Bachelorarbeit

Hermand Dieumo Kenfack

**Ein System für die Inbetriebnahme
und die Konfiguration
eines eingebetteten TCP/IP-Netzwerks**

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Hermand Dieumo Kenfack

Ein System für die Inbetriebnahme und die Konfiguration eines eingebetteten TCP/IP-Netzwerks

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Technische Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Gunter Klemke

Zweitgutachter: Prof. Dr. Thomas Canzler

Abgegeben am 30. Juni 2009

Hermand Dieumo Kenfack

Thema der Bachelorarbeit

Ein System für die Inbetriebnahme und die Konfiguration eines eingebetteten TCP/IP Netzwerks

Stichworte

Embedded TCP/IP-Netzwerk, Konfiguration, Sicherheit, RTOS, SDeDi-DHCP, .Net.

Kurzzusammenfassung

Zur Vernetzung eines eingebetteten Systems mit IP ist ein durchdachtes Konzept für die Inbetriebnahme und die Konfiguration von großer Bedeutung. Dies ist für eine reibungslose Funktionalität und Robustheit des Netzwerks unabdingbar.

In dieser Arbeit ist ein System entwickelt worden, welches die Inbetriebnahme und die Konfiguration eines eingebetteten TCP/IP-Netzwerks automatisiert, sichert und vereinfacht.

Hermand Dieumo Kenfack

Title of the paper

A system for the commissioning and the configuration of an embedded TCP/IP-network

Keywords

Embedded TCP/IP-Network, Configuration, Security, RTOS, SDeDi-DHCP, .Net.

Abstract

For the networking of an embedded system with IP is a thoughtful concept for the commissioning and the configuration of great importance. This is for a smooth functionality and robustness of the network indispensable.

In this work a system was developed which automates, secures and simplifies the commissioning and the configuration of an embedded TCP/IP network.

Danksagung

Hiermit möchte ich mich bei allen bedanken, die mich bei der Erstellung dieser Arbeit unterstützt haben. Insbesondere gilt mein Dank

der Firma detectomat GmbH für die Ermöglichung dieser Arbeit und die zur Verfügung gestellten Ressourcen,

Herrn Prof. Dr. Gunter Klemke (HAW-Hamburg) und
Herrn Dipl.-Ing. Daniel Schürmann (Fa. detectomat) für ihre hervorragende Betreuung,

Frau Gisela Juraschka, Lehrerin a. D., und
Herrn Dipl.-Ing. Eduard Klaus für das Korrekturlesen,

und meine Eltern für ihre moralische Unterstützung.

Inhaltsverzeichnis

1 Einführung

Eingebettete Systeme (engl. Embedded Systems) stellen einen Bereich in der Informationstechnologie dar, der zu den am stärksten wachsenden gehört. Weit über 90% aller Prozessoren sind heute nicht in Computern, sondern in anderen technischen Geräten eingebaut. Sie finden sich nicht nur in elektrischen Haushaltsgeräten, Autos, sicherheitstechnischen Geräten (Brandmeldeanlagen, Einbruchmeldeanlagen, etc.) oder Kommunikationswerkzeugen, sondern in beinahe allen technischen Produkten verschiedenster Lebensbereiche. Aufgrund ihrer Allgegenwart und Vielfältigkeit ist es schwierig, eine genauere Definition dafür anzugeben. Sie sind jedoch im Gegensatz zu herkömmlichen Computern einfache Rechner mit bestimmter Funktionalität, die in zu steuernden oder zu überwachenden Systemen integriert (eingebettet) sind. Sie bestehen aus einer Kombination von Hard- und Software (Mikrocontroller, Mikroprozessor, Bussysteme, etc.) und enthalten häufig Einheiten, die Aufgaben der Sensorik und Aktorik erfüllen und Kommunikationsschnittstellen besitzen. Darunter sind nicht nur „Mensch-Geräte“-Schnittstellen zu verstehen, sondern auch Vernetzungsmöglichkeiten zwischen den Geräten (vgl. FIT-IT).

Die Vernetzung der Geräte ermöglicht den Aufbau effizienter Systeme. Ein Beispiel wäre die Vernetzung aller Brandmeldeanlagen in einem Großgebäude oder von verschiedenen Gebäuden. Dadurch können bei einem Feueralarm in einem Stockwerk benachbarte Stockwerke bzw. Gebäude darüber informiert werden.

Der Trend ist, mehr und mehr Technologien aus der Informationstechnik für die Vernetzung von Geräten der Automatisierungstechnik einzusetzen. Zum Beispiel wird für einen schnellen und preiswerten Übertragungsweg gern Ethernet und IP eingesetzt (z. B. LON over IP). Das Problem ist, dass die IP-Technik von vornherein nicht für die Automatisierungstechnik konzipiert worden ist. Es gibt kein standardisiertes Protokoll, das auf die Inbetriebnahme/Konfiguration von IP-basierten Automatisierungsnetzwerken zugeschnitten ist. Deshalb müssen die für IP-Netze vorgesehenen Inbetriebnahme/Konfigurations-Protokolle überarbeitet und auf die Automatisierungsnetzwerke zugeschnitten werden. Diese Arbeit widmet sich der Aufgabe, ein System für die Inbetriebnahme und die Konfiguration des auf TCP/IP (Transmission Control Protocol/Internet Protocol) und Ethernet basierenden Brandmeldezentrale-Netzwerks (BMZ-TCP/IP-Netzwerk) der Firma detectomat GmbH zu entwicklen.

1.1 Aufgabenstellung

Die BMZs der Fa. detectomat kommunizieren momentan über Bit-Bus oder CAN-Bus und sollen zusätzlich mit TCP/IP über Ethernet vernetzt werden (Abbildung 1-1). Dafür soll ein System entworfen und implementiert werden, das das BMZ-TCP/IP-Netzwerk automatisch in Betrieb nimmt und konfiguriert.

