentlichen Schlüssel und
- ein vertrauenswürdiger Zertifizierer hat mit einem Zertifikat die Verbindung zwischen Nutzer und seinem öffentlichem Schlüssel beglaubigt.

Der Nutzer kann nun seine Authentifizierungsaufforderung mit seinem privatem Schlüssel signieren, die Authentifizierungsgegenstelle prüft mit dem öffentlichen Schlüssel die Echtheit der Signatur und anhand des Zertifikates den Zusammenhang zwischen Nutzer und öffentlichem Schlüssel. Sind beide Prüfungen positiv verlaufen, ist davon auszugehen, dass die angegebene Identität des Nutzers richtig ist. Eine solche zertifikatsgebundene Authentifizierung benötigt eine globale Schlüssel/ Zertifikatsinfrastruktur (z.B. PKI s. a. [PKI DRAFT]).

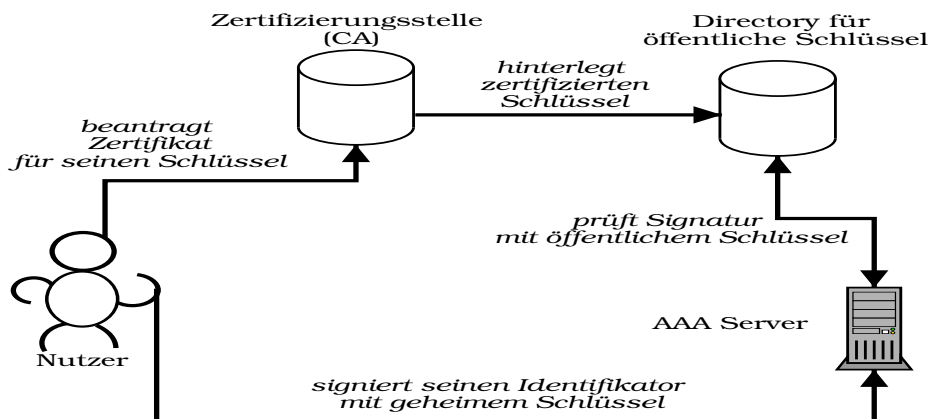


Abbildung 2.2: Public Key Infrastruktur

2.1.2 Autorisation

Autorisation ist das Prüfen nutzerabhängiger Vorschriften. In diesen Vorschriften wird der Zugriff auf bestimmte Dienste bzw. Ressourcen geregelt. Die Prüfung fällt entweder positiv aus, der Nutzer darf auf die Ressource zugreifen oder negativ, dem Nutzer ist der Zugriff zu verweigern. Bei der Prüfung wird das Autorisationssystem direkt von der Dienstverwaltungsinstanz befragt, ob der authentifizierte Nutzer berechtigt ist, auf den Dienst zuzugreifen.

Autorisationsentscheidungen hängen fast immer von folgenden Punkten ab:

- der Berechtigung des Nutzers, den Dienst zu nutzen,
- dem Nutzungsumfang des Dienstes,
- der Kreditwürdigkeit des Nutzers bzw. der Sicherstellung der Zahlung sowie
- Randbedingungen wie Uhrzeit, Netzwerkbelastung, usw.

Diese Regeln lassen sich untereinander beliebig kombinieren.

2.1.2.1 Verwaltung der Autorisationsregeln

Regeln für die Autorisation sind bei den meisten AAA- Systemen (z.B. RADIUS [RIGNEY 2000]) anhand von Access Control Lists (ACL) festgelegt. In diesen Regeln ist für jeden Nutzer genau beschrieben, welche Netzwerkdienste genutzt werden dürfen. Auch das gemeinsame Geheimnis kann in einer ACL hinterlegt sein.

Nachfolgend eine Beispiel ACL eines RADIUS Servers:

```
bobuser Password = "welcome"
  User-Service-Type = Shell-User,
  cisco-avpair = "shell:autocmd=access-profile"
  User-Service-Type = Framed-User,
  Framed-Protocol = PPP,
  cisco-avpair = "ip:inacl#3=permit tcp any host 10.0.0.2 eq telnet",
  cisco-avpair = "ip:inacl#4=deny icmp any any"
```

In obigem Beispiel wird für den Nutzer "bobuser" das Passwort "welcome" vergeben. Der Nutzer benutzt für die Einwahl das Point to Point Protokoll ("Framed-Protocol = PPP") und mit "permit tcp any host 10.0.0.2 eq telnet" wird ihm der Zugriff lediglich auf die Kommandoshell (über telnet) des Hosts 10.0.0.2 gestattet. Die Shell wird entsprechend der Angaben im "access-profile" gestartet. Weiterhin wird die Weiterleitung von ICMP Paketen des Nutzers in das besuchte Netzwerk verboten.

Im Beispiel ist die ACL in einer einfachen Textdatei auf dem RADIUS- Server hinterlegt, in der Praxis werden die ACLs zusammen mit den Accountingdaten meist in einer Datenbank oder in einer Verzeichnisstruktur wie z.B. LDAP verwaltet.

2.1.2.2 Autorisationsmanagement über Ticketsysteme

Bei einem Ticketsystem (z.B. Kerberos [KOHL 1993]) existiert ein zentrales Verwaltungssystem, in dem das gemeinsame Geheimnis des Nutzers und die Berechtigungen für einzelne Dienste gespeichert sind. Jeder Dienst ist in der Lage, Tickets anzufordern. Mit der ersten Anmeldung erhält der Nutzer ein Sitzungsticket, das ihn gegenüber anderen Diensten für die Zeit der Sitzung identifiziert. Greift er auf einen Dienst zu, fordert dieser vom Verwaltungssystem ein Ticket für den Nutzer an. Das Verwaltungssystem prüft, ob der Nutzer autorisiert ist, den Dienst zu nutzen und sendet gegebenenfalls ein entsprechendes Ticket an den Dienst.

Im Fall von Kerberos ist der Ablauf wie folgt (vereinfacht):

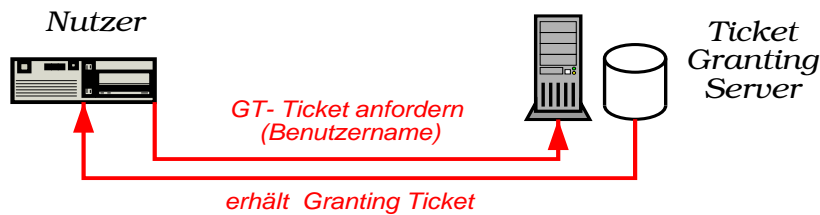


Abbildung 2.3: Erste Ticketanfrage

1. Der Klient fordert vom Ticket Granting Server (TGS) das Granting Ticket (GT) an, indem er seinen Benutzernamen an den TGS sendet.
2. Der TGS erstellt und sendet dem Klient ein Ticket, das mit dem Passwort des Nutzers verschlüsselt wird.
3. Nur der Klient kennt das Passwort und kann das Ticket entschlüsseln, es enthält einen zufälligen Sitzungsschlüssel und das mit diesem verschlüsselte GT.
4. Mit dem GT wiederum fordert der Klient für jeden Dienst, auf den er zugreifen möchte, beim TGS weitere Tickets an, damit ist die Authentizität des Nutzers sichergestellt, denn nur er konnte das GT und den Sitzungsschlüssel mit seinem Passwort entschlüsseln.
5. Der TGS sendet dem Klient für den gewünschten Dienst ein Ticket, sofern er für dessen Nutzung in seiner ACL berechtigt ist. Anschliessend wird dem Dienstserver das Ticket des Klienten übermittelt.
6. Der Klient sendet das Ticket an den Dienstserver, dieser vergleicht das Ticket mit dem vom TGS erhaltenen, bei Übereinstimmung wird dem Klienten der Zugang zum Dienst gewährt.

2.1.2.3 Regelbasierter Netzwerkverkehr

Mit COPS [DURHAM 2001] wurde ein Protokoll für ein regelbasiertes Netzwerk definiert. Die Motivation dafür war die Schaffung einer AAA- Umgebung für RSVP und Diffserv. Der Aufbau von COPS ist jedoch so modular, dass sein Einsatz für jegliche Dienste möglich ist.

Es besteht aus einem zentralen Policy Decision Point (PDP), der regelbasierte Entscheidungen trifft und aus Netzwerkgeräten, die als Policy Enforcement Points (PEP) diese Entscheidungen

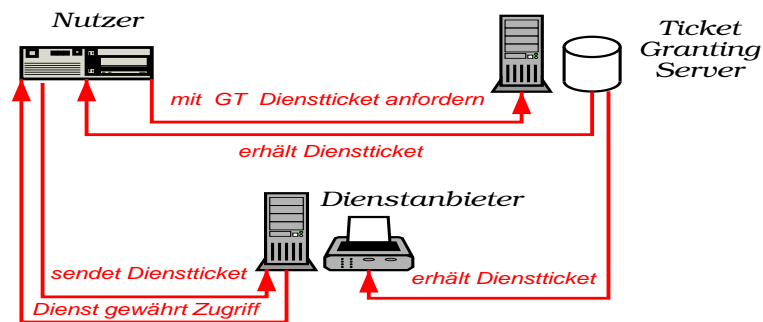


Abbildung 2.4: Dienstticket Anforderung

umsetzen. COPS unterscheidet zwischen zwei Modellen, dem Provisioning und dem Outsourcing. Beim Provisioning wendet sich der Benutzer, um einen bestimmten Dienst in Anspruch zu nehmen, an den PDP, der seine Entscheidung und entsprechende Regeln an den PEP übermittelt. Die PEPs setzen diese Regeln um und der Nutzer kann auf den Dienst zugreifen. Beim Outsourcing wendet sich der Nutzer z.B. über eine reservation message von RSVP an den PEP, der diese Anfrage dem zuständigen PDP übermittelt. Der PDP antwortet mit seiner Entscheidung in Form einer Decision (DEC) Nachricht.

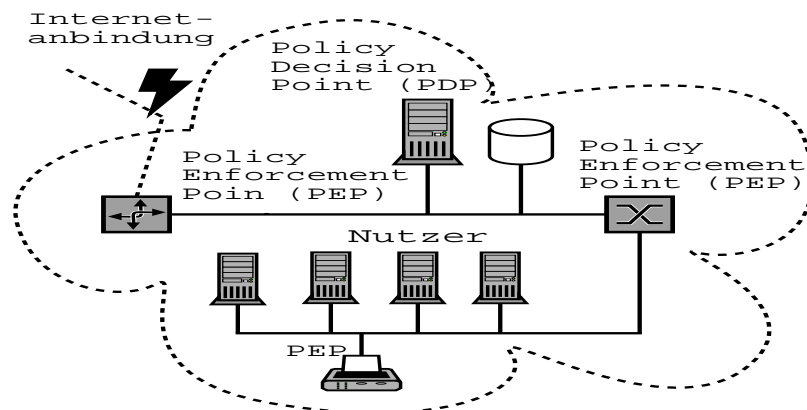


Abbildung 2.5: COPS Aufbau

2.1.3 Accounting

Der Begriff Accounting wird in [MILLS 1991] folgendermassen definiert:

- Messung von erfolgter Ressourcennutzung (metering),
- Sammeln der Messdaten (collecting),
- Errechnung der angefallen Kosten (charging) und
- Rechnungslegung pro Nutzer (billing).

Accountingdaten stellen gesammelte Messdaten dar. Sie spielen ausser für die Rechnungslegung noch für andere Auswertungen eine wichtige Rolle:

- die Ressourcenplanung,
- die Anzeige der Ressourcennutzung (z.B. Nutzungszeiten, Nutzungsspitzen),
- die Feststellung von Ressourcenmissbrauch.

Für ein erfolgreiches Accounting müssen die Nutzungsdaten der Dienste gesammelt und verlässlich gespeichert werden. Die Daten müssen entsprechend aussagekräftig sein, um sie bestimmtem Nutzern zuzuordnen und aus ihnen nachvollziehbare Rechnungen für die Nutzer erstellen zu können. Gleiches gilt bezüglich der Statistik.

2.1.3.1 Erfassung der Accountingdaten

Accountingdaten, wie z.B. Nutzungsdauer, gedruckte Seiten usw. fallen direkt beim Dienstbringer an. Um zentrales Accounting zu ermöglichen, müssen die Daten entweder vom Dienstbringer (DE) geliefert (push) oder vom Accountingsystem angefordert (poll) werden. Da zur Zeit DE mit SNMP Schnittstellen ausgestattet sind, müssen Daten direkt angefordert werden. Dies birgt Risiken, wie etwa den Verlust von Accountingdaten im DE bei kurzzeitigem Ausfall der Accountingzentrale (aufgrund zu kleinen Zwischenspeichers).

2.1.4 Bildung eines zentralen AAA- Servers

Ziel eines AAA- Servers ist es, AAA- Funktionalität zu vereinen. Folgenden Anforderungen sollte ein AAA- Serversystem genügen:

- Authentifizierung des Nutzers über verschiedene Protokolle bzw. Methoden,
- Verwaltung der Nutzerdaten,
- Verwaltung der nutzerspezifischen Regeln,
- Aufzeichnung der Abrechnungsdaten für jeden Nutzer,
- Bearbeitung von Dienstleistanfragen zur Nutzerberechtigung,
- Kommunikation mit anderen, nicht lokalen AAA- Systemen, um z.B. fremde Nutzer zu authentifizieren,
- Weiterleitung von Accountingdaten an nicht lokale AAA- Systeme, z.B. Daten fremder Nutzer,
- Unterstützung verschiedener AAA- Protokolle und
- Bereitstellung einer Management Schnittstelle.

Natürlich muss die gesamte Auslegung des Serversystems möglichst redundant und verteilt erfolgen, um einen einzelnen verwundbaren Punkt zu vermeiden.

2.2 Wahl einer geeigneten Netzwerkprotokollschicht für die Authentifizierung

Das Internet besteht aus verschiedenen aufeinander aufbauenden Protokollschichten. Dieser Aufbau entspricht, wie in Tabelle 2.1 gezeigt, dem ISO-OSI Schichtenmodell. Eine Nutzerauthentifizierung kann auf jeder der vier Schichten stattfinden. Es besteht natürlich ebenfalls die Möglichkeit, eine Authentifizierung auf mehreren Schichten hintereinander durchzuführen bzw. auf eine bereits erfolgte Authentifizierung in einer niederen Schicht zurückzugreifen. Grundsätzlich gilt, je grösser die Schicht in der die Authentifizierung abläuft, desto mehr Angriffsmöglichkeiten besitzt der Nichtauthentifizierte, wenn man davon ausgeht dass jede Implementierung einer bestimmten Schicht potentielle Schwächen besitzt bzw. Fehlkonfigurationen aufweisen kann.

| Internet Schicht Nr. | ISO-OSI Schicht Nr. | Name |
|----------------------|---------------------|-----------|
| 4 | 5,6,7 | Anwendung |
| 3 | 4,5 | Transport |
| 2 | 3 | Internet |
| 1 | 1,2 | Subnetz |

Tabelle 2.1: ISO-OSI Schichtenmodell

2.2.1 Authentifizierung in der Subnetzschiicht

Die Authentifizierung schon auf der Subnetzschiicht vorzunehmen, ist immer dann sinnvoll wenn:

1. die nächsthöhere Schicht nur authentifizierten Nutzern zur Verfügung gestellt werden soll und/ oder
2. die Subnetzschiicht schon eine zu schützende Ressource bzw. einen zu schützenden Dienst darstellt.

Ein Beispiel für den zweiten Fall ist eine Firma oder Universität mit einer grossen Netzinfrastruktur, die auch Netzwerkanschlussdosen in nicht vertrauensvollen Bereichen (Hörsäle, Gänge, Nebenräume) aufweist. Hier ist es wichtig, das potentielle Angreifer an Netzwerkports keine Möglichkeit haben, den Netzverkehr abzuhören oder zu beeinflussen. Deshalb wurde im IEEE 802.1x Standard [NW 14/99] ein Protokoll entwickelt, das eine Authentifizierung schon auf Switchportebene ermöglicht. Das Protokoll setzt voraus, dass die Netzwerkdosen über einen Switch angeschlossen sind. Ein Gerät bzw. Nutzer meldet sich an einem Port an, der Switch leitet die Anmeldedaten an eine AAA- Infrastruktur weiter und bei positiver Rückantwort wird der Port freigegeben. Bei nicht authentifizierten Geräten wird kein Netzwerkverkehr weitergeleitet.

Nachteile dieser Lösungen sind:

- sie sind nur an das jeweilige Subnetzprotokoll angepasst und
- es wird spezielle Hardware benötigt.

Bisherige Lösungen dieses Problems (z.B. das Portmanagersystem [BREILER 2000]) arbeiten mit einer Authentifizierung in Schicht 4, wobei die notwendige Kommunikation über ein gesichertes

virtuelles Subnetz abläuft. Danach wird dem Nutzer der Zugang zur Netzinfrastruktur anhand von Regeln gestattet, die seine Hardwareadresse identifizieren (MAC- Adresse).

2.2.2 Authentifizierung in der Internetschicht

Auf Internetprotokollebene kann in jedem IP- basierten Netzwerk eine Authentifizierung durchgeführt werden, unabhängig von der darunterliegenden Subnetzschiicht. Meiner Ansicht nach ist diese Schicht die geeignetste, um Authentifizierungsprotokolle ablaufen zu lassen.

2.2.3 Authentifizierung in der Transportschicht

Die Transportschicht bietet sich an, um bestimmte Dienste wie die Host- zu- Host Authentifizierung durchzuführen. Durch standardisierte Schnittstellen zur Anwendungsschicht ist es möglich, mit der Anwendung zu kommunizieren und sich über eine fehlgeschlagene Authentifizierung zu informieren. Das Transport Layer Security (TLS vormals Secure Socket Layer (SSL) [DIERKS 1998]) ist ein Beispiel für solch eine Schnittstelle.

2.2.4 Authentifizierung auf Anwendungsebene

Eine Authentifizierung des Nutzers auf Anwendungsebene ist immer dann erforderlich, wenn:

- die eigene AAA- Infrastruktur verlassen wird,
- die Anwendung keine Schnittstelle zur eigenen AAA- Infrastruktur besitzt oder
- bestimmte Dienste (u.a. Homebanking) eine weitere explizite Authentifizierung bzw. Autorisation erfordern.

Die Authentifizierung auf Anwendungsebene läuft genau wie auf jeder anderen Schicht ab. Es muss die Kenntnis des gemeinsamen Geheimnisses in irgendeiner Weise nachgewiesen werden. Das geschieht meist über eine direkte Passworteingabe bzw. das Erkennen eines biometrischen Merkmals des Nutzers über zusätzliche Hardware.

2.2.5 Authentifizierung am Kommunikationsstartpunkt

Diese Authentifizierungsmöglichkeit ist ein Sonderfall. Hier stellt der Klient eine direkte Verbindung (z.B. Telefon, seriell) zu einem Netzwerkszugangspunkt her (Network Access Server (NAS)), wo er sich authentifizieren muss. Der NAS prüft anhand seiner Datenbank, ob der Klient autorisiert ist, diese Verbindung zu nutzen, gegebenenfalls erhält er erforderliche Adressen, und die Verbindung wird gestartet. Zwischen NAS und Klient besteht jetzt ein Tunnel, der es unabhängig vom Subnetzprotokoll erlaubt, z.B. IP- Verkehr auszutauschen.

Ein Beispiel dafür ist ein ISP, der mit Telefoneinwahlzugängen arbeitet. Er möchte die IP-Funktionalität der Schicht 2 nur seinen angemeldeten Kunden bereitstellen. Diese benutzen das Point- to- Point Protokoll [PPP] als Zwischenschicht über dem Subnetzprotokoll. PPP beinhaltet die Authentifizierungsprotokolle [PAP, CHAP, EAP]. Es wird sofort nach der Einwahl die Authentifizierung durchgeführt und nur bei positivem Ergebnis (gültiges Passwort) durch Zuweisung einer IP- Adresse in die nächst höhere Schicht [Internet] vermittelt.

Das PPP wurde ursprünglich für die Telefoneinwahl entwickelt, es kann aber an jede Subnetzschiicht angepasst werden. Um die PPP- Funktionalität auch bei permanenten Leitungen wie DSL oder Breitbandfernsehkabel ohne Änderungen an der bestehenden AAA- Infrastruktur beizubehalten, wird PPP z.B. über Ethernet (PPPoE) verwendet, um damit nur authentifizierten Nutzern den Zugang zu ermöglichen und das bestehende Accountingssystem weiterzunutzen.

2.3 Aufbau einer AAA- Infrastruktur

2.3.1 Motivation für eine globale AAA- Infrastruktur

Das Internet wird zunehmend für Anwendungen und Dienste genutzt, deren langfristiger Erfolg vom Aufbau einer einheitlichen AAA- Infrastruktur abhängig ist. Dazu zählen in erster Linie:

- mobile Internetdienste im Zusammenhang mit der 3. Mobilfunkgeneration,
- Internet Telephonie,
- Micropaymentsysteme.

Allen gemeinsam ist, dass sie einen Austausch von Nutzerauthentifizierungs- und Accountingdaten zwischen mehreren Partnern erforderlich machen.

2.3.2 Ein Identifikator für jeden Nutzer

Ohne einen global eindeutigen Identifikator (ID) für jeden Nutzer gibt es keine verlässliche Möglichkeit, mit dieser ID einen öffentlichen Schlüssel oder ein gemeinsames Geheimnis für die Authentifizierung zu verbinden. Bei der Wahl eines geeigneten Identifikators muss folgenden Anforderungen genügt werden:

- global eindeutig,
- Unterstützung eines Namenssystems zur Bestimmung der Home Domain,
- begrenzte Länge, um direkte Eingabe zu ermöglichen,
- der Nutzer selbst muss identifiziert werden, nicht ein Endgerät, da nicht gewährleistet werden kann, dass dieses Endgerät von der berechtigten Person genutzt wird.

2.3.2.1 Eine IPv6 Adresse als ID

Mit der Entwicklung von IPv6 besteht nunmehr die Möglichkeit, jedem Nutzer und jedem Gerät eine feste, eindeutige IP- Adresse zuzuweisen. Sie kann dabei gleichzeitig als Identifikator dieses Nutzers eingesetzt werden. Bei genauerer Betrachtung erweist sich die IPv6- Adresse jedoch aus folgenden Gründen als Nutzeridentifikator ungeeignet:

- Die Nutzung der gesamten Adresse inklusive der Präfixe würde dem IPv6 Renumbering widersprechen, also sind nur die letzten 64 Bit EUI nutzbar.

- Wenn nur die 64 Bit EUI benutzt werden, ist es nahezu unmöglich zu ermitteln, zu welcher administrativen Domain der Nutzer gehört.
- Bei Multiuserbetriebssystemen wäre es notwendig, für jeden angemeldeten Nutzer eine andere Absenderadresse einzusetzen, um nicht nur das Gerät zu identifizieren.
- Die Privatsphäre wäre nicht mehr geschützt, man würde bei jeder Kommunikation über das Internet seine Identität hinterlassen. Im realen Leben wird auch nicht unaufgefordert in jedem besuchten Geschäft bzw. an jedem Ort eine Visitenkarte hinterlegt.

Eben gesagtes gilt auch bezüglich der Nutzung der Adresse als globaler Identifikator für Geräte. Für die Identifizierung von Geräten in der eigenen Netzinfrastruktur ist die IPv6 Adresse bzw. der Hostteil ebenfalls nur bedingt einsetzbar, da Geräte unter Umständen ausgetauscht werden können, der Service aber unter der selben Adresse erreichbar bleiben soll. Mit einem Tausch jedoch würde sich der Hostteil der IPv6 Adresse ändern.

2.3.2.2 ID im Mobilfunk / IMSI, IMEI

Die Identifikation von Nutzern in bestehenden GSM- Mobilfunkstrukturen erfolgt in eleganter Weise, denn es werden gleich mehrere Probleme auf einmal gelöst:

- die Identifikation des Gerätes,
- die Identifikation des Nutzers,
- die Geheimhaltung der Identität der Teilnehmer und
- die Verteilung eines gemeinsamen Geheimnisses.

Dies wird durch die Vergabe von SIM- Chipkarten (Subscriber Identity Module, (SIM)) erreicht, in denen folgendes hinterlegt ist:

| | |
|------|--|
| IMSI | Die Mobilteilnehmerkennung (International Mobile Subscriber Identity), eine bis zu 15 Ziffern lange Nummer, die vom Netzbetreiber der SIM-Karte fest zugeordnet wurde. In der IMSI ist gleichzeitig das Heimatnetzwerk des Teilnehmers codiert. Die ersten 3 Ziffern der IMSI bezeichnen die Mobillandeskennzahl, die nächsten 2 den Netzwerkcode und somit den Heimatprovider, die restlichen 10 Ziffern identifizieren den Teilnehmer. |
| ISAK | Individual Subscriber Authentication Key, ein bis zu 128 Bit langer, geheimer Schlüssel, der von der IMSI abhängt und wie diese nur im SIM und bei der Authentifizierungsstelle des Mobilfunkproviders (AUC) gespeichert ist. |

Im Mobilfunkgerät selbst ist, unabhängig von allen anderen Daten, die Mobilgeräteerkennung (International Mobile-Station Equipment Number, IMEI) gespeichert.

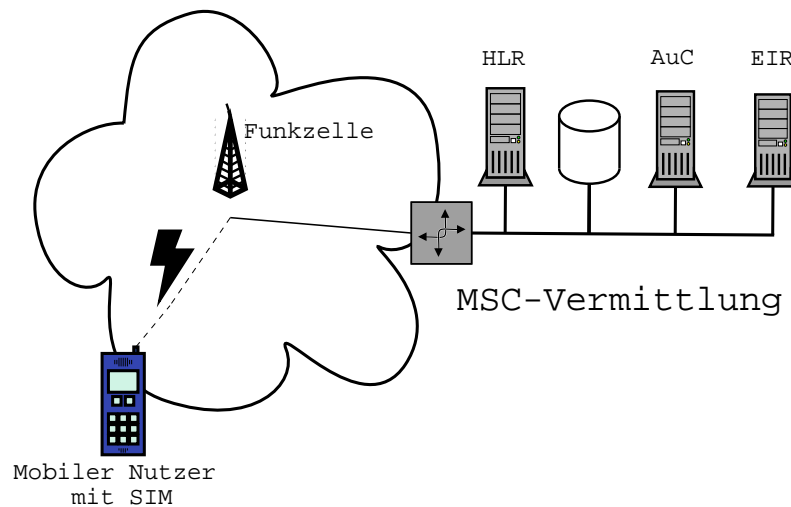


Abbildung 2.6: AAA im GSM- Standard

Ablauf der Authentifizierung im GSM- Netz

Nach dem Einschalten des mobilen Gerätes wird geprüft, ob eine Funkzelle erreichbar ist. Wenn ja, läuft folgendes ab (vereinfacht):

1. IMSI und IMEI Nummer werden an die Vermittlungsstelle MSC (Mobile Switching Center) übertragen.
2. Mit der IMSI wird anhand des HLR (Home Location Register) geprüft, ob der Teilnehmer Kunde des Providers und berechtigt ist, sich ins Funknetz einzubuchen.
3. Mit der IMEI wird anhand des EIR (Equipment Identity Register) überprüft, ob das Gerät auf der weissen Liste steht (schwarz = gestohlen, grau = fehlerhaft, weiss = ok).
4. Mit einem Challenge- Response Verfahren wird der Teilnehmer authentifiziert, indem er die Kenntnis über den ISAK Schlüssel nachweist.
5. Es wird ein Sitzungsschlüssel aus einer von der MSC übertragenen Zufallszahl und dem ISAK gebildet, mit dem die weitere Kommunikation zwischen mobilem Gerät und MSC verschlüsselt wird.
6. Dem Teilnehmer wird eine temporäre TMSI (Temporary Mobile Subscriber Identity) zugewiesen. Ein Rückschluss auf seine Identität (IMSI) ist nur im Home Location Register (HLR) möglich. Somit ist seine Identität und sein Aufenthaltsort (im Funkzellenradius) gegenüber dem Übertragungsnetzwerk verschleiert.
7. Der Authentifizierungsvorgang wurde abgeschlossen.

Nachteilig ist lediglich der Einsatz eines zusätzlichen Tokens in Form der SIM- Karte. Da sich jedoch kaum ein Nutzer mehrere 15 stellige Zahlen und 64 bis 128 Bit lange Schlüssel wird merken wollen, ist die Verwendung der SIM - Karte eine akzeptable Alternative.

2.3.2.3 Der Network Access Identifikator (NAI)

In aktuellen, von der IETF (Internet Engineering Task Force) vorgeschlagenen, Standards wird zur Nutzeridentifikation die Verwendung des Network Access Identifikators (NAI) nach [ABOBA 1999] empfohlen. Der Aufbau des NAI ähnelt dem einer E-Mail Adresse, er besteht aus Nutzernamen und Domainnamen getrennt durch ein “@” Symbol. Beispiele für gültige NAIs sind:

- lisa.simpson@homegrown.nl
 - ralph.meyer@informatik.tu-chemnitz.de

Über den Domainnamen ist die Heimatdomain bekannt und durch eine Anfrage an das Domain Name System (DNS), evtl. über einen neuen Recordtyp oder im “TEXT” Recordtyp, kann der Heimatauthentifizierungsserver herausgefunden werden. Die Verwaltung der Nutzernamen obliegt der Heimatdomain, sie müssen für diese Heimatdomain eindeutig sein.

2.3.2.4 Fazit

Die Verwendung der IPv6 Adresse als Nutzeridentifikator ist, wie vorangehend dargestellt, weniger praktikabel. Meiner Ansicht nach ist vielmehr die Kombination eines NAI mit einem SIM ähnlichen Hardwaretoken (z.B. Chipkarte, USB- Key) eine geeignete Möglichkeit, Nutzer zu identifizieren und anhand eines Geheimnisses zu authentifizieren. Dafür müssen jedoch durch eine geeignete Instanz (ISP) folgende Voraussetzungen geschaffen werden:

- ISPs müssten eine Infrastruktur schaffen, um jedem Nutzer Hardwaretoken auszugeben.
- Mobile Geräte (PDAs, Laptops) müssten mit einer Schnittstelle für den Hardwaretoken ausgestattet werden, entweder standardmässig oder über eine Erweiterung, z.B. durch Chipkartenaufnahme in eine Wireless LAN PC- Card.
- Der Inhalt des Hardwaretokens und die verwendeten Datenstrukturen müssten standardisiert werden.
- Benutzte Algorithmen im Hardwaretoken müssten frei zur Verfügung stehen, damit auch für kleine ISP bzw. Organisationen sichergestellt ist, dass sie ohne grossen finanziellen Aufwand ihren Nutzern Hardwaretoken ausgeben können.
- Zudem müsste eine AAA- Infrastruktur ähnlich der der Mobilfunknetzbetreiber von den ISP aufgebaut werden (Ansätze und Protokolle siehe Kapitel 3).



Abbildung 2.7: Beispiel eines USB Hardwaretokens

Ansätze einer solchen Entwicklung sind im Arbeitspapier der IETF [KNIVETON 1999] zu finden. Dort wird beschrieben, wie die bestehende SIM basierte Authentifizierung in einer AAA- Umgebung genutzt werden kann.

2.3.3 Eine globale AAA- Infrastruktur

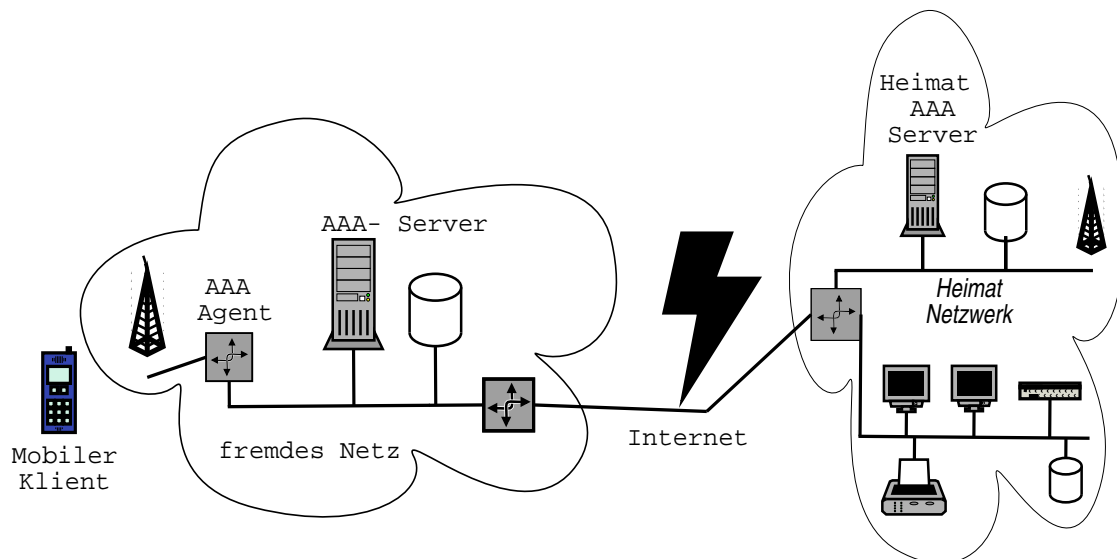


Abbildung 2.8: AAA- Infrastruktur

Um als mobiler Nutzer von seiner administrativen Heimatdomain in eine fremde Netzwerkdomain zu wechseln (roaming), wird eine AAA- Infrastruktur benötigt. Bevor ich auf die genauen Anforderungen an eine AAA- Infrastruktur eingehe, möchte ich die prinzipiellen Abläufe des Domainwechsels näher erläutern:

1. Der Nutzer befindet sich im fremden Netzwerk (z.B. über eine Wireless LAN Verbindung oder direkt per Ethernet).
2. Er nimmt Kontakt zum nächstgelegenen AAA- Agenten (FA) auf, meist der nächste Router.
3. Er übermittelt seinen Identifikator (z.B. seinen NAI) an den AAA- Agent.
4. Der AAA- Agent nimmt Kontakt mit dem lokalen AAA- Server auf (AAAL) und übermittelt diese Identifikatoren.
5. Der AAA- Server ermittelt anhand des Identifikators die Heimatdomain des Nutzers und nimmt Kontakt zum dortigen AAA- Server auf (AAAH).
6. AAAH authentifiziert den Nutzer direkt oder er stellt eine Verbindung über einen Tunnel her, um z.B. per Challenge Response den Nutzer zu authentifizieren.
7. AAAL erhält das OK und der AAA- Agent kann die Verbindungen zum mobilen Nutzer gestatten.

2.3.3.1 Globale AAA- Infrastrukturen für spezielle Einsatzgebiete

Einen anderen Ansatz verfolgen AAA- Systeme, deren Dienste speziell auf einen ausschliesslichen Einsatzbereich ausgerichtet sind. Zu dieser Kategorie zählen u.a das Passport System von Microsoft und das von SUN (Liberty Alliance) geplante Gegenstück NetID. Diese Systeme sind ausschliesslich darauf ausgerichtet, eine Infrastruktur dafür zu schaffen, Nutzer im World Wide Web (WWW) einmalig pro Sitzung an einer zentralen Stelle (AAA- Server) zu authentifizieren und zu autorisieren. Angeschlossene WWW Dienstanbieter können im Anschluss auf diese Authentifizierungsdaten zurückgreifen ohne dass sich der Nutzer erneut anmelden muss. Der Vorteil für den Nutzer besteht in einer Erhöhung des Komforts bei der Nutzung des WWW. Den weitaus grösseren Nutzen haben aber die Dienstleister, die nun durch diese Systeme an zentraler Stelle ihre Kundenprofile zusammenführen und auswerten können.

Am Beispiel von Microsofts Passport System möchte ich den Ablauf der Authentifizierung erläutern.

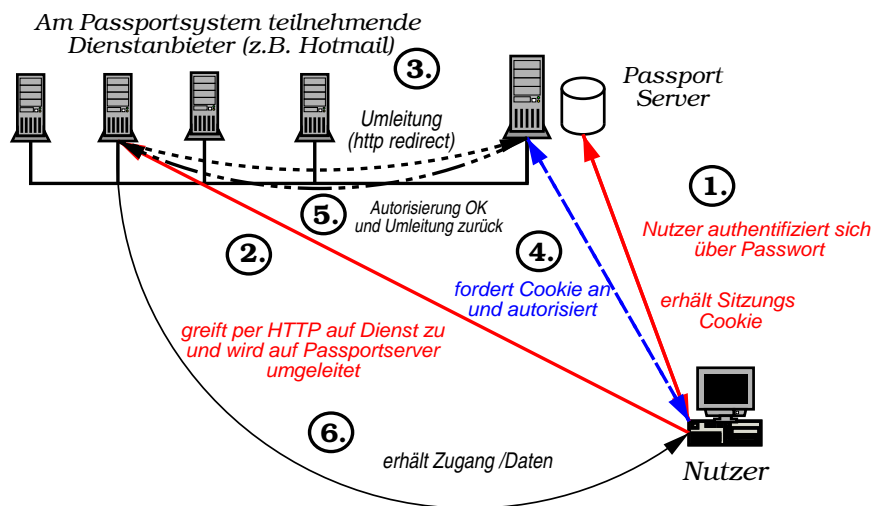


Abbildung 2.9: Microsoft Passport System

Das Passport System nutzt den Cookie und den HTTP Redirect Mechanismus. Cookies werden schon jetzt dazu eingesetzt, Nutzer wiederzuerkennen und z.B. bei amazon.de zu authentifizieren. Der Cookie Mechanismus hat jedoch den Nachteil, dass nur für denjenigen die Möglichkeit besteht den Cookie aus dem Internetbrowser des Nutzers zu lesen, der ihn auch geschrieben hat. Passport nutzt ein HTTP Redirect, um diese Einschränkung zu umgehen. Der Nutzer meldet sich für eine Sitzung auf dem Passport System an und erhält einen Cookie. Wenn er einen Passportpartner Dienstleister besucht, wird seine HTTP Anfrage kurzzeitig auf den Passportserver umgeleitet. An dieser Stelle wird der Cookie ausgelesen, die Authentifizierungsdaten an den Partner übermittelt und die Nutzeranfrage danach zum Partnerangebot zurückgeleitet. Für den Passportpartner ist der Nutzer nun authentifiziert.

2.4 Anforderungen an eine AAA- Infrastruktur

Um diese AAA- Funktionalität domainübergreifend zu gewährleisten, müssen von AAAL und

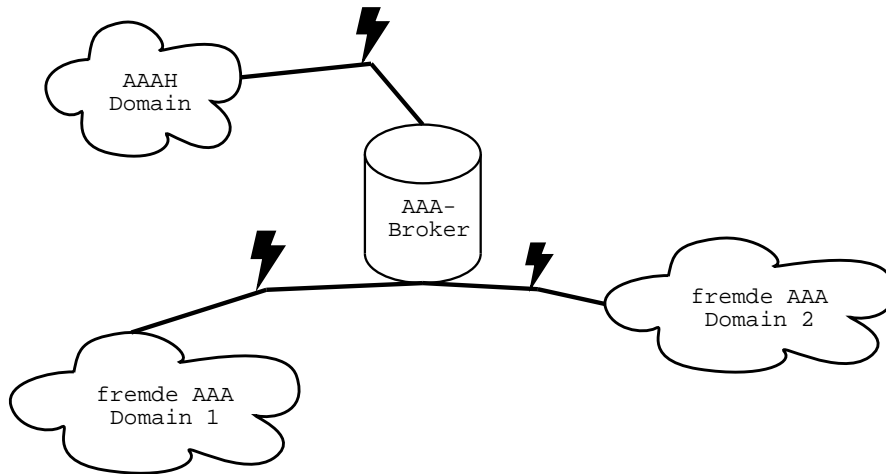


Abbildung 2.10: AAA- Broker Struktur

AAAH nachstehende Anforderungen erfüllt werden:

- Der Datenaustausch zwischen AAAL und AAAH muss über ein standardisiertes Protokoll ablaufen.
- AAAL und AAAH müssen sich gegenseitig vertrauen und authentifizieren.
- Die Verbindungen zwischen AAAL und AAAH müssen verschlüsselt werden.
- Accountingdaten müssen zeitnah und sicher zur AAAH übertragen werden.
- Die Autorisation von bestimmten Diensten muss explizit möglich sein.
- Ein zeitnaher Wechsel von einer Domain zur nächsten muss unterstützt werden.

Damit sich AAA- Server unterschiedlicher Domains gegenseitig vertrauen und authentifizieren können, wäre es notwendig, dass jeder AAA- Server einer Domain mit jedem anderen AAA- Server ein Geheimnis austauschen muss. Bei diesem System würde die Zunahme der AAA- Domains ein quadratisches Wachstum der Vertrauensstellungen bedeuten.

Ein Beispiel: Für 100 AAA- Domains die untereinander Vertrauensstellungen bilden möchten, müssten in jeder einzelnen AAA- Domain 4950 verschiedene Geheimnisse verwaltet werden. Es liegt ein quadratisches Wachstum vor. Um mit n Kommunikationspartnern eine Vertrauensstellung zu bilden, benötigt man $(n * (n-1))$ Vertrauensbeziehungen und jeder Partner müsste $\frac{n}{2} * (n - 1)$ Geheimnisse verwalten.

Um dies zu vermeiden, wird die Verwendung von AAA- Brokern empfohlen, so dass jeder AAA- Server nur noch mit einer begrenzten Anzahl von Brokern eine Vertrauensstellung bilden muss. Die Brokerstruktur könnte hierarchisch mehrschichtig sein, d.h. ein Broker kann AAA- Nachrichten an den zuständigen übergeordneten Broker weiterreichen. AAA- Anforderungen werden dann nicht direkt an die AAAH übertragen, sondern zum Broker geschickt, und dieser leitet sie weiter zur AAAH.

2.4.1 Verbindung zwischen Nutzer und AAA- Agent

Folgende Anforderungen werden an Protokolle für die Authentifizierungsinformationsübermittlung vom Nutzer zum AAA-Agenten gestellt:

- Unterstützung mehrerer Authentifizierungsmechanismen,
- Netzwerkgeräte der Schicht 3 bzw. 4 müssen als AAA- Agent agieren können,
- Nutzer muss AAA- Agenten lokalisieren können,
- Erweiterungsmöglichkeit für zukünftige Authentifizierungsmethoden.

Zur Erfüllung dieser Anforderungen bietet sich ein Protokoll an, das ausschliesslich auf die reibungslose Kommunikation zwischen Nutzer und Agent spezialisiert ist. In diesem Protokoll wären keine Authentifizierungsmethoden festgelegt, sondern die Authentifizierungsdaten selbst würden als Verbund entgegengenommen und an den AAA- Server weitergeleitet. Der AAA- Agent benötigte demzufolge keine Kenntnis über das verwendete Authentifizierungsprotokoll, er diene lediglich als Vermittlungsstelle zwischen Nutzer und AAA- Server.

2.4.2 Verbindung zwischen AAA- Agent und AAA- Server

Diese Voraussetzungen müssen durch Protokolle zur Kommunikation zwischen AAA- Agent und Server erfüllt werden:

- Flexibilität in der Unterstützung von Authentifikationsmethoden und Accountingdaten,
- sicherer, zuverlässiger, verschlüsselter Datenaustausch zum AAA- Server,
- zuverlässiges Verhalten bei Ausfall eines AAA- Servers (z.B. automatische Suche des Backup-servers),
- AAA- Server muss zeitnah Klientenautorisationen widerrufen können, die dann vom AAA- Agent umgesetzt werden.

Am Beispiel des RADIUS Protokolls wird die flexible Unterstützung von unterschiedlichen Authentifikationsmethoden deutlich. RADIUS wurde ursprünglich als Authentifizierungsprotokoll für Telefoneinwahlzugänge entwickelt. Inzwischen wird es für die unterschiedlichsten Authentifizierungsaufgaben eingesetzt. Dazu wurde nicht das Protokoll als solches verändert, sondern lediglich die verwendeten Datentypen entsprechend ergänzt.

2.5 Vorgeschlagene Protokolle für eine AAA- Infrastruktur

Der Aufbau einer globalen AAA- Infrastruktur erfordert auch die Entwicklung von neuen Protokollen, die alle geforderten Eigenschaften besitzen. Zur Anwendung vorgeschlagen wurden zwei Protokolle:

- Diameter [CALHOUN 2001],
- RADIUS(v2) [RIGNEY 2000].

Derzeit wird von den IETF Arbeitsgruppen als Grundlage für die AAA- Infrastruktur das Diameter Protokoll favorisiert. Aufgrund meiner Vermutung, dass sich in Zukunft tatsächlich das Diameter Protokoll durchsetzt, möchte ich lediglich auf dieses näher eingehen.

2.5.1 Das Diameter Protokoll

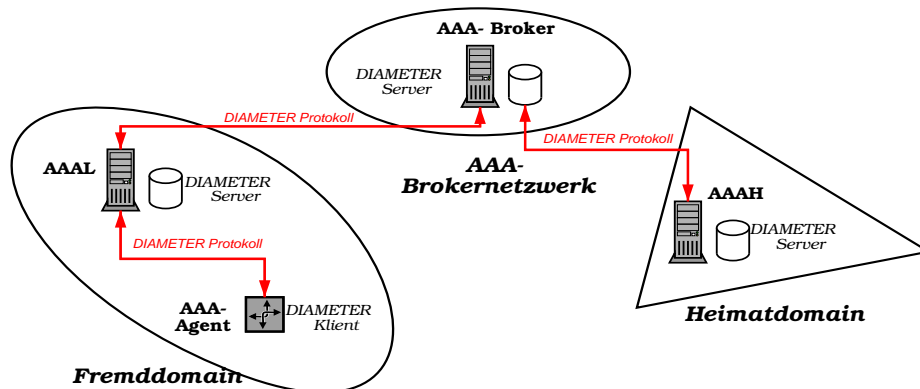


Abbildung 2.11: Diameter als AAA- Infrastruktur

Das Diameter Protokoll stellt eine Neuentwicklung dar, als dessen Basis das RADIUS Protokoll dient.