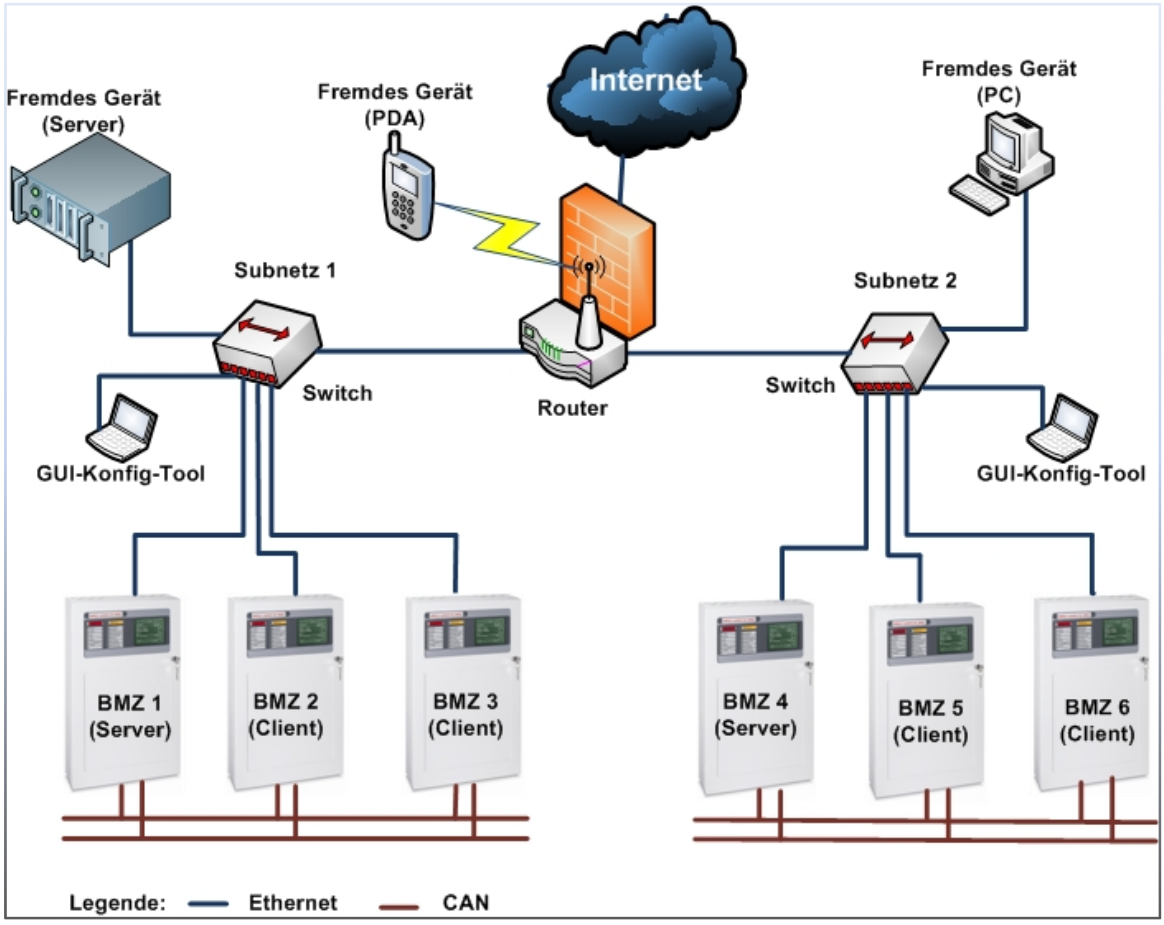


Abbildung 1-1: Vernetzung von BMZ mit Ethernet und redundantem CAN

Die BMZs befinden sich in einem Subnetzwerk mit anderen Geräten (PC, PDA, andere technische Geräte, etc.), müssen aber einen IP-Bereich aus diesem Subnetzwerk zugewiesen bekommen, welcher mithilfe des Systems zur Inbetriebnahme/Konfiguration des Netzwerks verwaltet wird. Das heißt, dieser Bereich darf nicht von unbekannten Konfigurationstools oder im Netz vorhandenen Servern (z. B. DHCP-Server) verwaltet werden.

Die Inbetriebnahme/Konfiguration des Netzwerks soll von einem Errichter mithilfe eines benutzerfreundlichen GUI-Konfig-Tools (Graphical User Interface Konfigurationstool) durchgeführt werden. Da dies durch wechselnde Errichter geschieht, die außerdem mobile Rechner benutzen/besitzen, dürfen die Konfigurationsdaten des Netzwerks nicht auf

dem Rechner eines Errichters gespeichert werden, sondern auf einer BMZ, die dann als Server operieren soll und die Konfigurationsdaten des Netzwerks bzw. der Geräte nach Bedarf zur Verfügung stellt.

Bei der Erstinbetriebnahme haben die BMZs die gleiche werkseingestellte IP-Adresse. Die Inbetriebnahme soll aber auch mit unbekannten oder undefinierten IP-Adressen (0.0.0.0) funktionieren. Darum muss ein Mechanismus entwickelt werden, der dazu dient die BMZs unabhängig von deren IP-Adressen aufzufinden.

Die Errichter sind keine TCP/IP-Netzwerk „Profis“. Es kann daher nicht erwartet werden, dass sie alle komplexen Aufgaben durchführen können, wie die Berechnung der Subnetz-Maske, den manuellen Eintrag der Geräte-IP-Adressen etc., die zur Inbetriebnahme/Konfiguration eines TCP/IP-Netzes benötigt werden. Aus diesem Grund soll die Inbetriebnahme/Konfiguration des Netzwerks automatisch erfolgen.

Für eine sichere Erreichbarkeit der Brandmeldezentralen sollen für deren Konfiguration nur statische IP-Adressen und statische Namen verwendet werden.

Ist das Netzwerk bereits in Betrieb genommen, soll es möglich sein, ein neues Gerät im Netzwerk zu integrieren.

Der wichtigste Faktor in der Brandmeldetechnik bleibt die Sicherheit. Deshalb dürfen die Konfigurationsparameter der BMZs nicht von fremden Geräten (Servers, Notebooks, PDA, etc.) geändert werden können. Außerdem muss sichergestellt werden, dass die empfangenen bzw. gesendeten Daten nicht von einem dritten Gerät geändert worden sind. Daher müssen alle Sicherheitsaspekte für den Zugriff auf die BMZs für eine Inbetriebnahme/Konfiguration und die Kommunikation zwischen diesen untersucht werden.

Die BMZs der Fa. detectomat operieren mit eCos (Embedded Configurable Operating System). Daher müssen die Implementierungen des Servers und des Clients unter eCos laufen.

Um sich an die Konformität von IP zu halten, muss sich die Konzeption des Systems hauptsächlich auf Standard IP-Protokolle stützen. Ein weiterer Grund dafür ist das Vermeiden der Falschinterpretation von Broadcast-Nachrichten von fremden Geräten.

1.2 Zielsetzung

Ziel dieser Arbeit ist die Entwicklung eines Systems für die Inbetriebnahme und die Konfiguration des BMZ-TCP/IP-Netzwerks der Fa. detectomat.

Erwähnenswert ist, dass die in dieser Arbeit vorgestellte Lösung nicht nur für das Netzwerk der Fa. detectomat einsetzt werden kann, sondern auch für jedes andre Embedded-TCP/IP-Netzwerk, das eine automatische und sichere Inbetriebnahme/Konfiguration benötigt. Deshalb werden in dieser Arbeit weiterhin nur noch allgemein die Begriffe Embedded Device, Gerät, Embedded-TCP/IP-Netzwerk verwendet. Jedoch werden weiterhin BMZs als Beispiele zur Unterstützung in allen Phasen (Anforderung, Entwurf, Implementierung, Test, etc.) der Entwicklung des Systems herangezogen.

Die Inbetriebnahme/Konfiguration eines Embedded-Geräts über IP erfolgt meistens über ein „Web-Interface“ oder Konfigurationstool von einem PC bzw. Notebook aus. Dafür muss das Gerät erreicht werden. Um ein Gerät erreichen zu können, müssen dessen Zugriffsinformationen bekannt sein, nämlich die IP-Adresse oder das FQDN (Fully Qualified Domain Name) mit Authentifizierung. Sind diese Informationen bekannt, werden alle Konfigurationsparameter über das Web-Interface oder Konfigurationstool eingeben. Das ist bei mehreren Geräten aufwendig und kann fehlerbehaftet und zeitintensiv sein. Diese Aufgabe erfordert fortgeschrittene IP-Kenntnisse, die nicht von jedem technischen Errichter erwartet werden können. Die Lösung ist ein im Rahmen dieser Bachelorarbeit entwickeltes System, das die Inbetriebnahme und Konfiguration des Netzwerks erleichtert, automatisiert und sichert, indem es folgende Funktionen bereitstellt:

- Mechanismen zur Auffindung und zur automatischen Konfiguration der Geräte,
- ein benutzerfreundliches GUI-Tool zur Darstellung und Erleichterung der automatischen Konfiguration,
- Mechanismen zur Sicherheit (security) bezüglich Zugangsberechtigung und Datenintegrität während der Konfiguration,
- Mechanismen zur Geräte-Ausfallerkennung,
- Mechanismen zur Plug-and-Play Fähigkeit der Geräte.

Die Konzeption basiert hauptsätzlich auf für die Konfiguration von IP-Netzen definierten Protokollen bzw. Standards. Diese werden in Kapitel 3 beschrieben und stellen die Grundlagen dieser Arbeit dar.

1.3 Motivation

Embedded Systems bzw. technische Geräte sollten nicht nur über technische Bussysteme (CAN-Bus, Profibus, LON, BITBUS, …) vernetzt werden, sondern auch zusätzlich oder besser ausschließlich mit IP über Ethernet, und zwar aus folgenden Gründen:

- Interoperabilität zwischen heterogenen Geräten
- Weltweite Erreichbarkeit der Geräte
- Fernzugriff (über Web-Interface, SSH (Secure Shell) etc.)
- Ferndiagnose, Fernmanagement
- Einsatz von Standard Tools für Diagnose, Management und Monitoring
- Einfaches Routing
- Ausnutzung der vorhandenen Internet-Technologien bzw. -dienste (E-Mail, FTP, Web-Server, ...)
- Ausnutzung der vorhandenen IP-Infrastrukturen (WLAN, LAN, WAN, Switche, Router, …)
- Vereinfachung der Firmware-Updates
- Vereinfachung der Service-Discovery (z. B. für neu im Netz angeschlossene Geräte)
- Embedded Geräte mit wenig Speicher können ihre Logs zu einem Log-Server senden
- …

Der Einsatz eines Systems zur automatischen Inbetriebnahme/Konfiguration eines Embedded-TCP/IP-Netzwerks hat viele Vorteile:

- Inbetriebnahme/Konfiguration des Netzwerks wird erheblich vereinfacht.
- Fehler, die bei manuellen Konfigurationen auftreten könnten, werden vermieden.
- Adressen Konflikte werden automatisch erkannt und aufgelöst.
- Inkonsistente Konfigurationszustände werden automatisch erkannt und aufgelöst.
- Die Integration von neuen Geräten im Netzwerk wird erheblich vereinfacht.
- ...