Es bietet verschiedene Vorteile:

- durch einfache Umsetzung abwärtskompatibel zu RADIUS und TACACS+ [CAREL 1997] (Translation Agent),
- das Grundprotokoll beschreibt nur den grundsätzlichen Aufbau des Datenaustauschs zwischen Diameter Klient /Server,
- einfache Erweiterbarkeit durch modularen Aufbau, vier vorgegebene Grund- Erweiterungen sind Voraussetzung für Server.

Ein Diameter Server muss neben dem Grundprotokoll wenigstens diese 3 Erweiterungen unterstützen:

- NASREQ [CALHOUN2 2001],

- Mobile IP [CALHOUN3 2001],
- CMS Security [CALHOUN4 2001].

Eine Klientenanwendung sollte neben dem Grundprotokoll mindestens die Mobile IP und die NASREQ Erweiterung beherrschen.

Eine Besonderheit des Diameter Standard ist die Vorgabe des SCT- Protokolls [STEWART 2000] als Verbindungsprotokoll zwischen Klienten und Servern. TCP als Verbindungsprotokoll stellt dagegen eine Übergangslösung dar. Durch die Nutzung von SCTP ergeben sich folgende Vorteile:

- zuverlässiger, Nachrichten orientierter Datentransport,
- Multi-streaming,
- Multi-homing.

2.5.1.1 Protokollaufbau

Das Diameterprotokoll setzt sich aus einem Kopf fester Länge und den angefügten AVP (Attribute Value Pairs) zusammen.

| | |
|-----------------------|----------------|
| 0.....31 | |
| version | Message Length |
| R P E r r r r r | Command-Code |
| Vendor-ID | |
| Hop-by-Hop Identifier | |
| End-to-End Identifier | |
| AVPs..... | |

Tabelle 2.2: Diameter Kopf Format

Die AVP beginnen ebenfalls mit einem festen Kopf, gefolgt von speziellen AAA- Werttypen. Sie müssen für jede neue Anwendung definiert werden. Eine Diameter AAA- Nachricht besteht also aus dem Kopf und einer definierten Anzahl von angehängten AVP.

| | |
|-----------------|------------|
| 0.....31 | |
| AVP Code | |
| V M P r r r r r | AVP Length |
| Vendor-ID | |
| Data... | |

Tabelle 2.3: AVP Kopf Format

Mit diesem universellen Protokollaufbau ist gewährleistet, dass verschiedenste Authentifizierungsmechanismen und Autorisationsnachrichten ohne Protokolländerungen unterstützt werden können. Es ist lediglich eine Ergänzung der bestehenden AVP- Codes erforderlich. Diameter Klienten, die neue AVP- Codes nicht kennen, ignorieren diese oder handeln mit dem Diameterserver bekannte Mechanismen aus.

Deshalb können Diameter Klienten mit begrenzter Hardwareausstattung auch nur eine Unter- menge an Diameterfunktionen unterstützen.

2.5.2 Protokolle für die Nutzer zu AAA- Agentenverbindung

Für die Verbindung zwischen AAA- Agent und Nutzer stehen je nach Einsatzzweck verschiedene Protokolle zur Verfügung. Das hängt mit der unterschiedlichen historischen Entwicklung der Dienste zusammen und damit, auf welcher OSI- Schicht die Kommunikation stattfindet. So läuft die Verbindung beim in 2.2.1 vorgestellten Portmanager zum AAA- Agent respektive Diameter Klient über ein Subnetzprotokoll ab.

2.5.2.1 Das Extensible Authentication Protokoll (EAP)

Das EAP wurde ursprünglich für PPP entwickelt, um sich nicht während des Verbindungsaufbaus auf einen speziell vordefinierten Authentifizierungsmechanismus festlegen zu müssen. Es beschreibt in einem einfachen Protokoll den Austausch der Authentifizierungsdaten vom Klient zum AAA- Agent und der Antwort vom AAA- Agent zum Nutzer. Dabei können beliebige Authentifizierungsmechanismen (Kerberos, SecurID) benutzt werden. Das EAP besteht aus einem

| | | |
|----------|-----------------|--------|
| 0.....31 | | |
| Code | Identifier | Length |
| Type | Type- Data..... | |

Tabelle 2.4: EAP Paketaufbau

Paketkopf und einem Authentifizierungsdatenfeld. Mit “Type” wird der verwendete Authentifizierungsmechanismus bezeichnet, danach folgen dann die spezifischen Daten (Type- Data).

EAP wird entweder in Verbindung mit PPP verwendet oder direkt zum Authentifizierungsdatenaustausch in anderen Protokollen, so etwa in IEEE 802.1X.

2.5.2.2 “Port Based Network Access Control” IEEE 802.1X

Der IEEE- Standard 802.1x beschreibt ein Protokoll, welches mit Hilfe von EAP eine Nutzerauthentifizierung auf Subnetzebene vornimmt. Mit diesem Protokoll kann, geeignete Switches vorausgesetzt, bereits am Netzwerkport eine Authentifizierung vorgenommen und nicht authentifizierten oder autorisierten Nutzern der Zugriff auf das Netzwerk verweigert werden. Der Switch fungiert dabei als AAA- Agent, der als RADIUS oder Diameter Klient in den EAP Nachrichten enthaltene Authentifizierungsdaten an den AAA- Server weiterleitet und bei positiver Rückantwort dem Nutzer den Netzwerkzugang gestattet.

Mangels eines geeigneten Authentifizierungsverfahrens im IEEE Wireless LAN Standard 802.11 (s.a. 3.1.4.2) wurde vorgeschlagen, 802.1X auch für Wireless LANs einzusetzen, da es vergleichbare Subnetzeigenschaften aufweist (s. Subnetzprotokolle der IEEE 802 Familie).

Im Falle von WLANs bietet 802.1X neben der Authentifikation weitere Vorteile wie :

- WEP Schlüsselmanagement,
- Unterstützung von Roaming bei Verwendung eines NA- Identifikators,
- managebarer unauthentifizierter Netzwerkzugang für öffentliche WLAN, z.B. durch Übermittlung einer Kreditkartennummer über das EA- Protokoll.

Neu erworbene Hardware, wie z.B. Access Points, unterstützen IEEE 802.1X schon, bzw. ist eine entsprechende Nachrüstung möglich. Bereits in Gebrauch befindliche Switches lassen sich hingegen in den seltensten Fällen erweitern, demzufolge müssten neue Hardwarekomponenten angeschafft werden. Daher vermute ich, dass sich 802.1X in Funk- LANs schneller durchsetzt als in bestehenden kabelgebundenen Infrastrukturen. Alternativ werden in bereits existierenden Netzen Programme wie das Portmanagersystem Verwendung finden, die keine Investitionen nach sich ziehen.

Der Protokollablauf beim Wireless LAN wäre folgender (s.a. Abbildung 2.12) :

1. Ein Klient baut über das 802.11 Subnetzprotokoll eine Verbindung zum AP auf (association).
2. Ein Klient sendet eine EAP Start Nachricht an den AP.
3. Der AP fordert Identitätsdaten (z.B. den NAI) an.
4. Klient sendet den NAI.
5. Der AP leitet Daten an den AAA- Server weiter und erhält die Challenge, welche an den Klient weitergeleitet wird.
6. Klient antwortet mit Response, diese wird vom AAA- Server geprüft und der AP erhält eine "Accept" Nachricht.
7. Der Klient kann auf das Netzwerk zugreifen.
8. Der AP kann dem Klienten über eine verschlüsselte EAP Methode (z.B. EAP-TLS) WEP Schlüssel übermitteln.

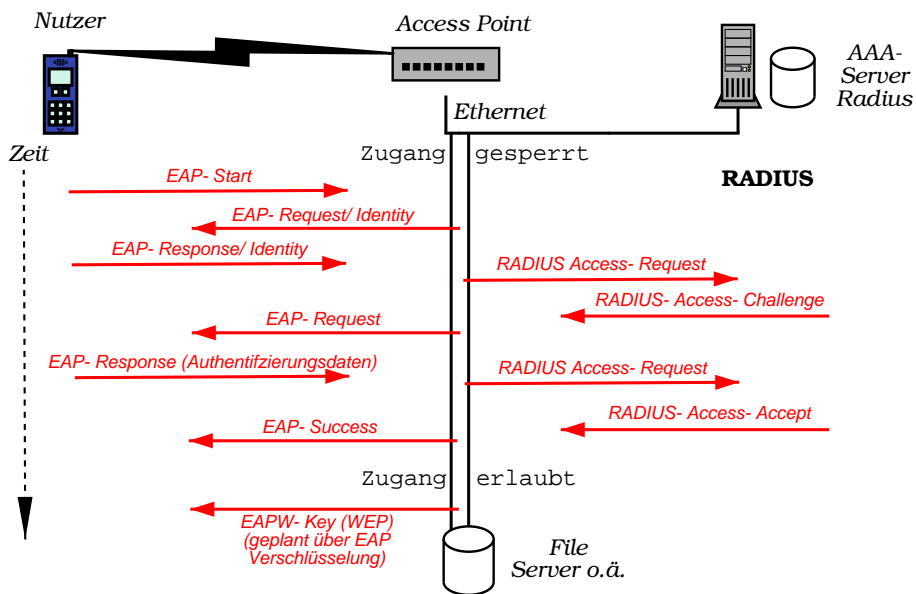


Abbildung 2.12: 802.1X Protokollablauf (nach [ABOBA 2000])

2.5.2.3 PPP

Das Point- to- Point Protokoll ist mit seiner Unterstützung für PAP und CHAP die klassische Authentifizierungsvariante für Telefoneinwahlzugänge. Die Authentifizierung wird vor Gewährung des Zugangs zur gewünschten Ressource durchgeführt. Vorhandene Network Access Server (NAS) bzw. Einwahlrouter können als RADIUS- oder TACACS- Klienten agieren. Durch eine Protokollumsetzung zwischen bestehenden Protokollen und Diameter haben sie Zugang zur AAA-Infrastruktur.

Nachteilig wirkt sich bei PPP allerdings der relativ hohe Bedarf an Protokollinformationen (Overhead) aus, was sich z.B. bei ADSL- Anbindungen, die über PPPoE realisiert werden, durch eine geringere Nutzdatenbandbreite bemerkbar macht.

2.5.2.4 Mobile IPv6

Im Gegensatz zu Mobile IPv4 wird in der IPv6 Variante kein Foreign Agent (FA) mehr vorausgesetzt, der auch die Rolle des AAA- Agenten übernommen hätte. Diese Rolle spielt nun das nächstgelegene Netzwerkgerät (z.B. Router) oder ein Wireless LAN Access Point der Schicht 3/4. Zu diesem Zweck werden Erweiterungen des Router Advertisements (RA siehe 4.3.2.1) und die Entwicklung neuer ICMPv6 Optionen notwendig.

Protokollablauf im Falle eines Routers als AAA- Agent:

1. Ein mobiler Klient empfängt RA mit gesetzter AAA- Option.
2. Er antwortet mit AAA- ICMPv6 Option und übergibt NAI und Authentifizierungsdaten.
3. Router extrahiert AAA- Daten, sendet sie als Diameter Klient an die Diameter- AAA- Infrastruktur
4. Router erhält z.B. positive Antwort und gibt Ressourcen für den mobilen Klienten frei (aktiviert z.B. das Packet forwarding)

2.5.2.5 DHCPv6 Erweiterung

In [MUKHERJEE 2001] wird eine DHCP Erweiterung vorgeschlagen, mit der ein DHCP- Server als AAA- Agent agieren kann. Dazu sendet der Klient eine AAA- Option in seinem DHCP- Request und der DHCP- Server leitet die AAA- Daten als Diameter Klient an die AAA- Infrastruktur weiter. Nach erfolgter Authentifizierung / Autorisation werden dem Klienten entsprechende Informationen wie die IP- Adresse übergeben.

2.5.2.6 DHCPv6 Relay

Um die momentan praktizierte Methode, PPP(oE) als Verbindungs- und Authentifizierungsprotokoll auch bei Breitbandkabelverbindungen (z.B. ADSL) einzusetzen, abzulösen, wurde vorgeschlagen, ein DHCPv6 Relay als AAA- Agenten agieren zu lassen.

Der Unterschied zum DHCP- Server AAA- Agent besteht darin, dass eine beliebige Netzwerkkomponente, die als DHCP- Relay agiert, direkt AAA- Anfragen an die Infrastruktur sendet und Ergebnisse umsetzt. Ein weiterer Vorteil ist, dass Accountingdaten, die in der Netzwerkkomponente

anfallen, direkt an die AAA- Server übergeben werden können. Der Ablauf der Kommunikation zwischen Relay und Klient ist genauso, wie in 2.5.2.5 beschrieben.

2.6 Mechanismen zum Schutz der Privatsphäre des Nutzers

Ebenso wie versucht wird, den Nutzer eindeutig zu identifizieren, müssen folgende Punkte beachtet werden, um die Privatsphäre des Nutzers zu schützen:

- Authentifizierungsdaten und Identifikator werden nur an den Kommunikationspartner übertragen, der an der Authentifikation primär beteiligt ist.
- Es muss eine wie auch immer geartete Verschlüsselung zwischen den AAA- Servern zum Einsatz kommen.
- Der Aufenthaltsort des Nutzers bei Einsatz von Mobile IP muss bei der Kommunikation zwischen AAA- Agent und AAAH verschleiert werden.
- Auch Accountingdaten sind sensible Daten und sollten nur über sichere Verbindungen übertragen werden.

Mit der Einführung von IPv6 könnte jedem Gerät eine eindeutige Adresse zugeordnet werden. Auch wenn diese Adresse nicht unmittelbar einem speziellen Nutzer zugeordnet wäre, liessen sich Rückschlüsse auf den Nutzer ziehen. Dies wäre besonders bei Geräten die gewöhnlich nur von einem bestimmten Nutzer genutzt werden, möglich. Die eindeutige Adresse muss bei jeglicher Kommunikation über das Internet zwingend mit übertragen werden. Folglich wäre es für Datensammler, z.B. Werbebannervermarkter wie "DoubleClick", noch einfacher als bisher, Profile von bestimmten Nutzern zu erstellen. Für dieses Problem gibt es allerdings einen Lösungsansatz.

2.6.1 Automatische Veränderung der eigenen IPv6 Adresse

In [NARTEN 2001] wird eine Methode vorgeschlagen, die es erlaubt, einen Interface Identifikator (siehe 4.2) anhand einer Zufallszahl (z.B. MD5- Hashwert) periodisch neu zu wählen. Die so erhaltene neue IPv6 Adresse wird für ausgehende Verbindungen eingesetzt, so das es unmöglich ist, gesammelte Daten über eine eindeutige unveränderliche IPv6 - Adresse zusammenzuführen.

Dieses Vorgehen widerspricht nicht den IPv6 Standards. Es muss lediglich als default- Adresse immer eine neu generierte Adresse gewählt werden. Dies könnte man dadurch erreichen, dass die Gültigkeit der generierten Adressen nach einer kurzen zufälligen Zeit abläuft.

Eine weitere, noch etwas elegantere Möglichkeit ist der Einsatz von DHCPv6, um zufällige Adressen zu verteilen. Das könnte über eine Reconfigure- Request geschehen. Ein Vorteil dieser Lösung ist, dass anhand der Logdatei auch weiterhin nachgewiesen werden kann, welcher Klient zu welchem Zeitpunkt welche Adresse genutzt hat.

Kapitel 3

Vergleich bestehender AAA- Infrastrukturen mit geplantem neuen Infrastrukturaufbau

3.1 Bestehende AAA- Strukturen am Beispiel TU- Chemnitz

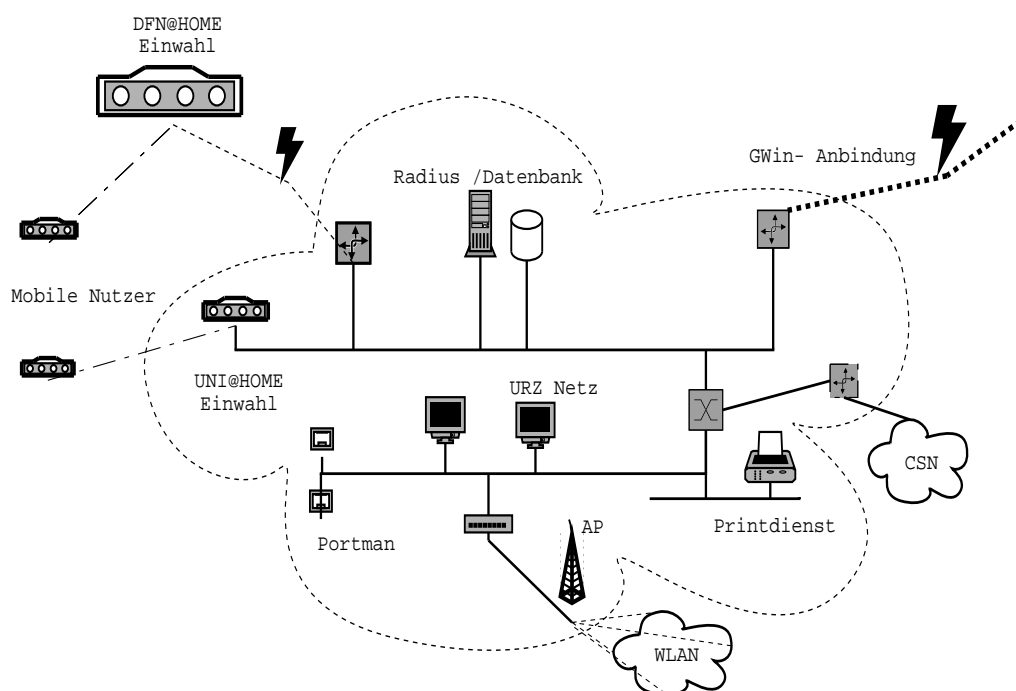


Abbildung 3.1: URZ Netzwerkstruktur

Anhand der Netzinfrastruktur der TU- Chemnitz möchte ich exemplarisch zeigen, wie die bestehenden AAA- Strukturen ineinander greifen. Die Infrastruktur der TU-Chemnitz bietet sich als Vergleichsobjekt an, da sie einer Netzinfrastruktur eines mittleren Unternehmens in Grösse

und Nutzerzahl entspricht.

3.1.1 Die AAA- Zentrale

Kernstück der universitären AAA- Struktur ist die zentrale Nutzerdatenbank. Sie stellt die Authentifizierungs- und Autorisierungsdaten für folgende Dienste bereit:

- Network Information System (NIS) für die Anmeldung an URZ- Rechnern
- AFS/ Kerberos Server
- Windows NT SAM Datenbank für die Anmeldung an Windows NT Rechnern
- RADIUS Einwahlserver
- Cronos Magnetkarten - Zugangssystem
- HTTP Authentifikation für geschützte WWW- Dokumente
- Host- Berechtigungssystem anhand von MAC- Adressen

Dabei werden aus der Nutzerdatenbank die für jeden Dienst erforderlichen Konfigurationsdateien erstellt bzw. greifen einige Dienste direkt auf die Nutzerdatenbank zu. Die Erstellung der Konfigurationsdateien erfolgt durch eigene Werkzeuge bzw. Shell- Skripte.

Alle im universitären Netz angebotenen Dienste nutzen die gleichen zentralen Authentifizierungsdaten. Änderungen darin können nur zentral vorgenommen und dann an alle Dienste propagiert werden.

3.1.2 Druckdienst

Der universitätsweite Druckdienst stellt eine Erweiterung des UNIX- Druckdienstes über Shell - Skripte dar, die den Nutzer anhand des Nutzerkennzeichens authentifizieren und durch Prüfung seines Druckkontos den Druckauftrag autorisieren.

3.1.3 Chemnitzer Studenten Netz - CSN

Das Chemnitzer Studenten Netz (CSN) stellt für das URZ eine eigenverantwortliche Netzwerkdomäne dar. Zwischen URZ und CSN besteht eine Vertrauensbeziehung dahingehend, dass nur autorisierte Nutzer Zugang zu den bereitgestellten Netzwerkressourcen erhalten. Das CSN betreibt eine eigene AAA- Infrastruktur, bei der jeder Nutzer zur Identifikation die Hardwareadresse seiner Netzwerkkarte registriert. Die Autorisierung erfolgt auf der Subnetzsicht durch entsprechend konfigurierte Switches.

3.1.4 Unterstützung von mobilen Nutzern

Das URZ unterstützt mobile Nutzer auf dem Gelände bzw. in den Gebäuden der TU- Chemnitz. Zusätzlich wird eine Einwahl über das Telefonnetz angeboten.

3.1.4.1 Das Portmanagersystem

Das Portmanagersystem ist eine Eigenentwicklung der TU- Chemnitz und verwaltet öffentlich zugängliche Netzwerkports. Um die nachfolgend genannten Dienste anbieten zu können, nutzt es sowohl die zentrale Nutzerdatenbank als auch die Hardwareadressdatenbank.

- Die Authentifizierung erfolgt anhand von Nutzernamen und Passwort. Die Autorisierung wird über unterschiedlich privilegierte Nutzergruppen vorgenommen, um Zugang zu Netzwerkressourcen zu erlangen.
- Nicht authentifizierte Nutzer erhalten keinen oder nur beschränkten Zugang zum Netzwerk.

Der AAA- Agent des Portmanagersystems ist eine spezielle Firewall, die die AAA- Anforderungen weiterleitet und Netzwerkkomponenten der Schicht 2 steuern kann.

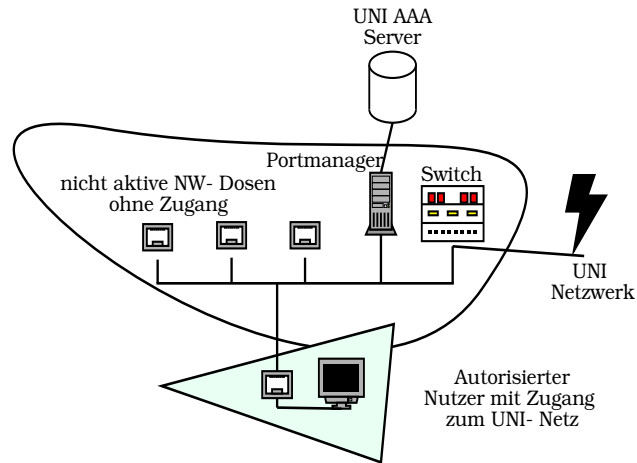


Abbildung 3.2: Portmanager Prinzip

3.1.4.2 Wireless LAN Infrastruktur

Für die Funknetzinfrastruktur nach dem IEEE 802.11b Standard wird zur Zeit eine ähnliche Lösung wie das Portmanagersystem entwickelt. Als AAA- Agent fungiert wieder ein spezielles Firewallsystem, das nur authentifizierten und autorisierten Nutzern den Zugang über den Access Point (AP) gestattet.

Die momentan praktizierte Authentifikationsmethode basiert auf der Registrierung der Hardwareadressen (MAC) der WLAN Karten. Zugang erhalten demzufolge nur Nutzer mit Wireless Adapterkarten deren MAC- Adresse bekannt ist.

Weiterhin wäre die Nutzung der im IEEE 802.11 Standard beschriebenen Authentifizierungs- und Verschlüsselungsmethode "Wired Equivalent Privacy" (WEP) möglich. Bei dieser Methode wird an jeden berechtigten Nutzer ein Schlüssel verteilt. Er dient zur Authentifizierung des Nutzers gegenüber dem AP und als Initialisierungsschlüssel für die Verschlüsselung der Kommunikation zwischen Nutzer und Access Point. Um mit dieser Methode einen einzelnen Nutzer zu identifizieren und zu authentifizieren, benötigt man zusätzlich die MAC- Adresse des Nutzers.

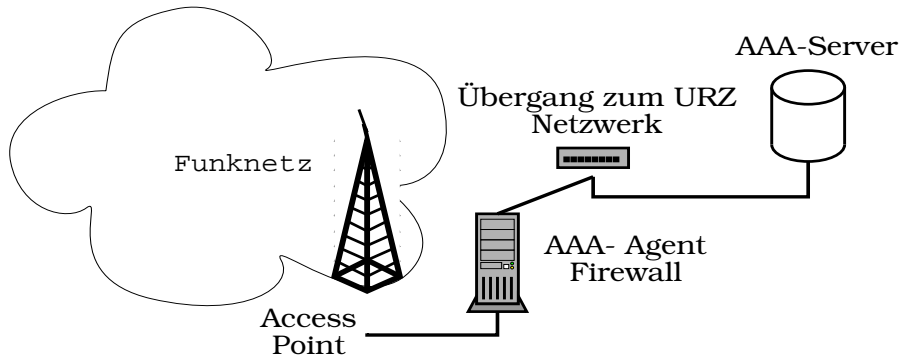


Abbildung 3.3: Funknetz AAA- System

Dieses Verfahren ist umständlich und unsicher. Mit Airsnort [AIRSNORT 2001] wurde ein Programm entwickelt, das in der Lage ist, aus dem abgehörten Netzwerkverkehr den WEP Schlüssel zu reproduzieren. Mit diesem Schlüssel ist es dann möglich, verschlüsselten Netzwerkverkehr mitzuhören sowie MAC- Adressen anderer Nutzer aufzuzeichnen. Angreifer könnten unter Verwendung dieser MAC- Adressen eine andere Nutzeridentität annehmen (bei einigen WLAN- Karten ist es möglich, die MAC- Adresse zu ändern) und in das Netzwerk hinter dem AP eindringen.

3.1.4.3 UNI@HOME Einwahlzugang

Beim UNI@HOME Telefoneinwahlzugang handelt es sich um eine klassische RADIUS Authentifizierung und Autorisierung. Der Einwahlserver agiert als AAA- Agent und leitet die per PPP empfangenen Authentifizierungsdaten an den zentralen RADIUS- Server weiter. Nach erfolgreicher Authentifizierung / Autorisation bekommt der Nutzer eine IP- Adresse aus dem Adressbereich des URZ.

3.1.4.4 DFN@HOME Einwahlzugang

Der DFN@HOME Zugang stellt eine Besonderheit dar, denn ein externer Provider (Talkline) bietet in Zusammenarbeit mit dem DFN- Verein deutschlandweit direkten Zugang in Universitätsnetzwerke an. Dabei stellt Talkline lediglich seine Einwahlressourcen zur Verfügung, der Nutzer erhält IP- Adressen aus dem Uni- Adressraum, und die Internetverbindung erfolgt über die DFN- Anbindung der Uni. Ein Vorteil dieser Lösung ist, dass innerhalb des Universitätsnetzes Dienste genutzt werden können, die Authentifizierung anhand einer Uni eigenen IP- Adresse erfordern. Der technische Ablauf ist, wie folgt:

- Der Student bildet aus seinem URZ Nutzerkennzeichen und der Erweiterung @hrz.tu-chemnitz seinen NAI und meldet sich bei DFN@HOME an.
- Die Einwahl erfolgt über eine bundesweit einheitliche Rufnummer. Als PPP- Benutzername wird der NAI und als Passwort das normale Uni- Login- Passwort verwendet.
- Der Talkline- RADIUS- Server agiert als RADIUS- Proxy und leitet die Anfrage anhand des NAI Domainnames über einen direkten IP- IP Tunnel in das URZ- LAN an den URZ- RADIUS Server weiter.

- Der URZ- RADIUS Server schickt das Authentifizierungsergebnis an den Talkline- RADIUS- Server und übergibt gleichzeitig die zu vergebene IP- Adresse aus dem URZ- Adressraum.
- Jeder weitere Datenverkehr des Nutzers erfolgt über den Tunnel ins URZ.

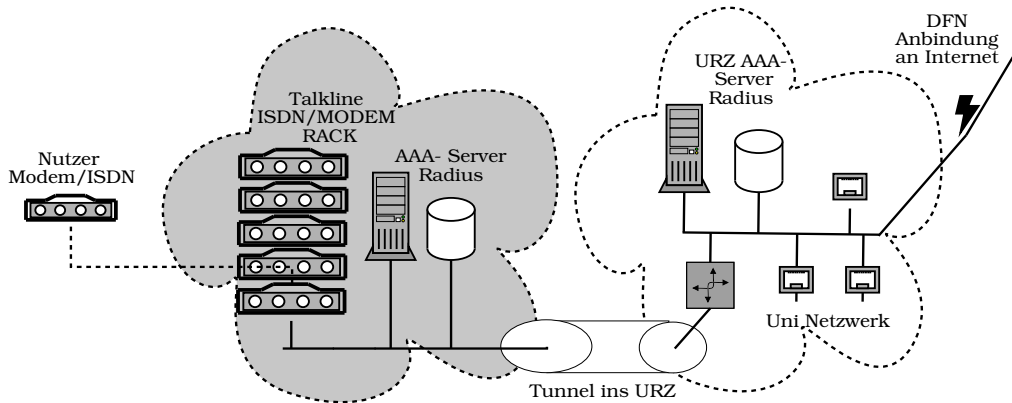


Abbildung 3.4: DFN@HOME Einwahl

3.2 Die neue AAA Infrastruktur

Nachfolgend soll gezeigt werden, wie die zukünftige AAA- Infrastruktur des URZ durch Einsatz von Diameter, Mobile IP und AAA- Agenten aussehen könnte. Folgende Voraussetzungen werden als gegeben angenommen:

- der DFN Verein betreibt AAA- Broker für DFN- Mitglieder (Universitäten, Forschungseinrichtungen),
- für alle genannten Anwendungen existieren spezielle AVPs für Diameter,
- alle Dienstleister agieren als Diameter Klienten,
- Netzwerkinfrastruktur ist IPv6 basiert.

Die AAA- Zentrale stellt dann einen Diameter- basierten Server dar. Durch vorgeschaltete Protokollumsetzer wird eine RADIUS Kompatibilität erreicht. Der AAA- Server besitzt eine Vertrauensbeziehung (SA) zum DFN- AAA- Broker.

3.2.1 Der Druckdienst

Der Druckdienst allgemein würde auf dem Internet Printing Protocol (IPP) basieren, d.h. jeder Drucker ist IPP fähig und würde über einen IPP- Server zentral verwaltet. Sämtliche Drucker würden als AAA- Agenten agieren, IPP Druckaufträge entgegennehmen, den Druckauftrag anhand von speziellen Diameter AVPs authentifizieren und autorisieren. Nach erfolgtem Druck würden Accountingdaten zur AAA- Zentrale übertragen.

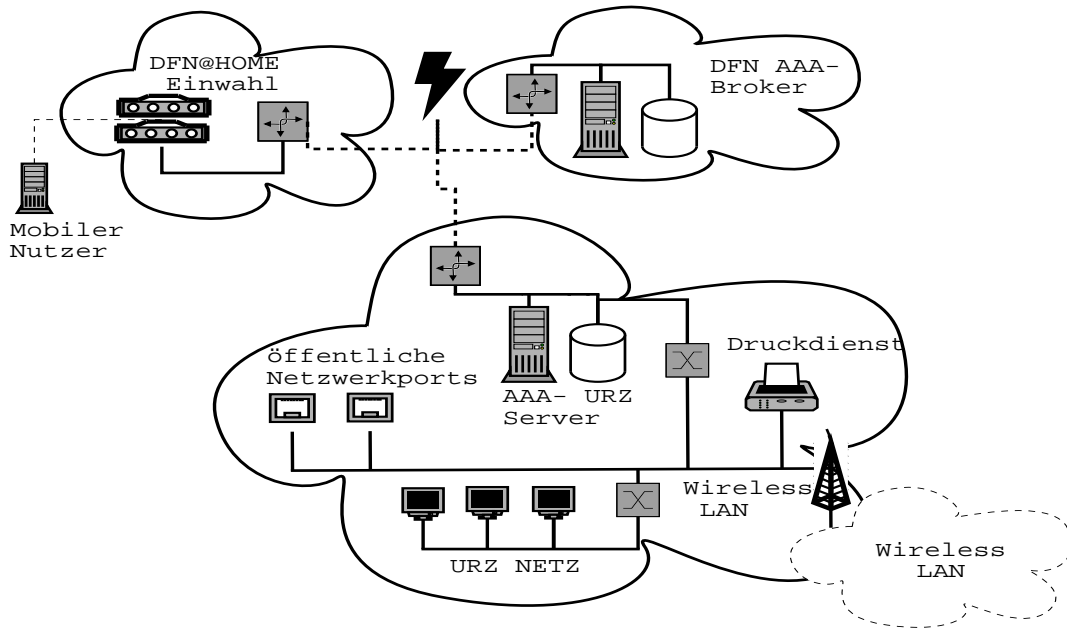


Abbildung 3.5: vorgeschlagene AAA- Infrastruktur

3.2.2 Nutzerauthentifizierung

Auch beim Login in URZ- Rechner könnte direkt auf die Diameter Infrastruktur zurückgegriffen werden. Durch die Nutzung z.B. eines Diametermoduls für das Pluggable Authentication Modul (PAM)Login- System könnte auf NIS verzichtet werden. Vorausgesetzt, es werden durch das Diametermodul auch ähnliche Informationen geliefert.

Auch das Magnetkartenzugangssystem (CRONOS) könnte als Diameterklient arbeiten.

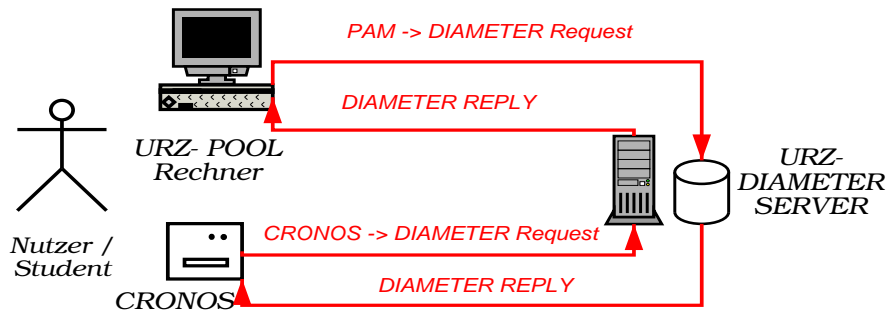


Abbildung 3.6: Nutzerauthentifizierung per Diameter

3.2.3 Unterstützung von mobilen Nutzern

Die Unterstützung von mobilen Nutzern hat sich grundlegend verändert, sie ist schon heute ein Grundbestandteil des neuen Systems, deshalb sind hier keine Spezialentwicklungen mehr nötig.

3.2.3.1 Verwaltung öffentlicher Ports

Netzwerkgeräte (z.B. Router) erhalten dann durch die AAA- Agentenfunktionalität auch die Portmanagerfunktionalität. Sie fordern über Routeradvertisements angeschlossene mobile Geräte zur Authentifikation auf, leiten die AAA- Daten an den Diameterserver weiter und bei erfolgter Authentifizierung und Autorisation werden Pakete der entsprechenden Geräte weitergeleitet. Auch Einschränkungen bestimmter Dienste für fremde Nutzer ist durch Filterung möglich.

Durch Mobile IP ist es z.B. auch für URZ - Mitarbeiter ohne Umschalten in das eigene LAN möglich, in ihrer gewohnten Umgebung weiterzuarbeiten. Weiterhin könnten über den DFN AAA-Broker, Nutzer fremder Universitäten authentifiziert und Mobile IP Dienste angeboten werden.

3.2.3.2 DFN@HOME

Bei der Nutzung von DFN@HOME könnte zukünftig ein beliebiger ISP gewählt werden, mit dem eine Vereinbarung zur Abrechnung getroffen wurde. Durch die AAA- Infrastruktur bestünde dann die Möglichkeit, Mobile IP und durch bestimmte IP- Adressbereiche geschützte Dienste zu nutzen, sowie in seiner vertrauten Umgebung zu arbeiten. Allerdings ist es unmöglich, für den Netzzugriff ausserhalb der Universität deren Ressourcen bzw. die des DFN zu nutzen. Dieses Problem liesse sich zwar über einen Tunnel vom mobilen Gerät in das URZ lösen, setzt aber voraus, dass der Datentransfer vom ISP zum DFN kostenfrei erfolgt bzw. der ISP und das DFN einen Peering Vertrag unterzeichnet haben.

Deswegen ist es sinnvoll auf den Schutz von Diensten anhand bestimmter IP- Adressen zu verzichten und universellere Authentifizierungsverfahren zu nutzen.

3.2.3.3 Wireless LAN

Hier wird sich höchstwahrscheinlich die Authentifizierung nach dem 802.1X Standard durchsetzen. Der jeweilige Access Point diene dann als AAA- Agent, der über Diameter Authentifizierungsanfragen an den UNI- AAA- Server weiterleitet. EAP bzw. 802.1X könnte gleichzeitig auch zur sicheren Übermittlung für IPsec Schlüssel genutzt werden. WEP dagegen wird wegen der bekannten Sicherheitsmängel nicht zum Einsatz kommen.

3.3 Fazit

Wie man in Abschnitt 3.1 am Beispiel des Universitätsrechenzentrums der TU- Chemnitz sieht, kann bereits mit den heute zur Verfügung stehenden Mitteln eine komplexe AAA- Infrastruktur aufgebaut werden, die die Nutzung verschiedenster Dienste auch ohne mehrfache Authentifizierung ermöglicht.

Am Beispiel des Portmanagers wird allerdings deutlich, dass diese Funktionalität nur unter grossem Aufwand erreicht wird, da es erforderlich ist, entsprechende Schnittstellen und Software selbst zu entwickeln. Bezüglich des Einsatzes mobiler Dienste muss festgestellt werden, dass nur eine domainübergreifende AAA- Infrastruktur deren tatsächliche mobile Nutzung ermöglicht. Denn nicht jeder ISP kann allen angeschlossenen Firmen bzw. Organisationen einen IP- IP Tunnel bieten.

Weiterhin ist erkennbar, dass die vorgeschlagenen Protokolle (Diameter) momentan speziell auf die Anwendung für mobile Dienste oder den Netzwerkzugang ausgerichtet sind. Aber in Hinblick auf eine homogene Gesamtlösung des Problems AAA müssten in die Weiterentwicklung Dienste wie das Dateimanagement und Druckdienste in weitaus grösserem Umfang einbezogen werden. Denn nur so wird gewährleistet, dass die Nutzung jedweden Dienstes gleiche Strukturen und Abläufe aufweist. Dies macht zum einen Entwicklern und Nutzern das Leben langfristig einfacher und erhöht zum anderen in erheblichem Masse Akzeptanz und Erfolg mobiler Dienste.

Kapitel 4

Adresskonfigurationsmechanismen in IPv6

4.1 IPv6 Neuerungen

IPv6 bietet folgende Neuerungen, die für die Adresskonfiguration von besonderer Bedeutung sind:

- Unterstützung von Renumbering,
- beliebige Anzahl von Adressen pro Interface,
- Unterstützung von mehreren Internetprovidern (multihomed).

4.1.1 IPv6 Renumbering

Ein wichtiges Designziel von IPv6 war die Schaffung einer einfachen Möglichkeit zur Ausstattung eines kompletten IPv6- Subnetzes mit neuen Adressen bzw. Adresspräfixen (Renumbering). Das Renumbering erlangt z.B. grosse Bedeutung beim Wechsel des Internet Service Providers. Die Unterstützung einfachen Renumberings basiert auf folgenden Funktionen:

- Adressautokonfiguration,
- Adresspräfixe werden über Router / DHCP propagiert,
- jede Adresse besitzt eine begrenzte Lebenszeit und muss nach Ablauf neu angefordert werden,
- pro Interface sind mehrere Präfixe vergeben,
- Router können sich untereinander neue Präfixe übermitteln,
- DNS Server unterstützt Renumbering (durch dynamische Updates).

Ein typischer Renumbering Ablauf könnte so aussehen:

1. Ein neuer ISP vergibt neue Präfixe.

2. Diese Präfixe werden durch das Router Renumbering Protokoll [CRAWFORD 2000] an die Router verteilt.
3. Die Router verwenden die neuen Präfixe in Routeradvertise Nachrichten.
4. Angeschlossene Hosts nehmen diese Präfixe mit in ihre Adresstabellen auf.
5. Mit Ablauf der Lebenszeit der alten Präfixe werden die neuen benutzt.
6. Dem DNS- Server müssen die neuen Präfixe mitgeteilt werden.

4.1.1.1 Einführung einer Adressgültigkeit zur Renumberingunterstützung

Jede IPv6 Adresse im System, egal ob durch Autokonfiguration oder DHCPv6 bezogen, besitzt zwei Indikatoren die aussagen, wie lange sie gültig ist. Die sog. “preferred lifetime” trifft eine Aussage darüber, wie lange eine Adresse als Quelladresse genutzt werden sollte. Die “valid lifetime” hingegen ist die Zeit, nach deren Ablauf die Adresse ihre Gültigkeit verliert. Anschliessend darf sie nicht mehr als Quelladresse genutzt werden und eingehende Verbindungen zu dieser Adresse sind nicht mehr anzunehmen. Diese Gültigkeitszeiten werden dem System entweder mit den Präfixen aus den Routeradvertisements übermittelt oder sie werden mit den Adressen die vom DHCPv6 Server empfangen wurden, festgelegt.

Mit Ablauf der “preferred lifetime” wird die Adresse entwertet (deprecated). Das bedeutet, sie sollte nicht mehr als Quelladresse zum Einsatz kommen. Allerdings werden eingehende Verbindungen noch akzeptiert und bereits bestehende Verbindungen werden vom geschilderten Ablauf nicht beeinflusst. Sobald eine Adresse sich im Zustand “entwertet” befindet, muss das System versuchen, über Routeradvertisements oder DHCPv6 RENEW Anfragen die Gültigkeit der bestehenden Adresse zu verlängern oder neue Adressen zu erhalten.

Auf diese Weise wird die ständige Neukonfigurierung der Adressen gewährleistet und es kann exakt bestimmt werden, zu welchem Zeitpunkt neue Adressen von allen Systemen des Netzwerkes übernommen werden.

4.1.2 Mehrere IP-Adressen pro Interface

Im Gegensatz zu IPv4 ist es mit IPv6 möglich, jedem Netzwerkinterface direkt, ohne den Umweg über Pseudointerfaces (wie z.B. Linux eth0:1, eth0:2), mehrere IP-Adressen unterschiedlicher Gültigkeit zuzuweisen.

Am Beispiel der Ausgabe des `ifconfig` Befehls eines Linux Systems wird deutlich, dass das Interface `eth0` sowohl zwei global gültige Adressen als auch die obligatorische Adresse mit Verbindungsgültigkeit besitzt.

```
eth0:
Linkverkapselung:Ethernet HWaddr 00:80:C8:48:BC:40 inet addr:192.168.11.1
inet6 Adr: 3ffe:400:100:f202::1/64 Gültigkeit:Global
inet6 Adr: 3ffe:400:100:f101::1/64 Gültigkeit:Global
inet6 Adr: fe80::280:c8ff:fe48:bc40/10 Gültigkeit:Verbindung
UP BROADCAST RUNNING MTU:1500 Metric:1
Empfangene Pakete:55549 Fehler:0 Weggeworfen:0 Überlauf:0 Rahmen:0
Verschickte Pakete:73451 Fehler:0 Weggeworfen:0 Überlauf:0 Rahmen:0
Kollisionen:1742 Sendewarteschlangenlänge:100
Interrupt:12 Basisadresse:0xe000
```

Diese Erweiterung zieht natürlich eine Veränderung der IPv6 Socket Adressstruktur nach sich. Besitzt ein Host mehrere Interfaces, ist nicht mehr eindeutig erkennbar, über welches Interface z.B. ein Host mit lediglich einer Verbindungsgültigkeitsadresse erreicht werden kann, da alle Interfaces Adressen mit Verbindungsgültigkeit haben. Im Falle von IPv4 würde das bedeuten, dass alle Interface Adressen aus dem selben Subnetz sind.

Dieses Problem liesse sich durch ein Versenden auf allen Interfaces umgehen, was der IPv6-Standard aber verbietet. Zur Lösung wird die Socket- Adressstruktur um einen Gültigkeitseintrag (`sin6_scope_id`) erweitert (s.a. [GILLIGAN 1999]) . Mit der `scope_id` ist es möglich, Interfaces, die die entsprechend gültigen Adressen aufweisen, direkt anzugeben um den Empfänger zu erreichen. Im Falle von Verbindungsgültigkeit (link local scope) wird im `scope_id` Feld selbst der Index des entsprechenden Interfaces angegeben.

Weiterhin ergibt sich die Frage, welche der Interfaceadressen als Quelladresse für ausgehende Verbindungen benutzt wird. In der aktuellen Entwicklung [DRAVES 2001] wird dieses Problem wie folgt gelöst: Durch Vergleich der Gültigkeit der Zieladresse mit den Interfaceadressen wird die Interfaceadresse mit der grössten Übereinstimmung, gewählt. Das heisst, standardmässig wird z.B. bei einem Ziel mit site- local Adresse als Quelladresse auch nur die site- local Adresse des Interfaces eingesetzt.

4.2 IPv6 Adressaufbau

Zum besseren Verständnis der Adressvergabeabläufe soll zuerst etwas genauer auf die Adressstrukturen von IPv6 eingegangen werden.

Generell ist eine IPv6 Adresse 128 Bit lang, wobei die ersten Bits den Adresstyp beschreiben, der entweder Unicast, Multicast oder Anycast ist. Weiterhin existieren zwei vordefinierte Adressarten, die un spezifizierte [unspecified] und die Loopback Adresse.

4.2.1 Spezielle Adresstypen

Vordefinierte Adresstypen sind die un spezifizierte Adresse [unspecified address] und die Loopback Adresse.