1.4 Untersuchung vorhandener Lösungen

Der Entwicklung des in dieser Arbeit vorgestellten Systems ging eine Untersuchung eventuell vorhandener Lösungen zur Inbetriebnahme/Konfiguration von TCP/IP-Netzen voraus. Diese werden hier der gestellten Aufgabe gegenübergestellt.

Die bekannteste und meist benutzte Lösung zur Inbetriebnahme/Konfiguration von IP-Netzen ist der Einsatz des DHCP-Protokolls (Abschnitt 3.1). Dabei wird eine zentrale Stelle (DHCP-Server) zur Verwaltung und Verteilung von Geräte-Konfigurationsparametern eingerichtet. Geräte, die den DHCP-Server-Dienst zur Konfiguration ihrer Netzwerkschnittstelle benutzen wollen, müssen einen DHCP-Client installiert haben. Der DHCP-Client hat als Aufgabe, die Konfigurationsparameter des Geräts vom DHCP-Server anzufordern. Will man ein Netzwerk mit DHCP in Betrieb nehmen/konfigurieren, muss zuerst der DHCP-Server in Betrieb genommen werden. Dafür muss das Gerät, auf dem der DHCP-Server läuft, mit einer statischen IP-Adresse konfiguriert werden. Danach wird das Netzwerk konfiguriert. Dafür werden die Netzwerkeigenschaften des vom DHCP-Server zu verwaltenden IP-Adressenbereichs (scope) in einer Konfigurationsdatei eingetragen. Die Konfiguration der Netzwerkschnittstelle des Server-Geräts und des Netzwerks erfolgt „manuell“, das heißt, dass alle Einträge, die für eine Konfiguration nötig sind, von einem Netzwerkadministrator direkt oder indirekt über ein Konfigurationstool in einer Konfigurationsdatei eingegeben werden müssen. Ist der Server erfolgreich in Betrieb genommen worden, muss der DHCP-Client jedes Gerät einzeln gestartet werden. Nach dem Start fordert er die Konfigurationsparameter des Geräts vom Server und stellt sie entsprechend ein.

Es gibt bereits viele Implementierungen des DHCP-Protokolls. Die am häufigsten eingesetzten in der „PC-Welt“ sind der Windows DHCP-Client, welcher Bestandteil aller Windows-Betriebssysteme ist, der Windows-DHCP-Server, welcher durch den Servermanager von Windows-Server (2000, 2003, 2008) als Rolle hinzugefügt und konfiguriert werden kann (vgl. Will 2008, S.165-231). Für Linux Client und Server ist es das open-source ISC (Internet Systems Consortium) -DHCP (vgl. ISC). Der ISC-DHCP-Client kann als Service für die jeweilige Linux Distribution installiert werden, sodass er beim Start des Rechners automatisch gestartet wird. Die Konfiguration des Servers bzw. Netzwerks erfolgt hier entweder per direkten Eintrag in die Konfigurationsdatei oder indirekt durch ein Konfigurationstool. Unter openSUSE wird YaST (Yet another Setup Tool) zur Konfiguration der ISC-DHCP-Server eingesetzt. Es gibt aber Tools wie Webmin, die für diese Aufgabe unter allen Linux Distributionen eingesetzt werden können.

Für die gestellte Aufgabe können die oben genannten Lösungen jedoch für das System nicht eingesetzt werden, denn sie stellen keine Verfahren zum automatischen Auffinden und zur Inbetriebnahme der Geräte bereit. Damit die Konfiguration automatisch erfolgen kann, müssen zuerst viele manuelle Einstellungen vorgenommen werden (Konfiguration des Servers, Eintrag der Client-Konfigurationsparameter, etc.). Es geht hier um Embedded-RTOS, das den Einsatz eines Windows-DHCP-Servers bzw. -Client ausschließt. Die Tools, die zur Konfiguration der Server (Windows und ISC) eingesetzt werden, können nur auf

dem Server-Rechner installiert werden. Es wird aber ein Tool benötigt, das auf einem anderen Rechner installiert wird und mit dem Server über das Netzwerk kommuniziert.

In der Gebäudeautomation werden zur Inbetriebnahme/Konfiguration der Geräte alle Konfigurationsparameter über das Webinterface des jeweiligen Geräts, zentral in einem „speziellen“ Konfigurationsserver oder in einem DHCP-Server statisch und manuell eingegeben. Vom Einsatz von DHCP für technische Geräte wird meist abgeraten.

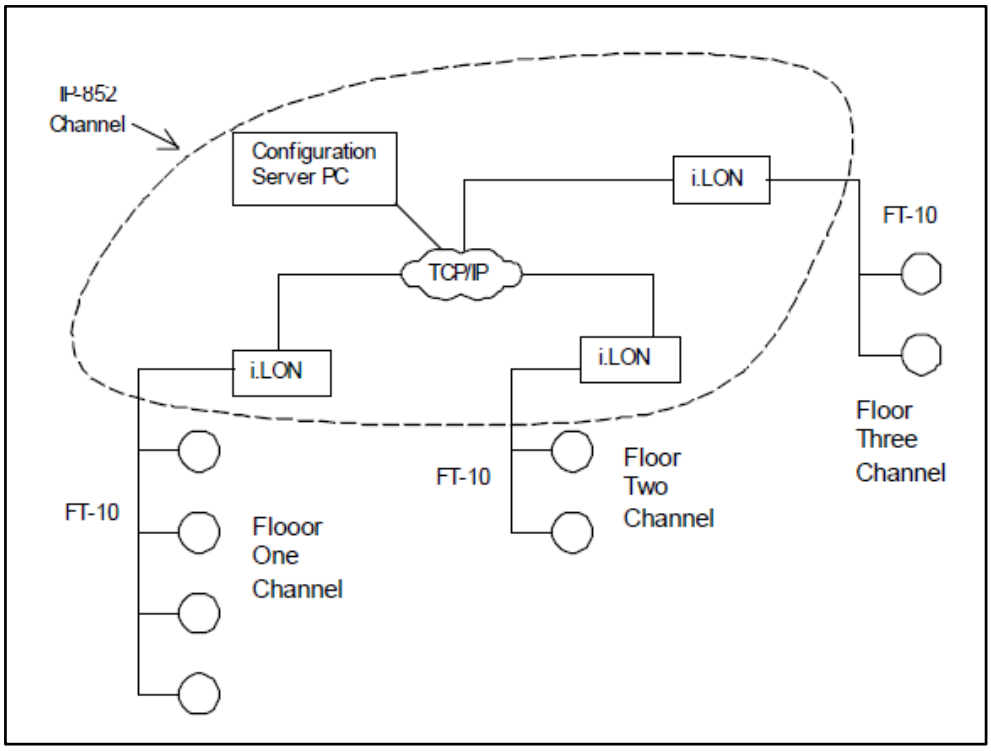


Abbildung 1-2: IP-852 Chanel [Echelon 2005, S.3]

Zur Erläuterung werden die i.LON-Servers von Echelon als Beispiel herangezogen (Vgl. Echelon 2005). Sie dienen als Gateways bei der Vernetzung von Geräten der Gebäudeautomation (Jalousiesteuerung, Raumregler, etc.) über Ethernet. Um die i.LON-Server (Abbildung 1-2) zur konfigurieren bzw. in Betrieb zu nehmen, muss jeder Server mit einer reservierten statischen IP-Adresse konfiguriert werden. Die statische Konfiguration erfolgt entweder manuell über das Webinterface des i.LON-Servers oder über einen Echelon LonWorks/IP Konfiguration Server, der auf einem PC installiert wird. Wird die Konfiguration jedoch über das Webinterface des Servers durchgeführt, muss dem PC eine IP-Adresse aus dem Subnetzwerk des Servers zugewiesen werden. Ein DHCP-Server kann auch für die Konfiguration von i.LON-Servern eingesetzt werden. Dafür müssen die Konfigurationsparameter jedes i.LON-Servers statisch am DHCP-Server eingetragen (static-leases) und die i.LON-Servers dann neu gestartet werden. Trotz des Aufwands und der Unvollständigkeit ist der Einsatz von DHCP-Servern mit „static-lease“-Einträgen eine ak-

zeptable Lösung. DHCP-Server werden jedoch von Netzwerkadministratoren verwaltet und deren Einsatz führt dazu, dass die technischen Errichter keine volle Kontrolle mehr über die Konfiguration der Geräte haben, was meist nicht gewollt ist.