Die unspesifizierte Adresse besteht ausschliesslich aus Nullen (0:0:0:0:0:0:0) und zeigt an, dass für ein Interface noch keine Adresse existiert bzw. vergeben wurde.

Mit der Loopback Adresse (0:0:0:0:0:0:1) wird das lokale Loopback- Interface bezeichnet, über das Verbindungen zwischen lokalen Prozessen abgewickelt werden.

4.2.2 Unicast Adressen

Der Unicast Adresstyp enthält 3 weitere Typen, die für verschiedene Einsatzszenarien verwendet werden.

4.2.2.1 Globale Unicast Adressen

Bei globalen Unicast Adressen handelt es sich um netzweit eindeutige und routbare IP- Adressen. Die Vergabe von Nummernbereichen erfolgt anhand von Zugehörigkeiten, z.B. zu einem bestimmten Internet Service Provider. Sie enthalten eine eindeutige Identifikation, die sie wiederum anhand von untergeordneten Identifikatoren in weitere Nummernbereiche aufspalten können.

| Netzteil | | | | | Hostteil |
|----------|-----------|-----|-----------|-----------|--------------|
| 3 | 13 | 8 | 24 | 16 | 64 |
| FP | TLA ID | RES | NLA ID | SLA ID | Interface ID |

Tabelle 4.1: IPv6 Adressstruktur

Netzteil

Der Netzteil einer globalen Unicast Adresse besteht aus dem Format Präfix (FP) und drei Netzwerkidentifikatoren, die für die Routingentscheidung benötigt werden und ist vergleichbar mit der Teilnetzbildung bei IPv4- Adressen.

- der Format Präfix (FP) ist für global gültige Unicastadressen 001,
- der Top Level Aggregator (TLA) Identifikator beschreibt die Zugehörigkeit zu übergeordneten Organisationen den wiederum NLAs vergeben,
- der Next Level Aggregator (NLA) Identifikator wird lokalen ISPs und Organisationen zugeordnet, um eine Hierarchie für das Routing zu bilden, dabei kann der NLA in 8 Bit Abschnitte unterteilt werden,
- der Site Level Aggregator (SLA) Identifikator dient zur Subnetzbildung im lokalen Netz.

Ein Beispiel anhand der IPv6 Adresse, 3FFE:5C9:3:1D:200:B4FF:FE32:E6FF:

- es handelt sich um eine Unicastadresse, die ersten Bits sind 001,
- der TLA ist der dem IPv6 Versuchsnetzwerk (6bone) zugewiesene (0x1FFe),
- der NLA ist ein ISP (z.B. die Telekom) (0x5C9:3),
- im Subnetz SLA (1D) des Kunden,
- die den Host mit Adresse 200:B4FF:FE32:E6FF beschreibt.

Hostteil

Interface ID Der Interface ID ist ein Identifikator im IEEE EUI-64 Format, der sich aus der Hardware Adresse (MAC) der OSI- Schicht 2 (z.B. Ethernet) zusammensetzt. Im Beispiel Ethernet wird die 48- Bit Adresse des Netzwerkadapters in einen EUI-64 (Extended Unique Identifier) Identifikator gewandelt. Die eindeutige Überführung wird durch ein Einfügen der Bytes FF:FE zwischen den ersten 3 und den letzten 3 Bytes der MAC erreicht.

Allerdings gibt es auch Hardwareschnittstellen, die nicht über eine eindeutige Hardwareadresse verfügen (z.B. serielle Interfaces). Für diese Interfaces muss eine EUI-64 Adresse generiert werden (z.B. aus einer Zufallszahl). Um eine Möglichkeit der Trennung zwischen generierten und "echten" eindeutigen Adressen zu erhalten, wird bei eindeutigen Adressen das 7. Bit auf 1 gesetzt.

Ein Beispiel: Aus der 48-Bit Hardwareadresse **08:00:20:C0:FF:EE** erhält man durch Einfügen von FF:FE, **08:00:20:FF:FE:C0:FF:EE** und nach Setzen des 7. Bits für eine eindeutige Adresse die global gültige EUI-64 Adresse **0A:00:20:FF:FE:C0:FF:EE**.

4.2.2.2 Site- local Adressen

Site-local Adressen sind vergleichbar mit den privaten Nummernbereichen in IPv4. Sie werden nicht über die Grenzen des eigenen Netzes hinaus geroutet oder weitergeleitet. Sie beginnen mit dem Präfix FEC0.

| Netzteil | | | Hostteil |
|----------|------|-------------------|--------------|
| 10 | 38 | 16 | 64 |
| FEC0 | NULL | SLA (Sub-netz ID) | Interface ID |

Tabelle 4.2: site- local Adressen

4.2.2.3 Link- local Adressen

Link- Local Adressen bestehen aus dem Präfix FE80 und dem Hostteil. Sie sind nur im direkt angeschlossenen Subnetz gültig (Ethernet Broadcast Domain). Sie spielen eine grosse Rolle für die automatische Adressvergabe, da sie direkt aus der MAC der Netzwerkkarte gebildet werden und sofort nach dem Booten mit lokalen Netzwerkkomponenten in Kontakt treten können. Jedes Netzwerkinterface besitzt eine link- local Adresse.

| Netzteil | | Hostteil |
|----------|------|--------------|
| 10 | 58 | 64 |
| FE80 | NULL | Interface ID |

Tabelle 4.3: Link- local Adressen

4.2.3 Multicast Adressen

Multicast spielt in IPv6 eine zentrale Rolle. Aufgaben, die bisher mit Broadcasts erledigt wurden, werden jetzt auf verschiedene Multicastgruppen abgebildet. Der Adressaufbau ist wie folgt, der Präfix FF, danach ein Bit das festlegt, ob eine in IPv6 Multicast Address Assignments [RFC2375]

vordefinierte Adresse (0) oder eine selbstdefinierte (1) vorliegt, anschliessend folgen 2 Bit für die verschiedenen Gültigkeitsbereiche (s. Tabelle).

| | | | |
|-----------|-----|---------------------------|-------------------------------------|
| 8 | 4 | 2 | 112 |
| FF | 0/1 | Gültigkeit <i>[scope]</i> | Multicast Gruppe beginnend beim LSB |

Tabelle 4.4: Multicast Adressaufbau

| Scope ID | Gültigkeit |
|----------|--------------|
| 0 | reserviert |
| 1 | node-local |
| 2 | link-local |
| 5 | site-local |
| 8 | organisation |
| E | global |

Tabelle 4.5: Multicast Gültigkeitsbereiche

So steht im RFC2375 zum Beispiel die Multicast-Adresse **FF05:0:0:0:0:1:3**, mit site-local Gültigkeit (05) für die Multicastgruppe “ alle DHCP Server [ALL DHCP SERVER]” des lokalen Netzes.

4.2.4 Anycast Adressen

Anycast Adressierung ist eine IPv6 Neuerung. Sie kombiniert die Unicast- mit der Multicast-adressierung. Eine Gruppe von Hosts bildet in diesem Fall eine Anycastgruppe. Ein Paket, das an die Gruppenadresse gesendet wird, wird nur einem Mitglied der Gruppe zugestellt, im Gegensatz zu Multicast, wo alle Gruppenmitglieder das Paket erhalten. Die lokalen Router entscheiden anhand der Entfernung (Anzahl HOPs) in ihrer Routingtabelle, welchem Gruppenmitglied das Paket zugestellt wird. Die Routingtabellen werden durch Routingprotokolle ständig angepasst. Anycastadressen haben kein spezielles Adressformat oder Präfixe. Sie sind eine Untermenge der Unicastadressen und auch nicht von Unicastadressen zu unterscheiden.

So wäre es möglich, eine DNS- Anfrage an eine vordefinierte DNS- Anycast Adresse zu richten, und der lokale Router würde die Anfrage an den zuständigen DNS- Server weiterleiten, ohne dass dem Sender die DNS- Serveradresse des Subnetzes bekannt sein muss.

Weiterhin existiert die Möglichkeit, Anycastgruppen auch zur Redundanzschaffung einzusetzen. So könnten z.B. die Router eines Unternehmensnetzwerkes zur redundanten ISP- Anbindung eine Anycastgruppe bilden. Jedem Host im Netzwerk ist dann nur diese Anycast Adresse bekannt. Bei Ausfall eines der Router oder ISP- Anbindungen würde automatisch ein anderer Router die ISP- Anbindung übernehmen.

4.3 Zustandslose Adresskonfiguration von IPv6- Hosts

Die zustandslose Adresskonfiguration (stateless address autoconfiguration) ermöglicht die Konfiguration von IPv6 Host Netzwerkschnittstellen mit gültigen, routbaren IPv6 Adressen ohne administrative Eingriffe oder Konfigurationsserver.

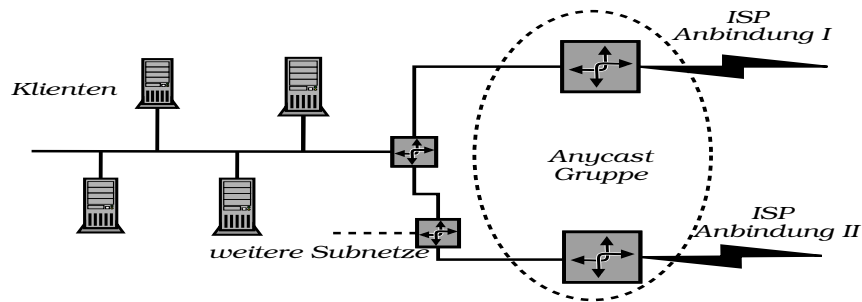


Abbildung 4.1: Anycast Beispiel

Es müssen ausschliesslich lokale Router zur Versorgung des angeschlossenen Netzes mit den nötigen Adress- Präfixen manuell konfiguriert werden.

4.3.1 Prinzip des Ablaufs der zustandslosen Adresskonfiguration

1. Aktivierung der Netzwerkschnittstellen (z.B. nach einem Neustart) mit einer vorläufigen link- local Adresse (siehe 4.2.2.3), die aus der MAC- Adresse der Netzwerkkarte gebildet wurde.
2. Abonnieren der Multicastgruppe "All Nodes".
3. Durch Senden eines Neighbor Solicit ist zu testen, ob die vorläufige Link Local Adresse am lokalen Netzabschnitt eindeutig ist (Duplicate Address Detection (DAD)).
4. Nach erfolgreichem Test, das Interface mit der link- local Adresse aktivieren.
5. Senden eines Router Solicit an die Multicastgruppe "All Routers" und warten auf ein Router Advertisement.
6. Bilden von site- local und globaler Adresse durch Übernahme des Präfixes aus dem Router Advertisement.
7. Falls das "OtherConfigFlag" im Router Advertisement gesetzt ist, anfordern von weiteren Informationen über DHCPv6.
8. Ist die Adresskonfiguration beendet, müssen auch weiterhin Router Advertisements ausgewertet werden, um eventuell Adresspräfixe zu aktualisieren.

4.3.2 Entdeckung benachbarter Hosts (Neighbor Discovery)

Mit dem Festlegen der eigenen IPv6- Adresse ist nur ein Teil der Arbeit erledigt. So ist weiterhin unklar, welche Adresse z.B. der nächste Router hat, diese wird jedoch zur Erzeugung einer default-Route benötigt, oder wie die Hardware- Adressen benachbarter Hosts lauten.

Diese Funktionalität wird durch das Neighbor Discovery Protocol sichergestellt. Es löst den IPv4 Broadcast Mechanismus und ARP für das Ermitteln der MAC- Adresse benachbarter Hosts ab und bietet weitere Mechanismen, um Informationen von lokalen Routern bzw. Hosts zu erhalten.

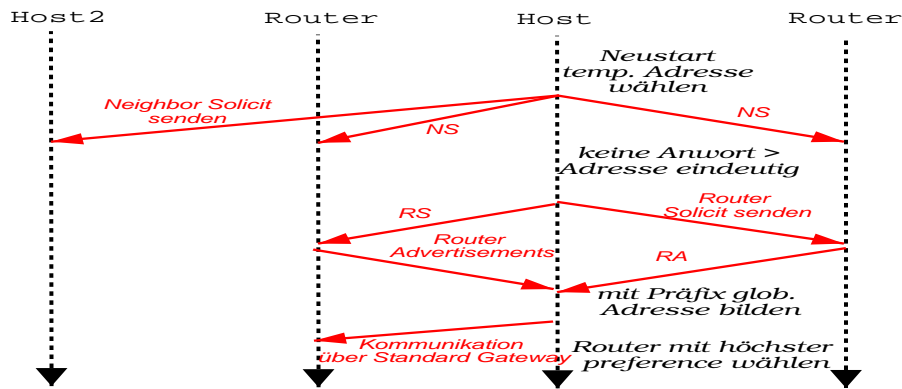


Abbildung 4.2: Zeitlinie zustandsloser Adresskonfiguration

4.3.2.1 Konfigurationsverbreitung über lokale Router (Router Advertisement)

Mit dem Router Advertisement werden im Router vordefinierte Informationen an angeschlossene Netzwerksegmente verbreitet. Hauptsächlich sind das:

- Adresspräfixe verschiedener Reichweite zur Autokonfiguration,
- Adresspräfixe des lokalen Segmentes und
- Informationen über den sendenden Router und dessen Eignung als default- Router des Segments.

Neighbor Discovery Nachrichten und somit auch Router Advertisement Nachrichten sind Subtypen des Internet Control Message Protocol (ICMPv6). Ein Router Advertisement Paket hat folgenden Aufbau:

| | | | | |
|-------------------------------------|------|---|----------|-----------------|
| 0.....31 | | | | |
| Type | Code | | | Checksum |
| Cur Hop Limit | M | O | Reserved | Router Lifetime |
| Reachable Time | | | | |
| Retrans Timer | | | | |
| Options...(z.B. Prefix Information) | | | | |

Tabelle 4.6: Router Advertisement Message Format

Informationen aus dem Router Advertisement spielen die tragende Rolle in der zustandslosen Autokonfiguration, grundsätzlich sind 3 Szenarien zu unterscheiden:

1. kein Router am lokalen Segment, Klient versucht zustandsgebundene Konfiguration,
2. Router vorhanden, aber Bit M bzw. O gesetzt, Klient versucht zustandsgebundene Konfiguration,
3. Router vorhanden, Bit M und O nicht gesetzt, Klient konfiguriert sich mit Informationen aus Router Advertise und dessen Optionen.

Im Anhang eines RA können verschiedene Optionen gesetzt werden, zur Zeit sind folgende definiert:

- Netzwerkadresspräfix und
- maximale Netzwerkpaketlänge (MTU).

| | | | | | |
|--------------------|--------|---------------|---|---|-----------|
| 0.....31 | | | | | |
| Type | Length | Prefix Length | L | A | Reserved1 |
| Valid Lifetime | | | | | |
| Preferred Lifetime | | | | | |
| Reserved2 | | | | | |
| Prefix | | | | | |

Tabelle 4.7: Präfix Information Option

4.4 Zustandsgebundene dynamische Adresskonfiguration von IPv6 Hosts

4.4.1 Entwicklung

Die Motivation für die Entwicklung von Adresskonfigurationsprotokollen war der Mitte der achtziger Jahre verstärkte Einsatz von festplattenlosen Systemen. Diese Systeme benötigen nach dem Start eine gültige IP- Adresse und einen Boot- Server, der die erforderlichen Startprogramme bereitstellt. Die ersten Systeme besaßen keinen batteriegestützten RAM, an dem man Adress-einstellungen manuell hätte vornehmen können, sondern ihr Startprogramm und zusätzliche In-formationen waren im nichtveränderbaren ROM gespeichert. Das einzige eindeutige Unterschei-dungsmerkmal dieser Systeme war die Hardwareadresse ihres Ethernet Adapters [MAC]. Mit dem Reverse Adress Resolution Protokoll [BRADLEY 1998] wurde ein Protokoll entwickelt, das die MAC- Adresse in eine IP- Adresse auflöst und gleichzeitig den Boot- Server bestimmt. Mit RARP konnten allerdings keine weiteren Informationen wie die Adresse eines Routers, Fileservers oder DNS- Servers übertragen werden.

Daraufhin wurde das Bootstrap- oder BOOTP-Protokoll [CROFT 1985] entwickelt, ein einfa-ches Klient- Server Protokoll. Der Klient sendet ein BOOTREQUEST per Broadcast an einen BOOTP Server, dieser antwortet mit einem BOOTREPLY Paket, das alle klientspezifischen Informationen enthält. Weiterhin wurde auch die Verwendung von BOOTP- Relays eingeführt, das Bootrequests aus dem lokalen Subnetz an entfernte BOOTP- Server weiterleitet. Die Informa-tionen wie IP- Adresse, Gateway usw. wurden aber weiterhin aus einer statischen Konfigurations-datei anhand der MAC- Adresse vergeben.

Ende der achtziger Jahre änderte sich die Situation dahingehend, dass die Vergabe und Ver-waltung von IP- Adresse und anderen Netzwerkkonfigurationseinstellungen dynamisch ohne Nut-zereingriff möglich sein sollte. Am Massachusetts Institute of Technology wurde zu diesem Zweck das Network Information Protokoll [MIT 1987] entwickelt. Es basiert auf RARP und beinhaltet ei-ne dynamische Adresswahl, Prüfung auf Vergabe der gewählten Adresse und die Übergabe diverser Konfigurationsparameter. Es ist in Teilen vergleichbar mit der zustandslosen Adresskonfiguration in IPv6. NIP kam allerdings nicht über den Status einer Beispielimplementation hinaus.

Mehr Erfolg hatte das DHC- Protokoll [DROMS 1993], es ist abwärtskompatibel zu BOOTP, bietet durch ein flexibles Optionsmodell dynamische Adressvergabe und vielfältige anwendungsspezifische Konfigurationserweiterungen.

4.4.2 Das DHCPv6 Protokoll

Bei dem DHC- Protokoll Version 6 handelt es sich nicht um eine Weiterentwicklung des bestehenden Systems, vielmehr es ist in fast allen Teilen neu entwickelt worden. Die wichtigsten Neuerungen umfassen:

- keine BOOTP- Kompatibilität mehr vorhanden,
- unterstützt Authentifizierung von Klient und Server,
- Klient kann durch Reconfigure Aufforderungen direkt beeinflusst werden,
- kann in eine AAA Infrastruktur eingebaut werden sowie
- Unterstützung von Multicast um alle Klienten gleichzeitig anzusprechen.

4.4.2.1 Nachrichtentypen

Es wird ein fester Nachrichtenkopf (s. Tab. 4.8) verwendet, mit dem jede Nachricht beginnt. Er enthält den Nachrichtentyp (msg-type), die Priorität des sendenden Servers (preference), einen Transaktionsidentifikator (transaction-ID), die Adresse des Klienten und die Adresse des Servers. DHCP unterscheidet zwischen folgenden Nachrichtentypen:

| | |
|---------------|--|
| Solicit | Klient möchte Server lokalisieren, |
| Advertise | Server antwortet auf Solicit mit seiner Adresse, |
| Request | Klient fragt nach Konfigurationsparametern, die im Optionsfeld definiert sind, |
| Confirm | Klient bestätigt, dass zugewiesene Adressen und Parameter noch Gültigkeit besitzen, |
| Renew | Klient fordert Erneuerung der vergebenen Adressen bzw. Parameter, |
| Rebind | vergleichbar mit Renew, aber an alle Server gerichtet, |
| Reply | Server antwortet auf Anfragen und übermittelt gleichzeitig angeforderte Adressen/Parameter, |
| Release | Klient informiert Server über Adressen, die wieder freigegeben werden können, |
| Decline | Klient informiert Server, dass übergebene Adresse(n) im lokalen Segment schon vorhanden ist, |
| Reconfig-Init | Server informiert Klient über neue Parameter, die per Request angefordert werden sollen, |
| Relay-Forw | ein Relay sendet eine Klient Nachricht verpackt in einer Option an den Server, |

Relay-Repl Server Antwort auf Relay- Forw, Antwort an Klient ist in Option enthalten, die das Relay an den Klienten weiterleitet.

Über diese zwölf Nachrichtentypen wird die Kernkommunikation zwischen Klient und Server abgewickelt. Der Wert des Prioritätsfeldes zeigt an, welcher Server vom Klient zu bevorzugen ist und die Transaktionsnummer, welche Nachrichten zu einer bestimmten Verbindung gehören. Alle weiteren Konfigurationsparameter werden durch verschiedene Optionsstrukturen übermittelt.

| | | |
|--------------------------|------------|----------------|
| 0.....31 | | |
| msg-type | preference | transaction-ID |
| client-link-local-adress | | |
| server-adress | | |
| options (variabel) | | |

Tabelle 4.8: DHCPv6 Nachrichtenaufbau

4.4.2.2 Der eindeutige DHCP Identifikator (DUID)

Jeder DHCP Klient besitzt einen global eindeutigen Identifikator. Er wird vom Server genutzt, um den Klienten zu identifizieren sowie vergebene Adressen und Konfigurationsparameter zu verwalten. Der Identifikator besteht aus einem der folgenden spezifischen Merkmale des Klienten:

- Hardware Adresse verknüpft mit der Zeit,
- eindeutiger herstellerspezifischer Identifikator,
- Hardware Adresse ,
- IMSI,
- IMEI.

Jeder Klient schickt bei jeder Anfrage seine DUID im Optionsfeld mit.

4.4.2.3 Klient- Adressen Beziehung (Identity association)

Wie schon erwähnt, kann ein Klient mehrere Netzwerkinterfaces besitzen, für die er jeweils mehrere Adressen vom Server beziehen kann. Damit Klient und Server die Adressen pro Interface effizient verwalten können, legt der Klient für jedes Interface eine IA mit einem eindeutigen Identifikator fest. Der Klient kann nun für jedes Interface über die IA und die IAID getrennt, Adressen vom Server fordern.

4.4.2.4 DHCP Optionen

Nachfolgend sind die wichtigsten DHCP Optionen für die DUID, IA und DNS-Server aufgelistet

| | |
|-------------|------------|
| 0.....31 | |
| OPTION DUID | option-len |
| DUID type | DUID len |
| DUID | |

Tabelle 4.9: DUID Option

| | | | | |
|------------------------|-------------|---------------|--------------------|---------------|
| 0.....31 | | | | |
| OPTION IA | | option len | | |
| IAID | | | | |
| T1 | | | | |
| T2 | | | | |
| IA status | num-addr | T | addr status | prefix length |
| IPv6 Address (16 Byte) | | | | |
| preferred lifetime | | | | |
| valid lifetime | | | | |
| T | addr status | prefix length | IPv6 - | |
| Address | | | preferred lifetime | |
| | | | | |

Tabelle 4.10: Identity Association Option

| | |
|-------------------------|-------------------------|
| 0.....31 | |
| OPTION_ORO | option-len |
| requested-option-code-1 | requested-option-code-2 |
| | |

Tabelle 4.11: Options Request (ORO) Option

| | |
|---------------------------|------------|
| 0.....31 | |
| OPTION_DNS_SERVERS | option-len |
| DNS Server 1(IP address) | |
| DNS Server 2 (IP address) | |
| | |

Tabelle 4.12: DNS Server Option

4.4.2.5 Ein typischer Protokollablauf

Es wird angenommen, dass der DHCP- Server am selben Subnetz angeschlossen und ein Router vorhanden ist, der site- local Präfixe an den Klienten propagiert. In diesem Fall würde folgendes ablaufen:

1. Der Klient bootet und startet die "Address Autoconfiguration".
2. Er erhält die site- local Präfixe durch ein Router Advertisement und erfährt durch gesetztes M Flag im RA, dass er DHCP nutzen soll.
3. Er schickt ein DHCP- Solicit Paket und als Option seine DUID.
4. Der Server prüft anhand der DUID, ob der Klient berechtigt ist und antwortet mit einem Advertise Paket, in dem seine Adresse enthalten ist.
5. Der Klient erhält ein Advertise und schickt ein Request Paket mit seiner DUID, einer IA- Option (die Adressfelder sind leer) und einer Domain- Search Option.
6. Der Server prüft seinerseits DUID und weist entsprechend der Voreinstellungen für diesen Klient bzw. globalen Regeln Adressen zu. Im Anschluss sendet er ein Reply Paket an den Klient mit ausgefüllter IA- Option.
7. Der Klient prüft ihm zugewiesene Adressen über den Duplicate Address Detection (DAD) Mechanismus (über das Senden von Neighbor Solicits) auf Eindeutigkeit im lokalen Netzwerksegment und schickt gegebenenfalls ein Decline Paket an den Server mit der abgelehnten Adresse. Danach werden die Adressen dem Interface zugewiesen.
8. Damit ist die Konfiguration abgeschlossen. Der Klient empfängt weiterhin DHCP- Nachrichten (z.B. Reconfigure- Init) und erneuert gegebenenfalls seine zugewiesenen Adressen, wenn deren Gültigkeit abgelaufen ist.

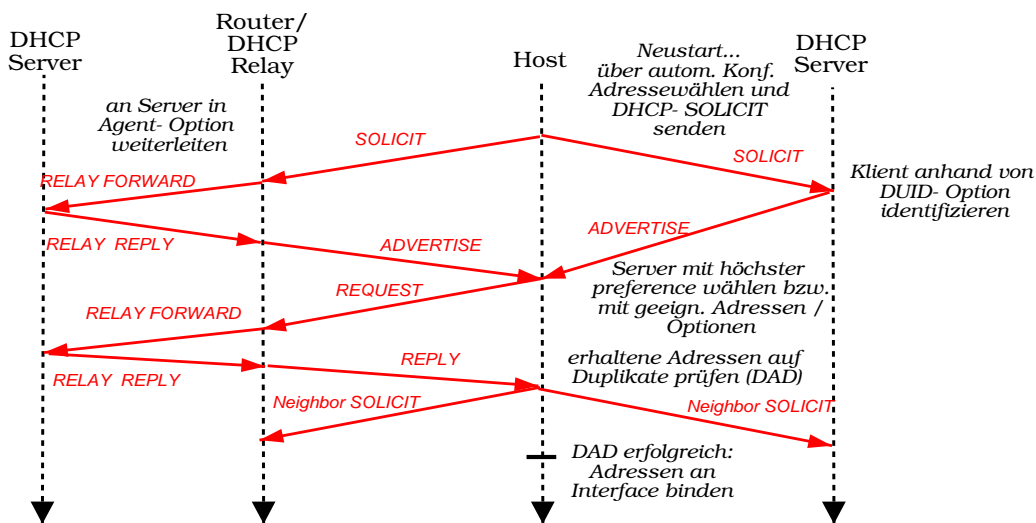


Abbildung 4.3: Zeitlinie DHCPv6 Konfigurationsablauf

4.4.2.6 Protokollablauf über ein DHCP- Relay

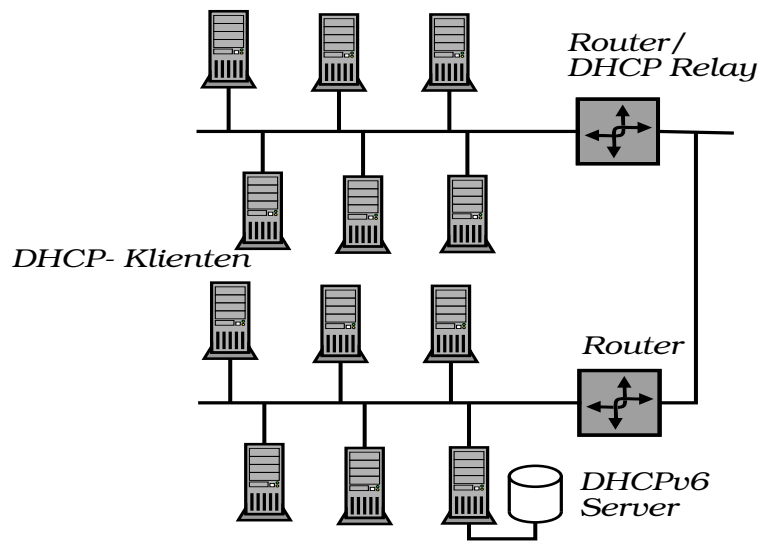


Abbildung 4.4: DHCP über einen Router als Relay

Sollte sich der DHCP- Server nicht innerhalb des selben Subnetz befinden, dies ist in grösseren Netzen Standard, wird ein DHCP- Relay Agent benötigt. Diese Funktion kann ein Netzwerkgerät der Schicht 3/4, z.B. der lokale Router, übernehmen.

Protokollablauf:

1. Ein Klient schickt ein DHCP- Solicit.
2. Relay empfängt das Solicit und schickt eine Relay-forward Nachricht, an die das Solicit des Klienten als Option (OPTION_CLIENT_MSG) angehängt wird.
3. Der DHCP- Server extrahiert das Klienten Solicit, formt eine Advertisement Nachricht und schickt diese als Option (OPTION_SERVER_MSG) in einer Relay- Server Nachricht wieder an das Relay zurück.
4. Das Relay extrahiert die als Option empfangene Advertisement Nachricht und schickt sie an den Klienten weiter.
5. Der weitere Protokollablauf läuft über den Mechanismus des Relay-forward und Relay- Server Nachrichtenaustausches identisch, wie in 4.4.2.5 beschrieben, ab.

| | |
|-------------------------------------|---------------|
| 0..... | 31 |
| msg-type (RELAY-FORWARD) | prefix length |
| relay-address | |
| options (variable Anzahl und Länge) | |

Tabelle 4.13: Relay Forward Nachricht

| |
|--------------------------------|
| 0.....31 |
| OPTION_CLIENT_MSG option-len |
| DHCP client message |
| |

Tabelle 4.14: Klient Nachricht Option

4.4.2.7 Verzicht auf Relay durch Adress Autoconfiguration

Bei der Betrachtung des Zusammenhangs zwischen zustandsloser und zustandsgebundener Adresskonfiguration stellt sich die Frage, weshalb die Einführung eines DHCP Relays notwendig ist. Über eine spezielle site- local Adresse, die über RAs dem Klienten mitgeteilt wird und mit der ausschliesslich der DHCP Server erreicht werden kann, wäre es möglich, den DHCP Server anzusprechen. Folglich könnte auf DHCP Relays verzichtet werden.

4.4.2.8 Klientenzustände

Die Übergänge zwischen den verschiedenen Zuständen des Klienten lassen sich am besten anhand der Abbildung 4.5 erläutern. Nach dem Neustart des Klienten werden SOLICIT Nachrichten an die DHCPv6 Server Multicastadresse gesendet. Empfangene ADVERTISEMENT Nachrichten werden gesammelt. Nach Ablauf der Zeit ADV_MSG_TIMEOUT geht der Klient in den Auswahlzustand (selecting). Dabei wird anhand der erhaltenen "prefer" Werte und den angebotenen Adressen ein DHCP- Server ausgewählt. Nach der Auswahl befindet sich der Klient im Anfragezustand (requesting), eine REQUEST Nachricht wird an den Server gesandt. Der Server antwortet mit einer REPLY- Nachricht, die die gewünschten Adressen und Konfigurationsparameter enthält. Diese Adressen werden dem jeweiligen Interface zugewiesen und die Konfigurationsparameter gesetzt. Der Klient befindet sich jetzt im Zustand der Adressbindung (bound), die erhaltenen IAs werden gespeichert. Nach Ablauf der Zeit T1 oder der "preferred lifetime" einer Adresse wird im Zustand Erneuern (renewing) eine RENEW- Nachricht an den Server gesendet. Der Server antwortet und verlängert die Zeiten oder sendet neue Adressen. Danach befindet sich der Klient wieder im Zustand Adressbindung. Wurde auf die RENEW- Anfragen keine Antwort gegeben und ist die Zeit T2 abgelaufen, wird schliesslich ein neuer Server im Zustand "Rebind" gewählt.

Nach einem Netzwerksystemneustart oder einem Subnetzwechsel wird nach vorhandenen noch gültigen IAs gesucht. Falls mindestens eine Adresse gefunden wurde, wird an die DHCPv6 Server Multicastadresse eine Bestätigungsaufforderung für die Nutzung (CONFIRM) dieser IA gesendet. Sollte die Antwort des Servers positiv sein, erfolgt der Übergang in den Zustand Adressbindung.

Der Server kann durch das Versenden einer RECONFIGURE- INIT Nachricht den Klienten aus dem Adressbindungs- Zustand auffordern, eine neue Anfrage zu starten.

4.4.3 DHCP und ein dynamisches DNS

Das Domain Name System ist historisch gesehen eine statische Zuordnung zwischen einem Fully Qualified Domain Name (FQDN) und einer IP Adresse. Mit der Verwendung von DHCP werden den Klienten wechselnde IP- Adressen aus einem Adresspool zugewiesen. Eine statische FQDN Zuweisung für alle Adressen aus einem Pool, z.B. pool-ip-200.domain.de, löst das Problem der

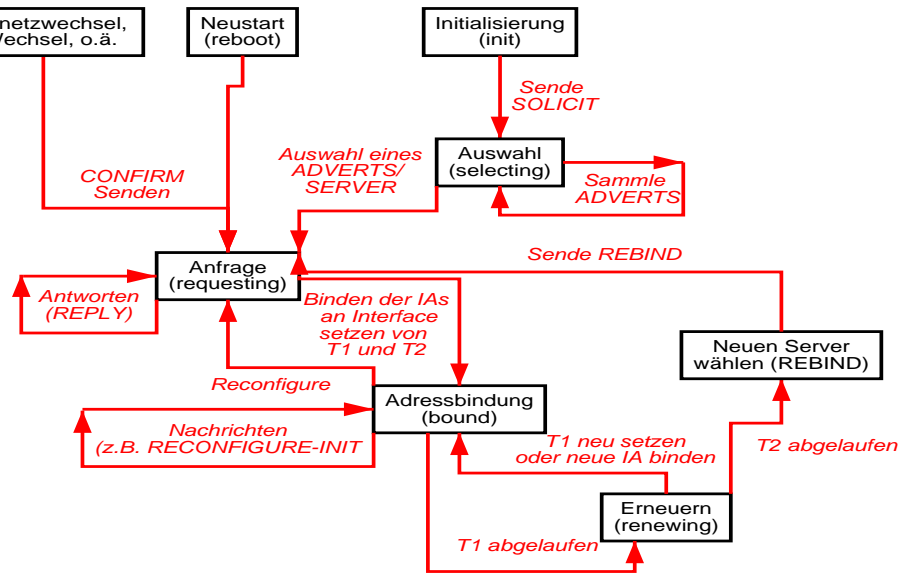


Abbildung 4.5: Klienten Zustandsdiagramm

Adresse zu FQDN Auflösung. Aber es ist nicht möglich über einen Namen einen bestimmten Klienten zu erreichen. Diese Funktionalität wird allerdings in folgenden Fällen benötigt:

- Adress- Renumbering
- DHCP- Klienten agieren als Dienstanbieter (WWW, FTP, IP- Telephonie)

Um die Funktionalität zu erreichen ist es notwendig, die Datenbank des lokalen Domainname-system (DNS) über die zur momentan benutzen IP- Adresse veränderte Zuordnungen des Fully Qualified Domain Names zu informieren und den Namensraum somit konsistent zu halten. Zur Unterstützung wurde das an sich statische DNS um eine dynamische Komponente erweitert (s. [VIXIE 1997]). Die erste Standardvariante des dynamischem DNS wird allerdings nicht genutzt, da keine sichere Authentifizierung spezifiziert ist. Erst mit der Erweiterung um DNS Security besteht die Möglichkeit der praktischen Anwendung des dynamischen DNS.

Der DHCP Server ist nunmehr in der Lage, DNS über veränderte IPv6 Adresszuweisungen zu informieren. Es ist prinzipiell auch möglich, dass der Klient das DNS Update ausführt. Diese Möglichkeit findet aber in der Praxis keine Anwendung, denn dazu würde jeder Klient für das Update ein gemeinsames Geheimnis mit dem DNS- Server benötigen.

4.4.3.1 Unterstützung von sicheren DNS Updates

Die Unterstützung von sicheren DNS Updates macht das Bestehen einer Vertrauensbeziehung zwischen DHCP- und DNS Server erforderlich. Diese Vertrauensbeziehung wird durch ein gemeinsames Geheimnis hergestellt, mit dessen Hilfe Update- Aufforderungen signiert werden. Der DNS Server ist somit in Lage den Update Datensatz auf Veränderungen und gleichzeitig die Berechtigung des Updates prüfen.

4.5 Nutzen eines zusätzlichen zustandsgebundenen Adresskonfigurationsprotokolls

Mit der Autoadresskonfiguration und DHCPv6 stehen zwei Mechanismen zur Verfügung, um IPv6 Adressen dynamisch zu vergeben. Dieser Abschnitt soll klären, ob der Einsatz von DHCPv6 Vorteile gegenüber einem alleinigen Einsatz der Autoadresskonfiguration bietet.

Ein Host der nur Autoadresskonfiguration nutzt, ist in der Lage:

- Adressen zu wählen und Adresspräfixe anhand von RAs zu lernen,
- Renumbering anhand von Timeouts der Präfixe zu unterstützen ,
- Authentifizierung gegenüber dem Netzwerk über die RA Authentifizierungs Option zu starten,
- anhand von vordefinierten Anycast- Adressen Dienste wie DNS oder Fileserver zu erreichen,
- die in [NARTEN 2001] gezeigte Privatsphärenerweiterung zu nutzen.

Es ist für ihn aber nicht möglich:

- dynamisch DNS Updates durchzuführen, da ein gemeinsames Geheimnis mit dem DNS Server nötig wäre (TSIG) oder
- ohne ein zusätzliches Protokoll, wie z.B. das Service Location Protokoll (SLP), weitere Dienste zu erreichen, die nicht durch eine Anycast- Adresse vordefiniert sind.

Ein zusätzlicher Vorteil von DHCPv6 ist die Kontroll- und Steuerungsmöglichkeit an einer zentralen Stelle.

So kann z.B. festgelegt werden :

- welchem Klient welche Adresse zugewiesen wird,
- welches die Konfigurationsparameter eines speziellen Klienten sind
- ob bestimmte Klienten autorisiert sind, Adressen zu erhalten.

Die meisten der obigen Eigenschaften liessen sich, neue Protokolle oder Erweiterungen von RA und ICMPv6 vorausgesetzt auch mit der zustandslosen Adresskonfiguration realisieren. Unter der Vielzahl von Erweiterungen (z.B. RA Authentifizierung) würde jedoch die Übersichtlichkeit im Netzwerk leiden und die Fehlersuche würde erschwert.

Die Frage eines Einsatzes von DHCPv6 oder der alleinigen Nutzung von automatischer Adresskonfiguration hängt von den jeweiligen Anforderungen der Netzwerkinfrastruktur ab. So sind Einsatzmöglichkeiten für die automatische Adresskonfiguration in einfachen kleinen bis einfachen mittleren Netzwerke zu sehen, deren Betrieb und Aufbau nur geringe administrative Eingriffe erfordern soll. Der Einsatz von DHCPv6 ist dann sinnvoll, wenn:

- grosse, komplexe Netzinfrastrukturen vorhanden sind bzw. aufgebaut werden sollen,
- spezielle Konfigurationen durchgeführt werden (wie die Anbindung an mehrere ISP),

- die Vorteile des Netzwerkmanagements den Nachteil des zusätzlichen administrativen Aufwands rechtfertigen,
- administratives Wissen vorhanden ist und Netzwerkmanagement betrieben werden soll und
- DHCP die Grundlage für ein Sicherheitskonzept wie die Klientenauthentifizierung bildet.

Für einen parallelen Betrieb beider Mechanismen, der vereinzelt vorgeschlagen wird, z.B. die automatische Adresskonfiguration mit Konfigurationsparametern die über DHCPv6 bezogen werden zu kombinieren, sehe ich keine sinnvollen Einsatzszenarien. Wird solch ein erhöhter Aufwand für DHCPv6 betrieben, sollte dann auch die Adressvergabe über DHCPv6 abgewickelt werden. Das hätte den Vorteil, dass die jeweiligen Daten zentral verwaltet werden könnten.

Zusammenfassend lässt sich sagen, dass beide Mechanismen ihre Berechtigung haben und notwendig sind. Die automatische Adresskonfiguration ist im Vergleich zu IPv4 der einfache Router in einem Büro, der angeschlossene Klienten mit IP- Adressen über DHCP versorgt. Der Einsatz von DHCPv6 ist dagegen eher in grossen Netzwerkinfrastrukturen, wie der des URZ, sinnvoll.

4.6 Sicherheitsüberlegungen

Mit der Einführung von DHCPv6 ergeben sich natürlich auch neue Angriffspunkte. Folgende Angriffe sind denkbar:

- das Senden von falschen Rekonfigurationsaufforderungen an die Klienten (reconfigure init),
- Betreiben eines DHCP- Servers, der falsche Informationen an die Klienten liefert,
- dem DHCP Server durch gefälschte Decline Nachrichten Adressen entziehen und somit den Dienst stören.

Dabei müssen solche Störungen nicht unbedingt von einem Angreifer ausgehen, sondern können auch durch Konfigurationsfehler in Komponenten (z.B. Router als DHCP Server konfiguriert) verursacht werden, die zum Netzwerk hinzugefügt wurden.

Die Art der Angriffe lässt sich nach den Zielen des Angreifers wie folgt klassifizieren:

1. Störung des Dienstes (z.B. DoS),
2. Übermittlung von falschen Informationen an den Klienten z.B. für einen “man in the middle” Angriff,
3. Nutzung geschützter Dienste und Ressourcen.

Angriffe der Art zwei und drei lassen sich durch den Einsatz von Authentifikation und Autorisation unterbinden. DoS Angriffe dagegen lassen sich dadurch leider nicht vollständig unterbinden, werden aber erschwert. Die in 4.6.1 vorgestellte Authentifizierungsoption stellt einen Lösungsansatz dar, deckt aber noch nicht alle Szenarien befriedigend ab. Dies versucht die vorgeschlagene Erweiterung mit einem AAA- Server bzw. Netzwerk.

Grundsätzlich ist ein Einsatz von DHCPv6 ohne Sicherung bzw. Nutzung der Authentifizierungsoption nicht zu empfehlen.

4.6.1 DHCP Authentifizierung

Diese Option beschreibt nicht die Authentifizierung an sich, sondern stellt nur die Struktur für verschiedene nutzbare Authentifizierungsmethoden dar. Bisher sind das "Configuration token" und das "Delayed authentication" Protokoll definiert. Ersteres beschreibt den nicht verschlüsselten Austausch eines gemeinsamen Geheimnisses z.B. ein Passwort, zusätzlich ist auch die Unterstützung von Kerberos Tickets durch das Protokoll geplant. Mit dem Delayed Authentication Protokoll ist es hingegen möglich:

- DHCP- Klient und Server wechselseitig zu authentifizieren,
- DHCP Nachrichten auf Unversehrtheit zu prüfen,
- Angriffe, die auf dem wiederholten Senden einer abgehörten Nachricht basieren, zu unterbinden (Replay detection).

Voraussetzung ist die Existenz eines gemeinsamen Geheimnisses zwischen jedem Klient und dem DHCP Server. Mit diesem Geheimnis und unter Zuhilfenahme des im Feld "Algorithm" bezeichneten Algorithmus (z.B. HMAC-MD5, HMAC-SHA) wird dann die gesamte DHCP Nachricht einschliesslich aller Optionen signiert. Die Signatur wird im Feld "Authentication Information" übertragen. Der DHCP Server kann nun anhand der DUID und der Secret ID (in Authentication Inf.) ermitteln, welches Geheimnis benutzt wurde. Somit ist er in der Lage, ebenfalls eine Signatur zu erstellen und mit der bereits übermittelten zu vergleichen. Sollte die ermittelte Signatur nicht mit der empfangenen übereinstimmen, wird die Nachricht verworfen und ein entsprechender Logdatei- Eintrag vorgenommen. Den Wunsch nach Authentifizierung teilt der Klient am Anfang der Kommunikation durch die Solicit Nachricht mit einer Authentifizierungsoption, die den zu verwendenden Algorithmus beschreibt, dem Server mit.

| | | | |
|----------------------------|-----------|---------------|----------------|
| 0.....31 | | | |
| OPTION_AUTH | | option-length | |
| Protocol | Algorithm | RDM | Replay detect. |
| Replay Detection (64 bit) | | | |
| Replay cont. | | Auth. Info | |
| Authenitcation Information | | | |

Tabelle 4.15: Aufbau der Authentifizierungsoption

Nachteile der "Delayed Authentication" sind:

- ungeklärter Geheimnisaustausch und
- die fehlende Unterstützung von Roaming zwischen verschiedenen administrativen Domänen.

4.6.2 DHCP Authentifizierung mit AAA- Server

Eine weitreichendere Lösung erfordert einen AAA- Server. Mit diesem unterhält der DHCP-Server eine Vertrauensbeziehung. Die Authentifizierung würde nicht mehr durch den DHCP Server erfolgen, sondern Anfragen nach Authentifizierung und Autorisation würden an den AAA- Server

weitergeleitet. Der AAA- Server nimmt dann die Authentifizierung selbst vor oder leitet bei Klienten anderer administrativer Domänen die Anfrage an den zuständigen AAA- Server weiter. Der DHCP Server würde in diesem Fall z.B. als Diameter Klient agieren. Mithin wäre es möglich, Klienten nur noch mit einem Geheimnis bzw. Schlüssel auszustatten, der für eine Reihe von AAA-Aufgaben verwendet werden kann.