Fazit: Es hat sich herausgestellt, dass keine der untersuchten Lösungen zur Inbetriebnahme/Konfiguration von TCP/IP-Netzen auf das hier gewünschte System zugeschnitten werden kann, da entweder proprietäre Lösungen oder DHCP eingesetzt werden. Eine neue Lösung musste gefunden werden.

2 Anforderungsdefinition

In diesem Kapitel werden die Anforderungen des Systems definiert. Zuvor werden einige wichtige Informationen über das System gegeben.

2.1 Ausgangszustand der Geräte

Vor einem Zugriff auf das Subnetzwerk für eine Konfiguration/Inbetriebnahme muss sichergestellt werden, dass die zu konfigurierenden Geräte betriebsbereit sind. Betriebsbereit heißt, dass alle Geräte an 230V angeschlossen und physikalisch oder logisch miteinander über Ethernet vernetzt sind. Jedes Gerät muss außerdem eindeutig identifizierbar sein. Die Identifizierung kann seine MAC (Media Access Control)-Adresse, Seriennummer oder IP-Adresse sein. Die Identifizierung kann auch durch eine Kombination dieser Parameter erfolgen. Bei einer Erstinbetriebnahme sind die IP-Adressen der Geräte die der Werkseinstellung. Jedes Gerät kann sowohl als Client als auch als Server operieren. Handelt es sich jedoch um eine Erstinbetriebnahme, operieren alle Geräte als Client. Dabei wird einer der Geräte als Server ausgewählt.

2.2 Beschreibung der Systemumgebung

Das System wird in einem IP-Subnetzwerk eingesetzt. Das heißt, dass alle zu konfigurierenden Geräte (z. B. BMZs aus Abbildung 1-1) sich physikalisch oder logisch in demselben Subnetzwerk befinden müssen. In diesem Subnetzwerk wird ein IP-Adressen-Bereich für die zu konfigurierenden Geräte reserviert. Besteht das Netzwerk aus mehreren Subnetzwerken (Abbildung 1-1) muss die Konfiguration/Inbetriebnahme für jedes Subnetzwerk separat durchgeführt werden.

Wie in Abbildung 1-1 zu sehen ist, können sich andere Netzwerkgeräte (Notebooks, PDAs, „Servers“, etc.) im selben Subnetzwerk befinden. Die zu konfigurierenden Geräte befinden sich nicht in einem exklusiven Subnetzwerk.

Da es sich um Technische bzw. Embedded Geräte handelt, könnten diese zusätzlich durch ein anderes Bussystem vernetzt sein (z. B. Abbildung 1-1: Vernetzung von BMZ mit Ethernet und redundantem CAN). Erwähnenswert ist, dass die Ethernet Vernetzung unabhängig von vorhandenen Bussystemen ist (kein Einsatz von Gateways).

Das Wort Netzwerk ist kontextsensitiv in dieser Arbeit. Netzwerk kann für ein gesamtes Netzwerk oder nur ein Subnetzwerk stehen.

2.3 Begriffsdefinition

Client	Applikation, die die Konfigurationsparameter der Netzwerkschnittstellen eines Geräts vom Server anfordert und die empfangenen Konfigurationsparameter entsprechend einstellt.
Client-Konfigurieren	Vorgang, bei dem der Server die vom Client geforderten Konfigurationsparameter aus der Konfigurationsdatei ausliest und dem Client sendet.
Errichter	Eine Person, die für die Inbetriebnahme/Konfiguration des Netzwerks zuständig ist, also auch für die Verwaltung des reservierten IP-Bereichs.
Konfigurationsparameter	Alle Informationen, die zur Konfiguration der Netzwerkschnittstelle eines Geräts benötigt werden. Sie bestehen aus den allgemeinen Netzwerkeigenschaften und den spezifischen Konfigurationsparametern des Geräts (IP-Adresse, MAC-Adresse, Name, …).
Konfigurationsserver (Server)	Applikation, die die Netzwerkkonfiguration speichert und verwaltet. Eine weitere Funktionalität des Konfigurationsservers ist die Client-Konfiguration.
Netzwerkadministrator	Eine Person, die für die Bereitstellung des für die zu konfigurierenden Geräte reservierten IP-Bereichs (scope) sowie die damit gebundenen Netzwerkeigenschaften zuständig ist. Er stellt außerdem sicher, dass keine IP-Adresse aus diesem Bereich einem anderen Gerät zugewiesen wird. Er ist zwar zuständig für das gesamte Netzwerk, hat aber keinen direkten Zugriff auf den reservierten IP-Bereich.
Netzwerkeigenschaften	Allgemeine Informationen (z. B.: scope (IP-Bereich), Subnet-Mask, Gateways, DNS-Server, Domain-Name …) über das Netzwerk bzw. Subnetzwerk, die für alle Geräte gelten.
Netzwerkkonfigurationsdaten	Alle Informationen über die Konfiguration des Netzwerks (Netzwerkeigenschaften und gerätespezifische

	Konfigurationsparameter), die sich in einer Konfigurationsdatei bzw. Datenbank befinden.
Netzwerk konfigurieren	Eintragen der Netzwerkeigenschaften sowie der spezifischen Konfigurationsparameter aller Geräte in einer Konfigurationsdatei bzw. Datenbank

2.4 Spezifikation der Systemprozesse

Die Spezifikation der Systemprozesse ist die Beschreibung der Anwendungsfälle (Use-Cases) des Systems (in Abbildung 2-1 dargestellt). Dies dient der klaren Beschreibung der Interaktion der Benutzer mit dem System, sowie die Funktionen, die das System leisten soll. Die Anwendungsfälle werden in tabellarischer Form angegeben (Verfahren aus Hruschka 2002).