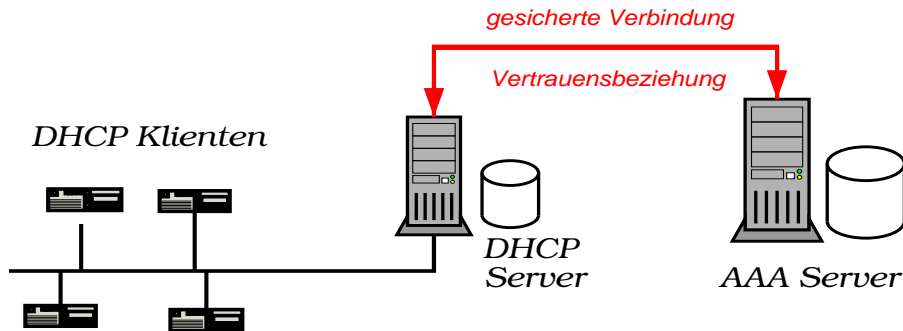


Abbildung 4.6: DHCP Server / AAA Server Beziehung

4.6.3 Unterstützung von IPv4 Geräten durch den DSTM Mechanismus

Der Dual Stack Transitions Mechanismus (DSTM) ist ein Protokollvorschlag, der unter anderem DHCPv6 nutzt, um die Interoperabilität zwischen bestehenden IPv4 Netzwerken und neu aufgebauten IPv6 Teilnetzwerken zu gewährleisten. Der Einsatz von DHCPv6 macht ihn für diese Arbeit interessant, weshalb ich kurz darauf eingehen möchte. DSTM verbindet zwei Mechanismen, die Vergabe von IPv4 Adressen an IPv6 Hosts (AIIH) und den dynamischen Aufbau von IPv4 in IPv6 Tunneln (DTI). Mit DSTM kann demnach ein IPv6 Host mit IPv4 Netzwerkschicht Verbindungen zu IPv4 Host herstellen und umgekehrt.

Dies geschieht im Fall IPv6 zu IPv4 folgendermassen:

1. Der IPv6 Host versucht den Namen des IPv4 Host über das DNS aufzulösen, erhält aber keinen A6 bzw. AAAA Eintrag mit IPv6 Adresse als Antwort, sondern nur eine IPv4 Adresse.
2. Der IPv6 Host schickt eine Anfrage (REQUEST) nach der Option DSTM an den DHCP Server und erhält eine IA mit einer IPv4 Adresse als Antwort.
3. Jetzt kann der IPv6 Host mit der IPv4 Adresse über das Tunnel Interface einen IPv4 in IPv6 Tunnel zum DSTM Grenzrouter aufbauen.
4. Der DSTM Grenzrouter als Tunnelende leitet die getunnelten IPv4 Pakete an den IPv4 Host weiter.
5. Die Antworten des IPv4 Host werden wieder vom DSTM Grenzrouter in das IPv6 Netzwerk getunnelt.

Im umgekehrten Fall fragt der IPv4 Host den DNS nach einem A Eintrag. Falls keiner vorhanden ist, weist der DNS Server den DHCP- Server an, dem IPv6 Host eine Adresse zuzuweisen. Der

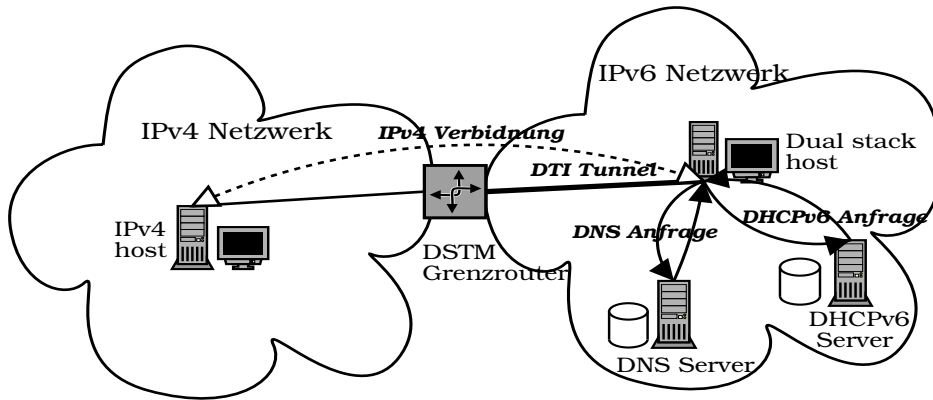


Abbildung 4.7: DSTM Mechanismus (nach [EURESCOM 2000])

DHCP- Server sendet ein Reconfigure - Init und weist die Adresse zu. Danach kann der IPv6 Host mit einer IPv4 Adresse erreicht werden.

Vorteile dieser Lösung sind :

- Es besteht die Möglichkeit IPv6 Netzwerke auch ohne IPv4 Adressvergabe und entsprechende Routingtabellen aufzubauen.
- Solche "IPv6 Inseln" sind in der Lage, mit dem bestehendem IPv4 Netzwerk zu kommunizieren.
- IPv4 Adressen müssen nicht statisch vergeben, sondern können ausschliesslich im Bedarfsfalle von den IPv6 Hosts per DHCPv6 angefordert werden. Somit können mit einer begrenzten Anzahl IPv4 Adressen grosse IPv6 Netzwerke versorgt werden.

| | |
|--------------------------------------|------------|
| 0.....31 | |
| OPTION_DSTM | option-len |
| Tunnel End Point (TEP) (optional) | |
| IA Option mit Adressen | |

Tabelle 4.16: DSTM Global IPv4 Adress Option

Der bedeutende Nachteil des DSTM Mechanismus liegt in der Komplexität des Protokolls. Seine Arbeitsweise hängt von zu vielen Komponenten ab, die aufeinander abgestimmt werden müssen. Es erfordert die Schaffung von Schnittstellen zwischen DNS und DHCP- Server, und zusätzlich werden spezielle Router für den Einsatz als Grenzrouter benötigt. Dieser recht hohe Aufwand steht in keiner Relation zum Nutzen, zumal das vorgestellte Einsatzszenario nicht allzu häufig anzutreffen sein wird. Ein weiterer Nachteil ist, dass Hosts, die lediglich mit einer IPv6 Netzwerkschicht ausgestattet sind, nicht erreicht werden können. Deshalb bezweifle ich, dass sich DSTM durchsetzen wird.

4.7 Einordnung von DHCPv6 in die AAA- Infrastruktur

Wie in den vorangegangenen Abschnitten schon erwähnt, liessen sich DHCP- Server und DHCP-Relay in bestimmten Einsatzbereichen für AAA- Dienste einsetzen. Es bieten sich dabei zum Beispiel die Zugangssteuerung für:

- Funk LAN Infrastrukturen,
- Breitbandkabelmodems,
- und mobile Endgeräte an.

Ein Einsatz also für diejenigen Dienste, die auf die Übermittlung von Konfigurationsparametern wie IP- Adressen angewiesen sind und gleichzeitig eine Authentifizierung und Autorisierung für den Netzwerkzugang erfordern. Bei den eben genannten Diensten kann ein eindeutiger Nutzer bzw. Vertragspartner nur jeweils einem Endgerät, für das er auch die Verantwortung trägt, zugeordnet werden. Diese Bindung kann durch eine Authentifizierung des Nutzers gegenüber dem Gerät sichergestellt werden. Zur Authentifizierung würden DHCP Server und Relay eingesetzt. Sie arbeiteten als AAA- Agenten arbeiten und leiteten die erforderlichen Daten an die AAA-Infrastruktur weiter.

4.8 Fazit

Mit DHCPv6 wird ein multifunktionales, modulares Protokoll definiert, welches durch sein leicht erweiterbares Optionskonzept an die verschiedensten Aufgaben anpassbar ist.

So könnte man mit DHCPv6:

- die Authentifizierung von Nutzern und Geräten gegenüber der Netzwerkinfrastruktur vornehmen,
- zentral Schlüssel für Anwendungen in höheren Schichten verteilen und
- mobile Dienste wie Roaming in verschiedenen Netzinfrastrukturen unterstützen

Im aktuellen Protokollvorschlag wird auf diese Möglichkeiten eher am Rande eingegangen, so existiert z.B. bislang noch kein Konzept für die Anbindung an Diameter AAA- Server. Der Nutzen von DHCPv6 ergibt sich aber weniger durch die Konfigurationsverteilung und die Adressvergabe sondern durch Zusatzdienste wie die Authentifizierung und die zentrale Abwicklung von dynamischen DNS Updates. Sollten diese Zusatzdienste nicht bereits jetzt, in die Protokolldefinition mit aufgenommen werden, wird sich das DHCPv6 Protokoll nicht zum Standardkonfigurationsprotokoll für IPv6 entwickeln. In diesem Fall vermute ich, werden die Adressautokonfigurationsmechanismen um Dienste wie Authentifizierung erweitert und mit Unterstützung spezialisierter Protokolle wie dem Dienstlokalisierungsprotokoll (SLP) diese Rolle übernehmen.

Kapitel 5

DHCPv6 Implementierung

5.1 Server

Die Aufgabenstellung lautete: Erstellung einer DHCPv6 Serverimplementation, die den bestehenden DHCPv4 Server des Internet Software Consortiums (ISC) derart erweitert, dass beide Protokolle vom Server unterstützt werden.

Das ISC ist eine seit 1995 bestehende gemeinnützige Organisation, die sich als Ziel die Schaffung hochwertiger Referenzimplementierungen von grundlegenden Internet Standards für die gebräuchlichsten Plattformen und Systeme gesetzt hat. Mit der Implementierung werden erfahrene Programmierer bzw. Programmiererteams beauftragt. Diese Arbeit wird durch Sponsorengelder, Supportdienstleistungen und Einnahmen aus Buchverkäufen finanziert. Zu den bedeutendsten Implementierungen zählen das Domain Nameserversystem ISC BIND, DHCP und der Internet News Server INN.

Die ISC DHCP Implementation in der Version 3 ist ein seit November 1996 stetig weiterentwickeltes und verbessertes Softwarepaket, welches folgende Eigenschaften aufweist:

- es beinhaltet einen DHCP- Server, einen Klient und ein Relay sowie
- eine Managementschnittstelle (Open Management Application Interface (OMAPI)), mit der das laufende Serversystem beeinflusst und gesteuert werden kann,
- Unterstützung von einem Backupserver der über ein Failover Protokoll über Zustandsänderungen informiert wird,
- Unterstützung von dynamischen DNS Updates.

Zielstellung war die Erweiterung des bestehenden ISC DHCPv4 Servers der Version 3.0 dahingehend, dass ein Parallelbetrieb von DHCPv4 und DHCPv6 möglich ist. Der Parallelbetrieb bietet den Vorteil, dass in der Übergangsphase die Möglichkeit der Konfiguration vereinzelter IPv6 Hosts besteht, ohne dass der Einsatz eines anderen DHCP- Servers erforderlich wird. Zusätzliche Vorteile einer Erweiterung des bestehenden Servers sind die Weiterverwendung von bestehenden Funktionen und Strukturen, z.B. für das Einlesen und Auswerten der Konfigdatei und der Lease-datenbank. Allerdings ist dafür Voraussetzung, sich in die Funktionsweise und die Strukturen des

zu erweiternden Programms exakt einzulesen. Desweiteren ergibt sich aus einer erfolgreichen Erweiterung auch der Zwang, sich an durch die Programmstrukturen vorgegebenen Vorgehensweisen anzupassen.

Das Verstehen der Programmstruktur in allen Einzelheiten hat mich etwas Zeit gekostet, deshalb konnte der geplante Funktionsumfang noch nicht vollständig realisiert werden. Eine eigenständige Implementation hätte zum jetzigen Zeitpunkt einen höheren Funktionsumfang bieten können.

Als Grundlage der Implementation dient der vorläufige Standard nach [DROMS 2001] in der Revision 19. Nachträgliche Veränderungen, die über die DHCP- Mailingliste bekannt gegeben wurden, sind in der Arbeit berücksichtigt.

5.1.1 ISC DHCP 3.0 Serveraufbau

Der DHCP Server ist, wie alle ISC Entwicklungen, aus Portabilitätsgründen ein Standard C Programm. Die internen Strukturen des ISC DHCP Servers bieten einige interessante Details, wie z.B.

- eigene Speicherverwaltung mit Referenzzählern,
- Lease-, Klienten- und Konfigurationsdatenbanken sind als Hashtabelle organisiert, wobei jeweils mehrere Suchschlüssel existieren,
- viele Strukturen sind als eine Art Pseudoobjekte angelegt, d.h. die Datenstrukturen enthalten für bestimmte Ereignisse Sprungziele (callbacks) zu Funktionen,
- OMAPI Schnittstelle, um interne Datenstrukturen zu verändern,
- der grösste Teil des Programmcodes wird von Server, Klient und Relay genutzt,
- Unterstützung des Berkley Packet Filter (BPF), um direkt von der Netzwerkschicht ankommende Pakete zu erhalten.

Da DHCP- Anfragen in der Regel innerhalb sehr kurzer Zeit beantwortet werden, kann die Verarbeitung in einem einzigen Prozess abgewickelt werden. Die Verwaltung mehrerer Verbindungen wird unter Zuhilfenahme des `select()` Systemrufs realisiert.

5.1.1.1 Programmaufbau im Detail

Nach dem Programmstart wird aus dem Konfigurationsfile (`/etc/dhcpd.conf`) und aus den existierenden Leases (`var/state/dhcpd.leases`) die Konfigurationsdatenbank und die Leasedatenbank aufgebaut. Im Anschluss daran werden die verfügbaren Interfaces konfiguriert. Für jedes vorhandene Interface wird ein raw Socket (BPF) erzeugt und gebunden. Das Interface erhält eine Struktur "interface_info", in der die Socketfiledescriptoren (FD) und zu rufende Funktionen für bestimmte Ereignisse wie z.B. `receive_packet()` oder `do_packet()` enthalten sind. Die FD der Interfaces werden für den `select()` Ruf in einem Feld registriert. Danach wird mit `dispatch()` in die Serverschleife gesprungen. Von diesem Punkt aus erfolgt die Verteilung der Netzwerkanfragen und die Verarbeitung von Timeouts.

Beim Eintreffen einer Anfrage wird anhand des FD das entsprechende Interface ermittelt und die zugehörige Funktion gerufen (`receive_packet`). Desweiteren wird das empfangene Packet an die Funktion `do_packet()` übergeben, diese wiederum ruft `dhcp()`. Mit der `dhcp` Funktion wird das Packet analysiert und klassifiziert (ist der Klient bekannt, ist es das richtige Interface für diese Anfrage usw.). Danach wird gemäss des DHCP Typs die entsprechende Verarbeitungsfunktion gerufen. Diese bearbeitet die enthaltene Anfrage (vergibt Lease, erneuert ihn usw.) und kehrt schliesslich wieder zur `dispatch` Funktion zurück.

5.1.1.2 Der Berkley Packet Filter

Der ISC DHCP Server nutzt den Berkley Packet Filter bzw. sein Linux Pendant den Linux Packet Filter, um das komplette Paket direkt und inklusive des Ethernetkopfes von der Netzwerkschicht zu beziehen. So können weitere Informationen wie die MAC- Absenderadresse aus dem Paket genutzt werden.

BPF ist ein Mechanismus der überwiegend von Netzwerkanalysewerkzeugen wie `tcpdump` oder `snort` benutzt wird, um den kompletten Netzwerkdatenverkehr anzuzeigen. Dabei wird einem RAW- Socket eine Paketfilterregel angehängt, dieser Regel wird auf jedes ankommende Paket angewendet. Passt das Paket zur Regel, wird es über den Socket an das entsprechende Programm weitergeleitet. Die Filterregeln werden in Form eines Assemblercode- ähnlichen Programmstückes verfasst, bzw. ist es mit `tcpdump` möglich, aus logischen Beschreibungen BPF- Filter zu erzeugen. So kommt in DHCP ein Filter zum Einsatz, welcher lediglich die UDP Protokollnummer und die gewünschte Portadresse enthält. Auf diese Weise wird gewährleistet, dass alle Pakete an den DHCP UDP Port und demzufolge an die Anwendung weitergeleitet werden, ohne sich an einen bestimmten Socket binden zu müssen.

Nachfolgend wird das Beispiel eines BPF Filters aufgeführt, welches ausschliesslich Pakete mit Absenderadressen von 128.3.112.15 bis 128.3.112.35 empfängt.

```
struct bpf_insn insns[] = { BPF_STMT(BPF_LD+BPF_H+BPF_ABS, 12),
                           BPF_JUMP(BPF_JMP+BPF_JEQ+BPF_K, ETHERTYPE_IP, 0, 8),
                           BPF_STMT(BPF_LD+BPF_W+BPF_ABS, 26),
                           BPF_JUMP(BPF_JMP+BPF_JEQ+BPF_K, 0x8003700f, 0, 2),
                           BPF_STMT(BPF_LD+BPF_W+BPF_ABS, 30),
                           BPF_JUMP(BPF_JMP+BPF_JEQ+BPF_K, 0x80037023, 3, 4),
                           BPF_JUMP(BPF_JMP+BPF_JEQ+BPF_K, 0x80037023, 0, 3),
                           BPF_STMT(BPF_LD+BPF_W+BPF_ABS, 30),
                           BPF_JUMP(BPF_JMP+BPF_JEQ+BPF_K, 0x8003700f, 0, 1),
                           BPF_STMT(BPF_RET+BPF_K, (u_int)-1),
                           BPF_STMT(BPF_RET+BPF_K, 0), };
```

Auch für den DHCPv6 Server bzw. mit IPv6 wäre ein Einsatz von BPF denkbar gewesen. Mangels sich daraus ergebender, im Verhältnis zu DHCPv4 signifikanter Vorteile wurde jedoch auf dessen Einsatz in der vorliegenden Lösung verzichtet. Stattdessen finden hier auch bisher gebräuchliche Netzwerksockets Verwendung.

5.1.2 Die DHCPv6 Erweiterung

Die vorliegende Version meiner DHCPv6 Implementation ist ein Prototyp, der noch nicht sämtliche Eigenschaften und Optionen des derzeitigen Standards unterstützt, aber entsprechende Funktionen für ein einfaches Erweitern bereitstellt und die Schnittstellen zum ISC DHCPv4 Server aufzeigt.

Der Funktionsumfang umfasst:

- Vergabe von festen IPv6 Adressen an bekannte Hosts,
- SOLICIT Verarbeitung,
- REQUEST Beantwortung,
- RENEW der vergebenen Gültigkeitszeiten,
- Verarbeitung von DECLINE Nachrichten des Klienten für bestimmte Adressen und
- Unterstützung von Optionstyp DUID und IA.

5.1.2.1 Programmaufbau

Die Netzwerkinterfaces unterstützen beide Netzwerkprotokolle, sowohl IPv4 als auch IPv6 (Dual Stack). Da aber das Interface eth0 bereits als IPv4 Interface erkannt wird und aufgrund dessen die entsprechende `interface_info` Struktur erzeugt wird, war es notwendig, einen Trick anzuwenden, um die generelle IPv6 Netzwerkunterstützung hinzuzufügen. Die Funktion `discover()` erzeugt dabei für die Erkennung der vorhandenen Interfaces einen Netzwerksocket und wendet auf diesem den `ioctl()` Ruf `SIOCGIFCONF` an. Der Socket liefert eine Liste mit vorhandenen Interfaces und deren Einstellungen zurück. Um IPv6 Interfaces in den Standardablauf einzubinden, wird jeweils ein Pseudointerface, gekennzeichnet durch `*_6`, angelegt und in die Liste der verfügbaren Interfaces aufgenommen. Vor der Interfaceregistrierung wird dann nach der Registrierung der Pseudointerfaces und der normalen Unterschiede die Registrierung mit unterschiedlichen Funktionszeigern gestartet. Danach erfolgt das Binden an einen Netzwerksocket und das Abonnieren der `ALL_SERVER` Multicastgruppe.

Anschließend erfolgt der Sprung in die `dispatch()` Funktion und bis zum Empfang eines Paketes der Sprung in den `select()` Systemruf. Dann werden die eigenen Funktionen für das Einlesen und Verarbeiten des Pakets gerufen. `Dhcp_6()` ist für die Klassifizierung des Klienten und das Rufen der dem DHCPv6 Pakettyp entsprechenden Funktion zuständig.

5.2 Der DHCPv6 Klient

Der DHCPv6 Klient basiert auch auf der ISC DHCP Implementation. Der Plan, auch hier einen Parallelbetrieb von DHCPv4 und DHCPv6 zu ermöglichen, wurde nicht umgesetzt. Die Abläufe und Strukturen des DHCPv6 Klienten sind zu DHCPv4 zu unterschiedlich, sodass der Aufwand einer Anpassung und Erweiterung nicht vernünftig erschien.

Klient und Server nutzen die wichtigsten Funktionen wie `receive_packet_6()`, `lookup_option_6()` gemeinsam. Auch im Aufbau ähneln sie sich. So besitzt der Klient eigene Funktionsumsetzungen

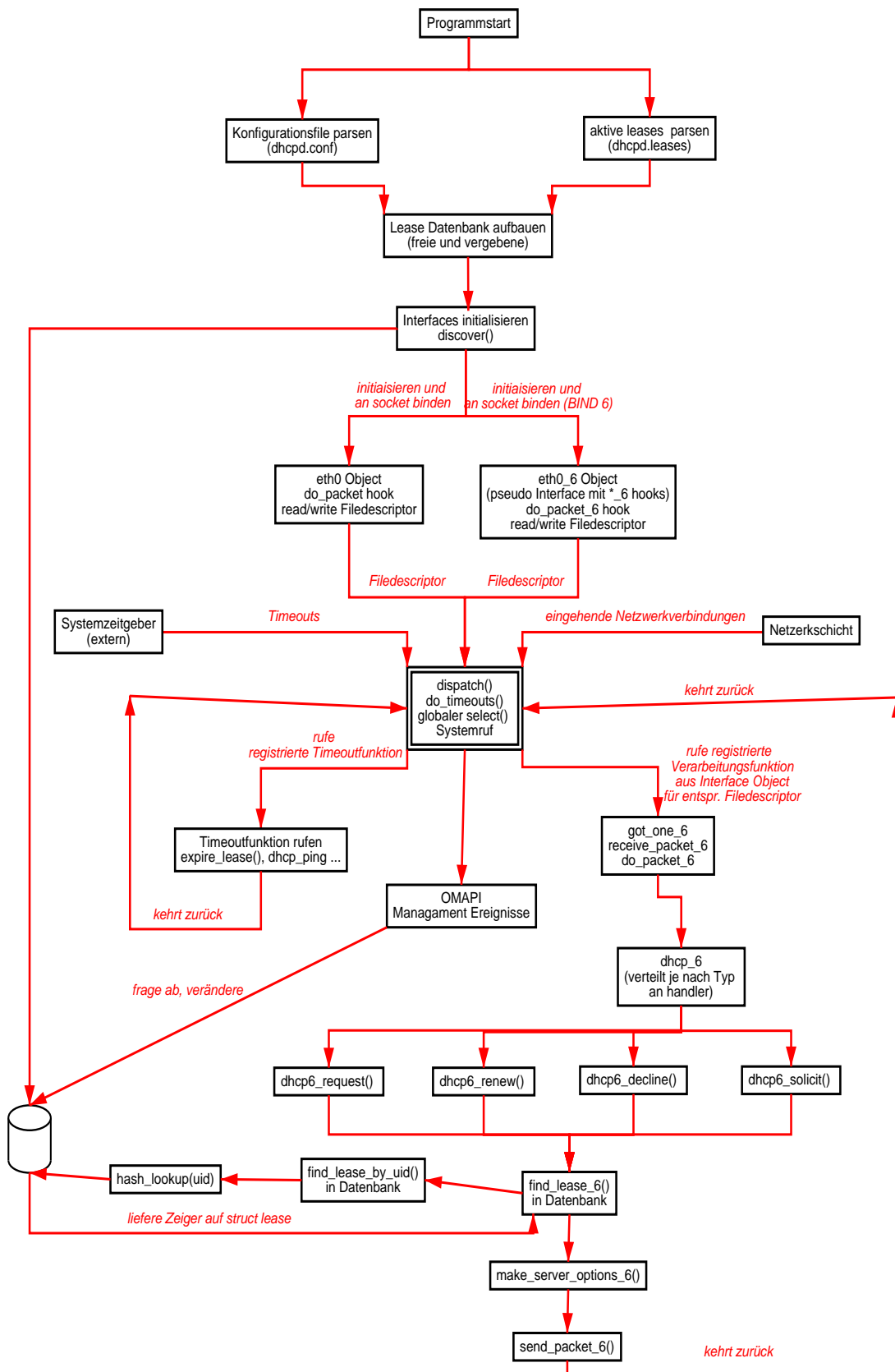


Abbildung 5.1: schematischer Aufbau des ISC DHCP Servers mit DHCPv6 Erweiterungen

für protokollbedingt andere Funktionen wie z.B. *dhcp_6()*. Im Klient kommt die Nutzung der Timeoutfunktionen besonders zum Tragen, was folgendes Beispiel veranschaulicht.