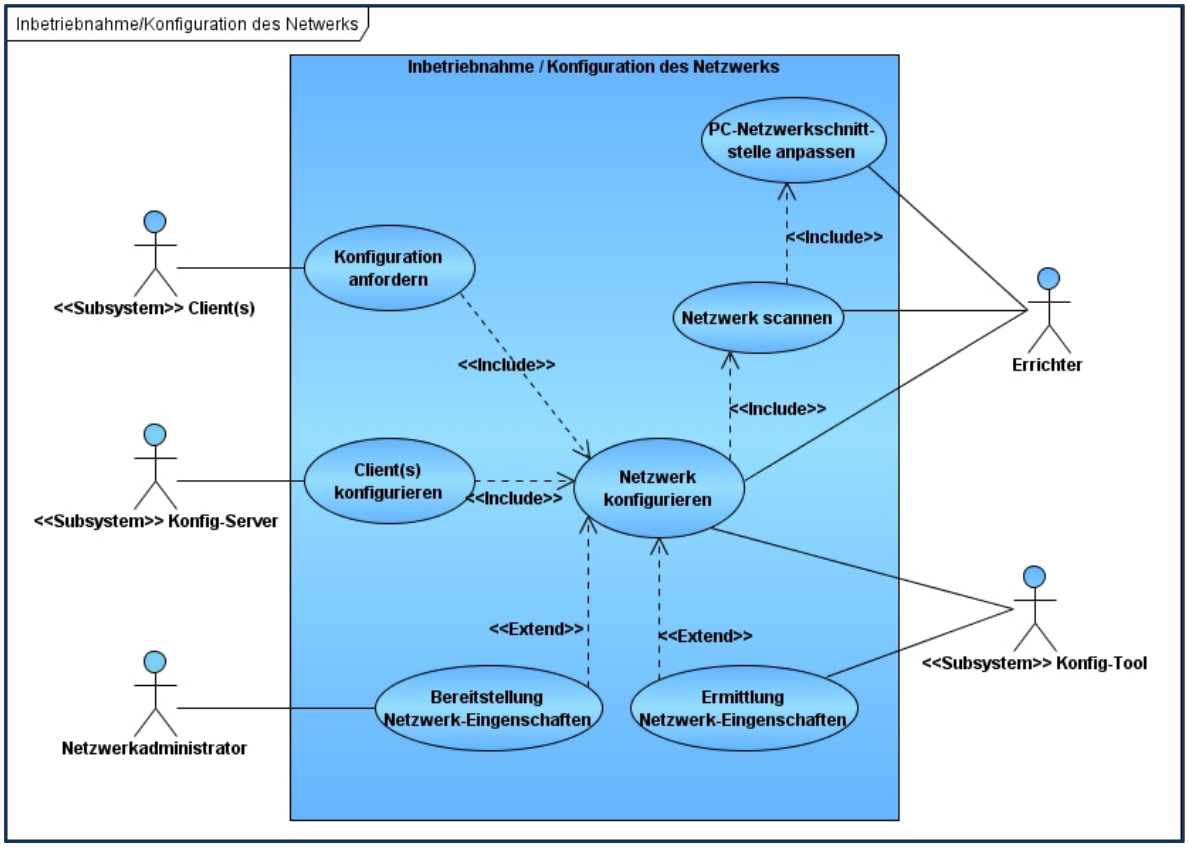


Abbildung 2-1: Use-Case Inbetriebnahme und Konfiguration des Netzwerks

Anwendungsfall „PC-Netzwerkschnittstelle anpassen“.

Name	PC-Netzwerkschnittstelle anpassen
Akteur	Errichter/Konfigurationstool
Auslösendes Ereignis	Errichter will das Netzwerk konfigurieren/in Betrieb nehmen.
Vorbedingungen	Ausgangszustand des Netzwerks wie im Abschnitt 2.1 beschrieben erfüllt.
Kurzbeschreibung	Die Netzwerkschnittstelle des PCs wird an das Subnetzwerk der zu konfigurierenden Geräte angepasst. Die erfolgt, indem sie mit einer IP-Adresse aus dem reservierten Bereich konfiguriert wird.
Essenzielle Schritte	1. Ausgangszustand des Netzwerks überprüfen. 2. PC ans Netzwerk anschließen. 3. Konfigurationstool starten. 4. PC-Netzwerkschnittstelle anpassen.
Nachbedingung	Die Netzwerkschnittstelle des PCs ist an das Subnetzwerk angepasst. Somit kann der PC durch das Konfigurationstool auf das Netzwerk bzw. die Geräte zugreifen.

Tabelle 2-1: Systemprozess Beschreibung: PC-Netzwerkschnittstelle anpassen.

Anwendungsfall „Netzwerk scannen“.

Name	Netzwerk scannen
Akteur	Konfigurationstool
Auslösendes Ereignis	Errichter klickt auf den Button Netzwerk-Scannen
Vorbedingungen	Prozess PC-Netzwerkschnittstelle anpassen durchgeführt.
Kurzbeschreibung	Nachdem der Errichte den Button Netzwerk-Scannen geklickt hat, wird das Netzwerk durch das Konfigurationstool gescannt. Ist der Scanvorgang abgeschlossen, werden alle gefundenen Geräte und deren aktuelle Konfigurationsparameter (MAC-Adresse,

	Seriennummer, IP-Adresse …) in einer angemessenen Form (Liste, Tabelle …) dargestellt.
Essenzielle Schritte	1. Errichter klickt auf den Button Netzwerk-Scannen. 2. Wartet, bis Scanvorgang beendet ist. 3. Überprüft, ob alle zu konfigurierenden Geräte geantwortet haben. Ist das der Fall, zu 4 übergehen, andernfalls zurück zu 1. 4. Netzwerk Konfigurieren [Tabelle 2-5].
Nachbedingung	Die aktuellen Konfigurationen der zu konfigurierenden Geräte liegen bereits vor.

Tabelle 2-2: Systemprozess Beschreibung: Netzwerk scannen.

Anwendungsfall „Netzwerkeigenschaften bereitstellen“.

Name	Netzwerkeigenschaften bereitstellen
Akteur	Netzwerkadministrator
Auslösendes Ereignis	Errichter braucht Informationen über das Netzwerk.
Kurzbeschreibung	Die wichtigste Netzwerkeigenschaft, die der Netzwerkadministrator zu Verfügung stellen muss, ist der für das System reservierte IP-Adressen-Bereich und die damit gebundenen Netzwerkeigenschaften. Außerdem muss er sicherstellen, dass keinem anderen Gerät im Netzwerk eine IP-Adresse aus diesem Bereich je zugewiesen wird.
Vorbedingungen	--
Essenzielle Schritte	--
Nachbedingung	Die vom Netzwerkadministrator geforderten Informationen über das Netzwerk liegen bereit.

Tabelle 2-3: Systemprozess Beschreibung: Netzwerkeigenschaften bereitstellen.

Anwendungsfall „Netzwerkeigenschaften ermitteln“.

Name	Netzwerkeigenschaften ermitteln
Akteur	Konfigurationstool
Auslösendes Ereignis	Betätigung des Buttons „Netzwerk-Eigenschaften ermitteln“
Vorbedingungen	Prozess PC-Netzwerkschnittstelle anpassen ist durchgeführt [Tabelle 2-1].
Kurzbeschreibung	Ziel dieses Systemprozesses ist, die Netzwerkeigenschaften bzw. Informationen über das Netzwerk automatisch zu ermitteln (z. B. Subnet-Mask, Default-Gateway, MTU …). Dies erfolgt durch Einsatz der Protokolle bzw. Methoden, die dafür geeignet sind (vgl. Abschnitt 3.9).
Essenzielle Schritte	--
Nachbedingung	Die zur Konfiguration des Netzwerks benötigten Eigenschaften liegen bereits vor.

Tabelle 2-4: Systemprozess Beschreibung: Netzwerkeigenschaften ermitteln.

Anwendungsfall „Netzwerk konfigurieren“.

Name	Netzwerk konfigurieren.
Akteur	Errichter/Konfigurationstool
Auslösendes Ereignis	--
Vorbedingungen	Netzwerk-Scannen [Tabelle 2-2] ist erfolgreich durchgeführt worden und alle notwendigen Informationen für die Konfiguration liegen bereits vor [Tabelle 2-3, Tabelle 2-4].
Kurzbeschreibung	Ist der Vorgang „Netzwerk scannen“ erfolgreich durchgeführt, stehen Informationen über alle berechtigten bzw. bekannten Geräte zur Verfügung, die der Zustand, die Art (Client, Server) sowie die aktuellen Konfigurationsparameter der Geräte sein könnten.

	Handelt es sich um eine Erstinbetriebnahme, sind dies Informationen der Werkseinstellung [vgl. Abschnitt 2.1]. Der Errichter kann sich zwischen Auto- Hybrid- oder einer manuellen Konfiguration entscheiden. Autokonfiguration heißt in diesem Kontext, dass die Netzwerkkonfiguration automatisch vom Konfigurationstool durchgeführt wird. Der Errichter überprüft dann die Autokonfiguration und passt sie nach Bedarf entsprechend an.
Essenzielle Schritte	1. Sicherstellen, dass „Netzwerk-Scannen“ erfolgreich durchgeführt worden ist. 2. Netzwerk konfigurieren. 3. Besteht ein Server, dann zu 6 andernfalls zu 4. 4. Auswahl des Servers. 5. Mitteilung der Wahl des Servers an alle Clients. 6. Mitteilung der Netzwerk-Konfiguration an den Server. 7. Mitteilung an die Clients, dass neue Konfigurationen beim Server zur Verfügung stehen.
Nachbedingung	Aktuelle Konfiguration des Netzwerks liegt beim Server bereits vor und die Clients sind darüber informiert.

Tabelle 2-5: Systemprozess Beschreibung: Netzwerk konfigurieren.

Anwendungsfall „Clients konfigurieren“.

Name	**Clients konfigurieren**
Akteur	Server
Auslösendes Ereignis	Server erhält Konfigurationsanforderungen von Client(s)
Vorbedingungen	Netzwerkkonfiguration liegt beim Server bereits vor.
Kurzbeschreibung	Nachdem der Server die Netzwerkkonfiguration von Konfigurationstool erhalten hat, speichert er diese lokal und wartet auf Konfigurationsanforderungen der Clients. Erhält er eine Anforderung, liest er die entsprechende Konfiguration und sendet sie dem Client.

Essenzielle Schritte	1. Server bekommt die aktuelle Netzwerkkonfiguration. 2. Server speichert die Netzwerkkonfiguration ab. 3. Server wartet auf die Anforderungen der Clients. 4. Empfängt der Server eine Client-Anforderung, liest er die entsprechenden Client-Konfigurationsparameter aus und sendet ihm diese.
Nachbedingung	Clients sind konfiguriert und über die neuen Konfigurationsdaten (z. B. IP-Adresse) kommunikationsfähig bzw. erreichbar.

Tabelle 2-6: Systemprozess Beschreibung: Clients konfigurieren

2.5 Funktionale und nicht funktionale Anforderungen

Anhand der Abbildung 2-1 und der im Abschnitt 2.4 beschriebenen Systemprozesse können die Komponenten, die Systembeteiligten (stakholder) sowie die funktionalen und nicht funktionalen Anforderungen des Systems hergeleitet werden. Dies wird in diesem Kapitel realisiert.

2.5.1 Systembeteiligte

Aus Abbildung 2-1 ist ersichtlich, dass zwei Personen an der Inbetriebnahme/Konfiguration des Netzwerks beteiligt sind, ein Netzwerkadministrator und ein Errichter. Der Netzwerkadministrator ist für die Bereitstellung des für die zu konfigurierenden Geräte reservierten IP-Bereichs (scope) sowie die damit gebundenen Netzwerkeigenschaften zuständig. Der Errichter ist für die Inbetriebnahme/Konfiguration des Netzwerks zuständig.

2.5.2 Komponenten des Systems

Die Komponenten des Systems werden durch Abbildung 2-2 illustriert. Daraus ist ersichtlich, dass das System aus vier Komponenten bestehen: Server- und Client, welche auf dem Gerät laufen, ein Konfigurationstool, welches auf einem PC (Notebook) läuft und eine Sicherheitskomponente, welche einen Sicherheitsdienst zur Überprüfung der Nachrichten bereitstellt. Diese ist sowohl auf dem PC als auch auf dem Gerät zu finden. Die Kommunikation zwischen den Komponenten des Systems wird durch die Assoziationspfeile in Abbildung 2-2 illustriert. Das Konfigurationstool kommuniziert mit allen Geräten, und die Client-Geräte kommunizieren mit den Server-Geräten.

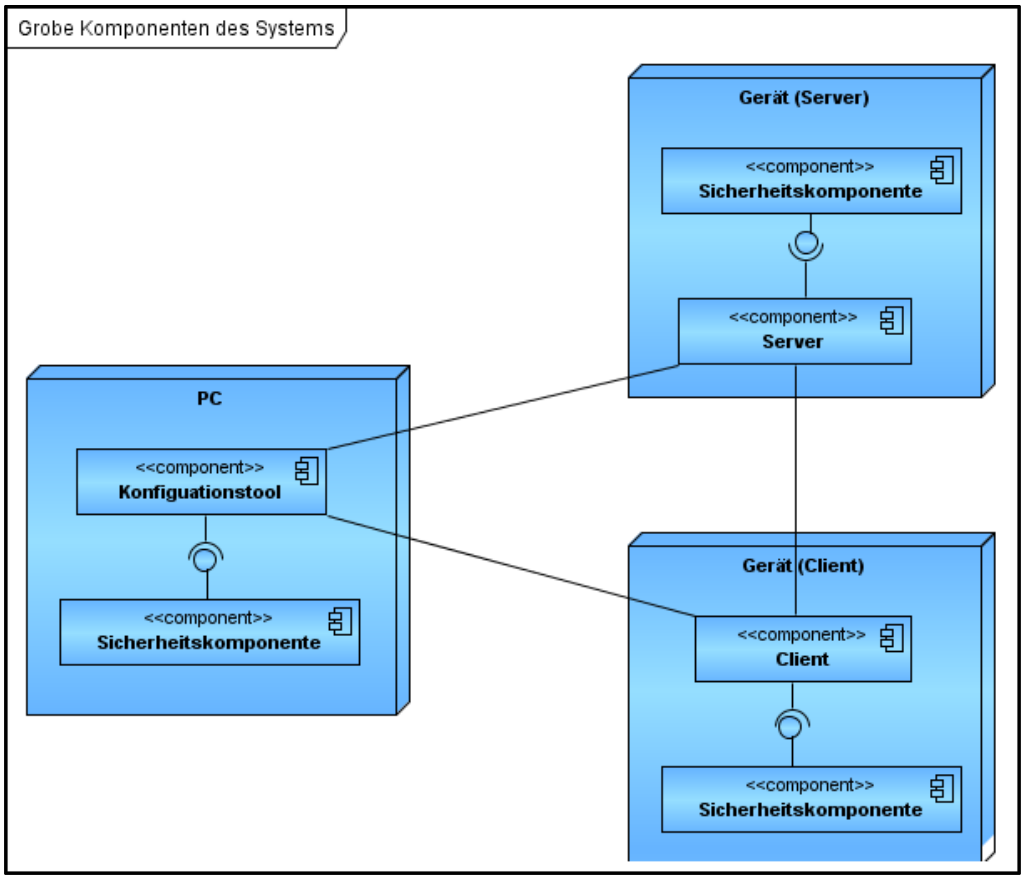


Abbildung 2-2: Komponenten des Systems

2.5.3 Funktionale Anforderungen

Die funktionalen Anforderungen der Komponenten des Systems werden in diesem Abschnitt festgelegt.

2.5.3.1 Konfigurationstool

Das Konfigurationstool ist das GUI-Tool, welches zum Zugriff auf das Netzwerk, zur Konfiguration des Netzwerks und zur Darstellung der Geräte dient.

Folgende funktionale Anforderungen sind für das Konfigurationstool festgelegt worden.

FAKt[1]1 **Geräte auffinden:** Es muss möglich sein, mit dem Konfigurationstool das Netzwerk zu scannen, um alle zu konfigurierenden Geräte aufzufinden. Dafür kann eine Broadcast-Nachricht an das Subnetzwerk gesendet werden, wo sich die Geräte befinden.

FAKt2 **Aktuelle Netzwerkkonfigurationsdaten vom Server fordern:** Wurde ein Server nach dem Scanvorgang gefunden, muss das Konfigurationstool die aktuelle Netzwerkkonfigurationsdaten von ihm fordern.

FAKt3 **Geräteausfall erkennen:** Ist ein Gerät (Client oder Server) ausgefallen, muss das Konfigurationstool dieses nach einem Scanvorgang erkennen und eine entsprechende Meldung anzeigen. Dann kann der Errichter eine Fehlersuche durchführen.

FAKt4 **Erkennung von inkonsistenten Konfigurationen:** Das Konfigurationstool muss die aktuellen Konfigurationsparameter aller Clients mit den entsprechenden Einträgen in der Netzwerkkonfigurationsdaten auf Konsistenz prüfen. Werden inkonsistente Konfigurationsparameter gefunden, müssen diese gemeldet bzw. dargestellt werden. Inkonsistente Konfigurationen müssen automatisch vom Konfigurationstool oder manuell vom Errichter beseitigt werden.

FAKt5 **Darstellung der Geräte:** Nach dem Scanvorgang müssen alle gefundenen Geräte nach dem Operationsmodus (Client, Server) und mit deren aktuellen Konfigurationsparametern (IP-Adresse, Subnetzwerk-Adresse, Default-Route, …) in angemessener Form (Liste, Tabelle, Tree) dargestellt werden. Ausgefallene Geräte, sowie Geräte mit inkonsistenten Konfigurationen werden getrennt dargestellt, unbekannte Geräte werden gefiltert.

FAKt6 **Ermittlung der allgemeinen Subnetzwerk-Eigenschaften:** Es muss möglich sein, über das Konfigurationstool die allgemeinen (für alle Geräte geltenden) Netzwerkeigenschaften (Default-Route, Subnetzwerk-Maske, MTU, etc.) durch entsprechende Protokolle (siehe Abschnitt 3.9) ermitteln zu können.

FAKt7 **Konfliktauflösung der IP-Adresse:** Bevor eine IP-Adresse für ein Gerät konfiguriert wird, muss sie auf einen Konflikt hin überprüft werden. Wird ein Konflikt hin erkannt, muss eine andere IP-Adresse aus dem reservierten Bereich ausgewählt werden und eine Fehlermeldung angezeigt werden.