Nach dem Start des DHCP Klienten muss eine SOLICIT Nachricht an die Multicastgruppe ALL_SERVER geschickt werden. Und zwar solange, bis geeignete ADVERTISE Nachrichten von einem DHCP- Server empfangen wurden. Dazu wird für die Funktion *send_solicit_6()* mit *add_timeout()* ein Timeout vereinbart, das regelt, wie lange und in welchem Intervall der Sendeversuch erfolgen soll. Die *dispatch()* Funktion kehrt dafür nach abgelaufenem Intervall aus dem *select()* zurück und ruft die in *add_timeout()* vereinbarte Funktion. Wenn Advertisements empfangen wurden, wird das Timeout mit *cancel_timeout()* für *solicit_send_6()* abgebrochen und der Klient kann in den nächsten Zustand übergehen. In diesem Fall geht er in den Zustand Selecting. (Vgl. dazu auch Abbildung 4.5)

5.2.1 Duplicate Address Detection

Der DHCPv6 Standard schreibt vor, dass vor dem Übernehmen einer neu erhaltenen Adresse diese der Duplicate Address Detection (DAD) unterzogen werden muss. Bei der DAD handelt es sich um eine Anwendung des Neighbor Solicitation Mechanismus. Dabei wird eine Neighbor Solicit Anfrage mit der undefinierten Quelladresse (::) an die Multicastgruppe ALL_NODES gesandt und eine bestimmte Zeit gewartet. Wird innerhalb dieser Wartezeit eine Neighbor Advertisementnachricht für diese Adresse empfangen, bedeutet dies, sie wurde schon einmal vergeben und ist damit nicht eindeutig. In diesem Fall wird ein Logdateieintrag erzeugt und die Adresse findet keine Verwendung.

| IPv6 Adresse | Interface | Präfix länge | Gültigkeit | DAD Status | Interface |
|----------------------------------|-----------|-----------------|------------|---------------|-----------|
| 00000000000000000000000000000001 | 01 | 80 | 10 | 80 | lo |
| 3ffe04000100f1010000000000000001 | 02 | 40 | 00 | 80 | eth0 |
| fe800000000000000210b5ffe01bc98 | 02 | 0a | 20 | 80 | eth0 |

Tabelle 5.1: Aufbau der Datei `/proc/net/if_inet6`

Sollte der DHCPv6 Klient feststellen, dass die DAD für eine Adresse fehlschlägt, so sendet er eine DECLINE Nachricht mit der entsprechenden Adresse an den Server.

Die DAD Prüfung wird bei dem von mir entworfenen Klienten durch den Linuxkern erledigt. Dieser führt die DAD Prüfung mit jeder neu zugewiesenen Adresse selbstständig aus und schreibt den entsprechenden Status in die Datei `/proc/net/if_inet6` im Prozessinformationsdateisystem (proc). Dieser Umstand wird vom *dhclient-script_6* dahingehend genutzt, dass es die neue Adresse per *ifconfig* zuweist, wartet und den Adressstatus aus dem Proc- Dateisystem liest. Ausserdem gibt das Skript gegebenenfalls eine Fehlermeldung an den DHCP- Klienten zurück, der dann seinerseits die DECLINE- Nachricht verschickt.

Nachfolgend die DAD Routine aus dem *dhclient-script_6*.

```

ifconfig $interface add $new_ip_address
sleep 5
exitvar="0"
convertresult="/bin/ipv6calc --addr_to_ifinet6 $new_ip_address"
test_addr="echo $convertresult | awk '{ print $1 }'"
grep $test_addr /proc/net/if_inet6 |(while read hexaddr dummy1
                                hexprefixlenth he xscope dup device;
do
    if [ "$dup" != "80" ];
    then echo "duplicate detected! removing address"
    ifconfig $interface del $new_ip_address
    exit 1;
    break;
fi;
done;)
exitvar=$?;
if [ "$exitvar" != "0" ];
then exit 1;
fi;

```

5.2.2 Verbindung zwischen Adresskonfigurationssystem im Systemkern und DHCPv6 Klient

Der Zusammenhang zwischen DHCPv6, dem Adressautokonfigurationsmechanismus und der Adressgültigkeitsverwaltung im Systemkern selbst stellt für den DHCPv6 Klienten noch ein Problem dar. Normalerweise verwaltet das System die Gültigkeitszeiten der Adressen (preferred, valid lifetimes) selbst, sobald der Klient dem System eine erhaltene Adresse zugewiesen hat. Der DHCP Klient muss diese Zeiten ebenfalls verwalten, um z.B. bei Bedarf Adressenlebenszeiten zu verlängern. Hat der Klient keine Kenntnis der genauen Verwaltung innerhalb des System, kann es zu folgenden Fehlersituationen kommen:

- Der Klient erneuert die Adressen zu spät, die Lebenszeit im System ist dann schon abgelaufen.
- Der Klient entzieht Adressen, obwohl noch Verbindungen bestehen.

Um dies zu vermeiden, muss im System eine standardisierte Schnittstelle zum IPv6 Adressmanagement geschaffen werden. Dann könnten die Gültigkeitszeiten gemeinsam verwaltet werden, und der DHCPv6 Klient wäre in der Lage, den Status der Adressen direkt abzufragen und zu beeinflussen. Diese Schnittstelle sollte direkt über Systemfunktionen bereitgestellt werden, ohne den Umweg z.B. über ein */proc* Dateisystem zu nehmen.

Bisher existiert in keiner IPv6 Implementation solch eine Schnittstelle, und auch im DHCPv6 Standard wird eine derartige Schnittstelle nicht erwähnt. Ohne solch eine Schnittstelle ist aber ein IPv6 Standard- konformer DHCPv6 Klient nicht implementierbar.

5.3 Testumgebung

Die Testumgebung zur Implementierung bestand aus zwei Rechnern mit RedHat Server System Version 7.1, Systemkern Version 2.4.2-2. Diese Kernversion enthält die IPv6 Implementation der Version 0.8. Zur Anzeige des IPv6 Netzwerkverkehrs mussten nachträglich aktuelle Versionen der libpcap (0.6.2-7) und tcpdump (3.6.2-7) installiert werden. Sonstigedeaktiviert Systemfunktionen und Softwarepakete blieben unverändert.

5.3.1 IPv6 Konfiguration

Die verwendete RedHat Version 7.1 bietet bereits eine entsprechende IPv6 Unterstützung an, diese ist allerdings standardmässig deaktiviert. Um sie zu aktivieren muss lediglich in der Datei */etc/sysconfig/network* die Zeile "NETWORKING_IPV6=yes" angefügt werden. Zudem muss durch Hinzufügen von "IPV6INIT="yes"" in */etc/sysconfig/network-scripts/ifcfg-ethX* die IPv6 Unterstützung für jedes Netzwerkinterface einzeln aktiviert werden. Nur in diesem Fall bekommt mit dem Systemstart jedes entsprechend aktivierte Interface auch eine IPv6 link-local Adresse zugewiesen. Weitere IPv6 Adressen lassen sich ebenfalls durch Anfügen z.B. von "IPV6ADDR="3ffe:400:100:f101::1/64""in */etc/sysconfig/network-scripts/ifcfg-ethX* für jedes Interface festlegen.

5.3.2 Besonderheiten der IPv6 Implementation in Linux

Derzeit gibt es zwei aktuelle und ständig weiterentwickelte OpenSource IPv6 Implementationen. Zum einen ist das die KAME Implementation [KAME] für Betriebssysteme der BSD Familie (FreeBSD, OpenBSD, NetBSD) und zum anderen die Linuximplementation. Die bisher am weitesten fortgeschrittene und standardkonforme ist die KAME Implementation. Das KAME Projekt wird von sieben japanischen Technologiekonzernen unterstützt (z.B. NEC, Hitachi), die sich die Förderung von IPv6 auf die Fahnen geschrieben haben. Für Linux existieren Systemerweiterungen des USAGI Entwicklerteams [USAGI], dessen Ziel die Entwicklung einer standardkonforme IPv6 Unterstützung von Linux ist. USAGI orientiert sich dabei sehr stark an den Entwicklungen des KAME Projektes. Die USAGI Systemerweiterungen sind nicht im Standardlinuxkern enthalten. Ein Nachteil der USAGI Erweiterungen ist, dass sie nur bei bestimmten Standardkernen funktionieren. Deshalb kommen sie bei kommerziellen Distributionen wie RedHat oder Suse nicht standardmässig zum Einsatz.

Dies hat zur Folge, dass die Linux IPv6 Implementation, bestimmte Schnittstellen nicht unterstützt und teilweise nicht standardkonform agiert, wie in Konformitätstest (siehe [TAHI]) nachzulesen ist. Das wiederum bedeutet, dass Software zwischen den einzelnen IPv6 Plattformen nicht ohne teilweise umfangreiche, systemspezifische Anpassungen ausgetauscht werden kann. In der vorliegenden Implementation von DHCPv6 betrifft dies z.B. das Auslesen der link-local Adresse. Mit Linux ist es demnach nur möglich, über umständliches Auswerten des */proc* Dateisystems die aktuellen Adressen eines Interfaces auszulesen. KAME und USAGI basierte Systeme unterstützen aus diesem Grund einen eigenen Funktionsaufruf (*getifaddr()*).

5.4 Test des DHCPv6 Prototyps

Um das Funktionieren von DHCPv6 Server und Klient miteinander zu testen, ist es notwendig zwei entsprechend den Anforderungen konfigurierte Testsysteme zu installieren. Es muss IPv6 aktiviert, die für den Server erforderliche Konfigurationsdatei in */etc/dhcpd.conf* angelegt und Klient (dhclient) und Server (dhcpd) aus dem Verzeichnis */dhcpv6/work-linux-2.2/client* bzw. *server* gestartet werden. Empfehlenswert ist in beiden Fällen die Angabe der Option *-d* zur Anzeige von Log- Meldungen.

5.4.1 DHCPv6 Server Konfiguration

Die DHCPv6 Server Konfigurationsdatei enthält neben den normalen DHCPv4 Parametern auch die für DHCPv6. Für DHCPv6 Klienten ist es derzeit noch notwendig, eine eigene Host Deklaration anzulegen, in der die DUID und die zuzuweisende IPv6 Adresse steht.

Um die DAD zu testen, ist es am einfachsten, dem Klient eine schon vorhandene link- local Adresse des Subnetzes zuzuweisen. Der Klient sollte dann die Adresse ablehnen. Laut bisherigem DHCPv6 Standard werden vom Server nur vollständige IPv6 Adressen vergeben, es sind also keine Präfixe im Feld "fixed-ip6-address" erlaubt.

Nachfolgend die DHCPv6 Server Beispielkonfigurationsdatei, */etc/dhcpd.conf*

```
#subnet deklaration for eth0_6
subnet 0.0.0.0 netmask 255.255.255.0
{
    range 0.0.0.1 0.0.0.2;
}
host freebsd { option dhcp-client-identifier "87878";
                fixed-ip6-address "3ffe:400:100:f101:280:c8ff:fe4e:20d2" ;
                #fixed-ip6-address "fe80::280:c8ff:fe4e:20d2" ;
}
```

Abkürzungsverzeichnis

| | |
|----------|---|
| AAA | Authentifizierung, Autorisation, Accounting |
| ACL | Access Control List |
| ADSL | Asynchron Digital Subscriber Line |
| ARP | Address Resolution Protocol |
| AP | Access Point |
| AVP | Attribute Value Pairs |
| BOOTP | Boostrap Protocol |
| COPS | Common Open Policy Service Protocol |
| DAD | Duplicate Address Detection |
| DDL | Druckdienstleister |
| DE | Diensterbringer |
| DFN | Deutsches Forschungsnetz |
| DHCP | Dynamic Host Configuration Protocol |
| Diffserv | Differentiated Services |
| DNS | Domain Name System |
| DoS | Denial of Service |
| DSL | Digital Subscriber Line |
| DSTM | Dual Stack Transition Mechanismus |
| DTI | Dynamic Tunneling Interface |
| DUID | DHCP User Identifikator |
| EAP | Enhanced Authentication Protocol |
| EUI | Extended Unique Identifier |

| | |
|--------|--|
| FP | Format Präfix |
| HTTP | Hyper Text Transfer Protocol |
| HLR | Home Location Register |
| IA | Identity Association |
| ICMP | Internet Control Message Protocol |
| ID | Identifikator |
| IMEI | International Mobile- Station Equipment Number |
| IMSI | International Mobile Subscriber Identity |
| IP | Internet Protocol |
| ISAK | Individual Subscriber Authentication Key |
| ISC | Internet Software Consortium |
| ISP | Internet Service Provider |
| LAN | lokales Netz (Local Area Network) |
| MAC | Media Access Control |
| NAI | Network Access Identifikator |
| NAS | Network Access Server |
| NIS | Network Information System |
| NLA | Next Level Aggregator |
| PAM | Plugable Authentifikation Module |
| PIN | Persönliche Identifikations Nummer |
| PKI | Public Key Infrastructure |
| PPP | Point to Point Protocol |
| RADIUS | Remote Authentication Dial in User Service |
| RAM | Random Access Memory |
| ROM | Read Only Memory |
| RRP | Router Renumbering Protocol |
| RSVP | Ressource Reservation Protokoll |
| SCTP | Stream Control Transmission Protocol |
| SIM | Subscriber Identity Module |

| | |
|--------|--|
| SLA | Site Level Aggregator |
| SLP | Service Location Protokoll |
| SNMP | Simple Network Management Protocol |
| TCP | Transmission Control Protocol |
| TACACS | Terminal Access Controller Access Control System |
| TLA | Top Level Aggregator |
| TLS | Transport Layer Security |
| UDP | User Datagram Protocol |
| URI | Uniform Ressource Identifier |
| URZ | Universitätsrechenzentrum |
| USB | Universal Serial Bus |
| VLAN | Virtuelles LAN |
| WEP | Wired Equivalent Privacy |
| WWW | World Wide Web |

Abbildungsverzeichnis

| | | |
|------|--|----|
| 2.1 | SecurID Token | 11 |
| 2.2 | Public Key Infrastruktur | 12 |
| 2.3 | Erste Ticketanfrage | 14 |
| 2.4 | Dienstticket Anforderung | 15 |
| 2.5 | COPS Aufbau | 15 |
| 2.6 | AAA im GSM- Standard | 21 |
| 2.7 | Beispiel eines USB Hardwaretokens | 22 |
| 2.8 | AAA- Infrastruktur | 23 |
| 2.9 | Microsoft Passport System | 24 |
| 2.10 | AAA- Broker Struktur | 25 |
| 2.11 | Diameter als AAA- Infrastruktur | 27 |
| 2.12 | 802.1X Protokollablauf (nach [ABOBA 2000]) | 30 |
| 3.1 | URZ Netzwerkstruktur | 33 |
| 3.2 | Portmanager Prinzip | 35 |
| 3.3 | Funknetz AAA- System | 36 |
| 3.4 | DFN@HOME Einwahl | 37 |
| 3.5 | vorgeschlagene AAA- Infrastruktur | 38 |
| 3.6 | Nutzerauthentifizierung per Diameter | 38 |
| 4.1 | Anycast Beispiel | 47 |
| 4.2 | Zeitlinie zustandsloser Adresskonfiguration | 48 |
| 4.3 | Zeitlinie DHCPv6 Konfigurationsablauf | 53 |
| 4.4 | DHCP über einen Router als Relay | 54 |
| 4.5 | Klienten Zustandsdiagramm | 56 |
| 4.6 | DHCP Server/ AAA Server Beziehung | 60 |
| 4.7 | DSTM Mechanismus (nach [EURESCOM 2000]) | 61 |
| 5.1 | schematischer Aufbau des ISC DHCP Servers mit DHCPv6 Erweiterungen | 67 |

Tabellenverzeichnis

| | | |
|------|---|----|
| 2.1 | ISO-OSI Schichtenmodell | 17 |
| 2.2 | Diameter Kopf Format | 28 |
| 2.3 | AVP Kopf Format | 28 |
| 2.4 | EAP Paketaufbau | 29 |
| 4.1 | IPv6 Adressstruktur | 44 |
| 4.2 | site- local Adressen | 45 |
| 4.3 | Link- local Adressen | 45 |
| 4.4 | Multicast Adressaufbau | 46 |
| 4.5 | Multicast Gültigkeitsbereiche | 46 |
| 4.6 | Router Advertisement Message Format | 48 |
| 4.7 | Präfix Information Option | 49 |
| 4.8 | DHCPv6 Nachrichtenaufbau | 51 |
| 4.9 | DUID Option | 52 |
| 4.10 | Identity Association Option | 52 |
| 4.11 | Options Request (ORO) Option | 52 |
| 4.12 | DNS Server Option | 52 |
| 4.13 | Relay Forward Nachricht | 54 |
| 4.14 | Klient Nachricht Option | 55 |
| 4.15 | Aufbau der Authentifizierungsoption | 59 |
| 4.16 | DSTM Global IPv4 Adress Option | 61 |
| 5.1 | Aufbau der Datei /proc/net/if_inet6 | 68 |

Literaturverzeichnis

- [MIT 1987] NIP MIT Athena Network Information Protocol Draft <http://web.mit.edu/afs/net/project/dhcp/doc/athena-nip>.
- [BRADLEY 1998] T. Bradley, u.a. : RFC 2390 "Inverse Address Resolution Protocol", September 1998.
- [CROFT 1985] Bill Croft, John Gilmore : RFC951 "BOOTSTRAP PROTOCOL (BOOTP)", September 1985.
- [DROMS 1993] R. Droms : RFC 1541 "Dynamic Host Configuration Protocol", October 1993.
- [BREILER 2000] Andre Breiler : Diplomarbeit "Portmanager System", Oktober 2000.
- [NW 14/99] IEEE 802.1x "Kontrolle auf Portebene! Network World 14 /S. 13
- [TRAPP 1999] Script zur Vorlesung "Rechnernetzsicherheit" H. Trapp, 1999.
- [ABOBA 1999] B. Aboba, M. Beadles: RFC 2486 "The Network Access Identifier", Januar 1999.
- [KNIVETON 1999] Timothy J. Kniveton, Jari T. Malinen: "SIM Authentication EAP extension over AAAv6", Juli 2001
- [CALHOUN 2001] Pat R. Calhoun u. a.: "Diameter Base Protocol", Juli 2001.
- [RIGNEY 2000] C. Rigney u.a. : RFC 2865 "Remote Authentication Dial In User Service (RADIUS)", Juni 2000.
- [CALHOUN2 2001] Pat R. Calhoun u.a.: "Diameter NASREQ Application", Juli 2001.
- [CALHOUN3 2001] P. Calhoun, C. Perkins : "Diameter Mobile IP Application", Juli 2001
- [CALHOUN4 2001] P. Calhoun, W. Bulley, S. Farrell : "Diameter CMS Security application", Juli 2001.
- [CAREL 1997] D. Carrel, Lol Grant : "The TACACS+ Protocol", Januar 1997.
- [MUKHERJEE 2001] B. Mukherjee, B. Gage, Y. Liu : "Extensions to DHCP for Roaming Users", Februar 2001.

- [TSIRTSIS 2001] G. Tsirtsis, J. Privat : "Triggering AAA from DHCP Relay Agents", Januar 2001.
- [DURHAM 2001] D. Durham u.a.: RFC 2748 "The COPS (Common Open Policy Service) Protocol", Januar 2001.
- [NARTEN 2001] Thomas Narten, Richard Draves : "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", Juli 2001.
- [MILLS 1991] C. Mills, D. Hirsh, G. Ruth : RFC 1272 "INTERNET ACCOUNTING: BACKGROUND", November 1991.
- [STEWART 2000] R. Stewart, Q. Xie u.a.: RFC 2960 "Stream Control Transmission Protocol", Oktober 2000.
- [VIXIE 1997] P. Vixie u.a.: "Dynamic Updates in the Domain Name System (DNS UPDATE)", April 1997.
- [EURESCOM 2000] Eurescom Survey Of Transition mechanisms <http://www.eurescom.de/~public-webospace/P1000-series/P1009/index.html>, Juli 2000.
- [GILLIGAN 1999] R. Gilligan, S. Thomson, J. Bound, W. Stevens : RFC 2553 "Basic Socket Interface Extensions for IPv6", März 1999.
- [CRAWFORD 2000] M. Crawford : RFC 2894 "Router Renumbering for IPv6", August 2000.
- [ABOBA 2000] B. Aboba, Tim Moore : doc IEEE 802.11-00/035, März 2000.
- [DRAVES 2001] Richard Draves : "Default Address Selection for IPv6", Juni 2001.
- [AIRSNORT 2001] Airsnort Entwicklungsseite <http://airsnort.sourceforge.net>, August 2001.
- [DROMS 2001] J. Bound, C. Perkins, R. Droms : Internet Draft "Dynamic Host Configuration Protocol for IPv6" Rev. 19, August 2001.
- [USAGI] USAGI Project - Linux IPv6 Development Project <http://www.linux-ipv6.org/>
- [KAME] KAME Project <http://www.kame.net/>
- [TAHI] TAHI Project <http://www.tahi.org/>
- [PKI DRAFT] PKI IETF Draft <http://www.opengroup.org/public/tech/security/pki/index.htm>
- [KOHL 1993] J. Kohl, C. Neuman : RFC 1510 "The Kerberos Network Authentication Service", September 1993.
- [DIERKS 1998] Dierks, T., and C. Allen : RFC2246 "The TLS Protocol Version 1.0", November 1998.

Selbständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbständig und ohne unzulässige Hilfe verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.