FAKt8 **Automatische Konfiguration des Netzwerks:** Das Konfigurationstool muss das Netzwerk automatisch konfigurieren können. Nachdem der reservierte Bereich

[1] Funktionale Anforderung des Konfigurationstools

eingetragen und der Server ausgewählt wurde.

Bei einer automatischen Konfiguration sind folgende Aktionen durchzuführen:

- Automatisches Auffinden der Geräte (FAKt1)
- Automatische Ermittlung der allgemeinen Netzwerk Eigenschaften (FAKt6).
- Automatische Auswahl von gerätespezifischen Konfigurationsparametern.
- Automatische Zuweisung von IP-Adressen (statisch oder dynamisch) aus dem reservierten Bereich.

FAKt9 **Manuelle Konfiguration des Netzwerks:** Das Konfigurationstool muss Funktionen bzw. benutzerfreundliche Schnittstellen zur manuellen Konfiguration des Netzwerks bereitstellen.

FAKt10 **Mitteilung der Netzwerkkonfiguration an den Server:** Nachdem die Netzwerkkonfiguration abgeschlossen ist, muss das Konfigurationstool die Netzwerkkonfigurationsdaten an den Server senden. Der Server muss den Empfang mit einer entsprechenden Nachricht bestätigen.

FAKt11 **Server Bekanntgabe:** Das Konfigurationstool kann den Clients mitteilen, welcher von ihnen als Server ausgewählt wurde. Somit können die Clients beim Anfordern ihrer Konfigurationsparameter den Server direkt ansprechen (per unicast).

FAKt12 **Clients über neue Konfigurationsparameter informieren:** Wurde eine Änderung an den Netzwerkkonfigurationsdaten vorgenommen, muss das Konfigurationstool die Clients darüber informieren.

2.5.3.2 Konfigurationsserver

Der Konfigurationsserver ist eine zentrale Stelle zur permanenten Speicherung und Verwaltung der Netzwerkkonfigurationsdaten, und zur Verteilung von Clients-Konfigurationsparametern.

Folgende funktionale Anforderungen sind für den Konfigurationsserver festgelegt worden.

FAK[2]s1 **Permanente Speicherung der Netzwerk-Konfiguration:** Der Server muss die vom Konfigurationstool empfangenen Netzwerkkonfigurationsdaten permanent (in einer Konfigurationsdatei oder Datenbank) speichern. Die vom Server zur Laufzeit vorgenommenen Konfigurationen müssen auch gespeichert werden.

FAKs2 **Automatische Konfiguration der Clients:** Der Server muss nach Empfang einer Client-Anforderung die entsprechenden Client-Konfigurationsparameter aus dem Speicher lesen und an ihn versenden. Sind die Konfigurationsparameter des Clients noch nicht vorhanden, muss der Server entsprechende Konfigurationsparameter zusammenstellen.

FAKs3 **Überprüfung von Konflikten der IP-Adresse:** Bevor der Server einem Client eine IP-Adresse sendet, muss er überprüfen, ob diese von keinem anderen Gerät benutzt wird. Wird ein Konflikt erkannt, muss der Server entsprechend protokollieren und eine neue IP-Adresse für den Client auswählen.

FAKs4 **Verwaltung des reservierten Bereichs:** Der Konfigurationsserver ist zuständig für die Verwaltung des reservierten Bereichs. Es dürfen nur IP-Adressen aus diesem Bereich verteilt werden.

2.5.3.3 Client

Ein Client ist die Applikation, die auf einem Gerät läuft. Diese dient dazu, die Konfigurationsparameter des Geräts vom Konfigurationsserver zu fordern, um die Netzwerk-Schnittstelle des Geräts entsprechend zu konfigurieren.

Nachfolgende funktionale Anforderungen sind für den Client festgelegt worden.

FACl[3]1 **Anfordern der Netzwerkkonfiguration des Geräts:** Der Client muss die Netzwerkkonfigurationsparameter des Geräts vom Server bei jedem Neustart oder bei einer Änderung dieser anfordern. Er soll vom Konfigurationstool mitgeteilt bekommen, dass seine Konfigurationsparameter sich geändert haben.

FACl2 **Überprüfung von Konflikten der IP-Adresse:** Bevor der Client eine empfangene IP-Adresse einsetzt, muss er sie auf einen Konflikt hin überprüfen. Erkennt

[2] Funktionale Anforderung des Konfigurationsservers

[3] Funktionale Anforderung des Clients

er einen Konflikt, muss er es dem Server mitteilen, damit ihm eine neue IP-Adresse zugewiesen werden kann.

FACl3 **Konfiguration der Netzwerkschnittstelle des Geräts:** Erhält ein Client Konfigurationsparameter mit einer konfliktfreien IP-Adresse, muss die Netzwerkschnittstelle des Geräts entsprechend konfiguriert werden.

FACl4 **Permanente Speicherung der Konfigurationsparameter des Geräts:** Nach dem Einsatz neuer Konfigurationsparameter, müssen diese permanent gespeichert werden.

2.5.4 Nicht funktionale Anforderungen

Nicht funktionale Anforderungen werden in diesem Abschnitt festgelegt. Sie stellen die Umstände dar, unter denen die funktionalen Anforderungen erfüllt werden müssen.

Nachfolgende nicht funktionale Anforderungen sind für das System festgelegt worden.

NFA[4]1 **Einhaltung von Standard Protokollen:** Die Konzeption muss sich auf die Standard IP-Protokolle stützen.

NFA2 **Sicherheit:** Der Zugang zum System für eine Konfiguration muss gesichert sein. Jeglicher Nachrichtenaustausch zwischen den Komponenten des Systems muss authentifiziert sein. Die empfangenen Nachrichten müssen auf Konsistenz geprüft werden.

NFA3 **Nachrichten:** Es dürfen keine (Broadcast) Nachrichten gesendet werden, die von anderen (unbekannten) Geräten falsch interpretiert werden könnten.

NFA4 **Adressenkonflikt:** Es dürfen keine Adressen eingesetzt werden, die im Konflikt stehen.

[4] Nicht funktionale Anforderung

3 Untersuchte Protokolle

In Kapitel 2 wurden die Anforderungen an das System festgelegt. Zur Erfüllung einiger dieser Anforderungen, werden Protokolle benötigt. Es wurden zuerst alle untersucht, die in Fragen kommen könnten, zur (automatischen) Inbetriebnahme/Konfiguration des Netzwerks (DHCP, NETCONF, Zeroconfig, SNMP), zur automatischen Ermittlung der Netzwerkeigenschaften(ICMP), für die Sicherheit (DHCP Sicherheit) und zur Adressen-Konflikt Erkennung(ARP, Ping) eingesetzt zu werden. Sie werden in diesem Kapitel beschrieben. Welche davon tatsächlich eingesetzt werden, wird erst im Kapitel 4 festgelegt.

3.1 DHCPv4

DHCP (Dynamic Host Configuration Protocol) ist der Nachfolger von BOOTP (Bootstrap Protocol), das ursprünglich entwickelt wurde, um Rechner ohne Festplatte als Endsysteme in IP-Netzen zu starten und automatisch zu konfigurieren. Seine heutige Bedeutung liegt jedoch darin, Endsysteme bzw. Router, die beispielsweise via DSL (Digital Subscriber Line) mit dem Internet verbunden sind, mit offiziellen IP-Adressen und einigen Konfigurationsparametern auszustatten sowie Rechner in lokalen und internen Netzen (oft) mit privaten (nicht routingfähigen) IP-Adressen und einigen Konfigurationsparametern zu versehen (vgl. Badach 2007, S. 192), wie zum Beispiel mit einer Subnetzmaske, einem DNS-Domänennamen, einer DNS-Server-IP-Adresse, etc.

Mit DHCP lassen sich vor allem die mit der manuellen Konfiguration von IP-Adressen verbundenen Probleme beseitigen. Die erste Version von DHCP wurde bereits Ende 1993 im RFC 1541 veröffentlicht und als Standard im März 1997 durch den RFC 2131 mit einer neuen DHCP-Version abgelöst. Inzwischen wurde RFC 2131 um die RFCs 2132, 3396, 4039, 3203, 3118, etc. erweitert.

3.1.1 Das DHCP-Prinzip

DHCP funktioniert nach dem Client/Server Prinzip. Ein DHCP-Server ist eine Applikation in einem „Server-Rechner“, mit der die IP-Konfigurationsparameter vom „Client-Rechner“ zentral verwaltet und gespeichert werden, oft nur innerhalb eines Subnetzes. Ein DHCP-Client ist eine Applikation in einem „Client-Rechner“, mit der die Konfigurationsparameter des Rechners vom DHCP-Server gefordert und eingestellt werden. Zu Konfigurationsparametern zählen die IP-Adressen, die Subnetz-Maske, die Lease-Dauer, die Identifikation des DHCP-Servers, usw. Alle Konfigurationsparameter, bis auf die IP-Adresse, werden durch das „options field“ der DHCP-Nachricht übermittelt.

Die zu vergebenen IP-Adressen werden aus einem IP-Adressenbereich (Scope) entnommen, den der DHCP-Server verwaltet. Die Vergabe kann nach folgenden Regeln erfolgen:

- **Automatische Zuweisung:** Dem Rechner wird eine IP-Adresse zugewiesen, die dieser während seiner Laufzeit, das heißt vom Start der DHCP-Abfrage bis zum Herunterfahren der TCP/IP-Applikation behält.
- **Dynamische Zuweisung:** Die IP-Adresse bzw. die gesamte IP-Konfiguration wird für einen bestimmten Zeitraum, die Lease-Dauer, zugewiesen. Nach Ablauf der Lease-Dauer verfällt die Zuweisung der IP-Adresse für den Rechner. Der DHCP-Server kann die IP-Adresse nun an einen anderen Rechner vergeben.
- **Statische Zuweisung*:*** Hier führt der DHCP-Server eine Zuweisungstabelle der MAC Adresse des Rechners mit seiner IP-Adresse. Dies stellt sicher, dass der Rechner immer die gleiche IP-Adresse zugewiesen bekommt, und dass nur Rechner mit bekannten und registrierten MAC-Adressen mit IP-Adressen ausgestattet werden.

3.1.2 DHCP Optionen

Neben der IP-Adresse kann dem Client noch eine Reihe weiterer Konfigurationsparameter über DHCP übermittelt werden. Dies geschieht in Form von sog. DHCP-Optionen, welche in RFC 2132 definiert sind und im Optionensfeld einzutragen sind.

Die Semantik bei der Konfiguration von Optionen im DHCP-Server ist folgendermaßen zu verstehen:

Es gibt global gültige Optionen und Optionen, die sich auf einen bestimmten Scope beziehen. Globale Optionen gelten für alle bei dem jeweiligen DHCP-Server anfragenden Clients, sofern sie nicht von einem niedrigeren Option Level überschrieben werden. Scope Optionen hingegen gelten für einen speziellen reservierten Bereich.

Ist eine bestimmte Option auf mehreren Ebenen definiert, so gilt der in der Hierarchie weiter unten definierte Wert, d. h. Server-Optionen werden durch Scope-Optionen überschrieben, und Scope-Optionen werden durch Client-spezifische Optionen überschrieben.

Zusätzlich können auf jeder dieser drei Ebenen bestimmte Optionswerte innerhalb einer sog. Benutzerklasse festgelegt werden. Das bedeutet, dass anfragende Clients, die einer bestimmten Benutzerklasse angehören, die Optionswerte erhalten, die am DHCP-Server für diese jeweilige Benutzerklasse festgelegt wurden. Diese Benutzerklasse-Optionen werden jedoch auch wieder durch Client-spezifische Optionen überschrieben.

Einige Beispiele von DHCP-Optionen sind: Subnet-Mask, Hostname (Name des Rechners), Default-Gateway (Router IP Adresse), DNS-Server IP Adresse, DNS-Domänenname, Lease-Dauer (die Zeit, in der der Rechner die von dem DHCP-Server zugewiesene IP Adresse verwenden darf), DHCP-Server IP-Adresse, MTU (Maximum Transmission Unit) per Interface, die IP-Adresse der externen Server wie Time-, Mail-, Log-Server.

3.1.3 Aufbau von DHCP-Nachrichten

Zwischen einem DHCP-Client und einem DHCP-Server werden festgelegte DHCP-Nachrichten übermittelt, für die das verbindungslose Transportprotokoll UDP (User Datagramm Protokoll) eingesetzt wird. Der DHCP-Client ist über den „Well Known“ Port 68 zu erreichen und der DHCP-Server über den „Well Known“ Port 67.

Die folgenden Felder werden in DHCP-Nachrichten verwendet (vgl. Abbildung 3-1)

- **op** (1 Oktett), Operation: Angabe, ob es sich um eine Anforderung (Request) oder eine Antwort (Reply) handelt.
- **htype** (1 Oktett): Angabe des Netztyps gemäß RFC 1340 (z. B. 6 = IEEE 802.X-LAN).
- **hlen** (1 Oktett): Länge der Hardware-Adresse (6 = MAC-Adresse).
- **hops** (1 Oktett). Hier wird die Anzahl von Routern mit der DHCP-Relay-Funktion auf dem Datenpfad zwischen DHCP-Client und -Server angegeben.
- **xid** (4 Oktette), Transaktions-ID: Dies ist die Identifikation für die Transaktion zwischen dem Client und dem Server, um den DHCP-Clients im Server die Antworten zu den richtigen Anforderungen (Requests) zuordnen zu können.
- **secs** (Sekunden): Wird vom Client ausgefüllt und stellt in Sekunden die Zeit dar, die seit Beginn des Vorgangs abgelaufen ist.
- **Flags:** Das höchstwertige Bit dieses Feldes zeigt an, ob ein Client in der Lage ist, die IP-Pakete zu empfangen. Ist dies der Fall, verfügt der Client noch über eine gültige IP-Adresse. Die restlichen Bits dieses Feldes werden z.Z. nur auf 0 gesetzt und sind für zukünftige Zwecke reserviert.
- **ciaddr** (Client-IP-Adresse): Wird vom Client angegeben, falls er eine IP-Adresse besitzt.
- **yiaddr** (4 Oktette), Your-IP-Adresse: Hier wird die IP-Adresse eingetragen, die der Server dem Client zugewiesen hat.
- **siaddr** *(Server-IP-Adresse):* Hier wird die IP-Adresse des Servers angegeben (z. B. in der Nachricht DHCP_OFFER), die bei der nächsten Anforderung benutzt werden soll.
- **giaddr:** IP-Adresse des Routers mit der DHCP-Relay-Funktion.
- **chaddr:** Client-MAC-Adresse.

+	bit 0 - 7	8- 15	16-23	24-31
0	op	htype	hlen	hops
4	xid(4)			
8	secs(2)		flags(2)	
16	ciaddr(4)			
20	yiaddr(4)			
24	siaddr(4)			
28	giaddr(4)			
32	chaddr(16)			
48	sname(64)			
240	file(128)			
-	options(variable)			

Abbildung 3-1: Struktur von DHCP-Nachrichten [vgl. RFC 2131, S. 9]

- **sname** *(Server-Name,* optional): Der Server kann beim Senden einer DHCP_OFFER- oder DHCP_ACK-Nachricht seinen Namen in diesem Feld eintragen. Dieses Feld kann auch für options benutzt werden, die „option overload“ muss aber angewendet werden, um darauf hinzuweisen.
- **file** (optional): Der File-Name (z. B. Boot file) ist ein alphanumerischer String (Zeichenfolge). Diese Angabe ermöglicht es einem DHCP-Client, eine bestimmte Datei zu identifizieren, die er vom Server abrufen will. Der Server ist somit in der Lage, die richtige Datei auszuwählen und sie z. B. mittels des Protokolls FTP dem Client zukommen zu lassen.
- **options** (optional): Zusätzliche bzw. herstellerspezifische Konfigurationsparameter. Dieses Feld enthält die sog. DHCP-Optionen, die in RFC 2132 festgelegt sind.

3.1.4 Ablauf des DHCP-Verfahrens

Der Einsatz von DHCP zur automatischen Konfiguration von IP-Adressen bedeutet, dass der Benutzer eines Rechners keine IP-Adressen manuell eingeben muss. Der DHCP-Server stellt allen DHCP-Clients die erforderlichen Adressen zur Verfügung. Den Ablauf von DHCP bei der Zuweisung der IP-Adresse zu einem Rechner illustriert Abbildung 3-2. DHCP lässt mehrere DHCP-Server zu. Ein wichtiger Grund dafür ist die Serververfügbarkeit. Fällt ein Server aus, werden seine Funktionen automatisch von anderen Servern übernommen (siehe Abschnitt 3.2).

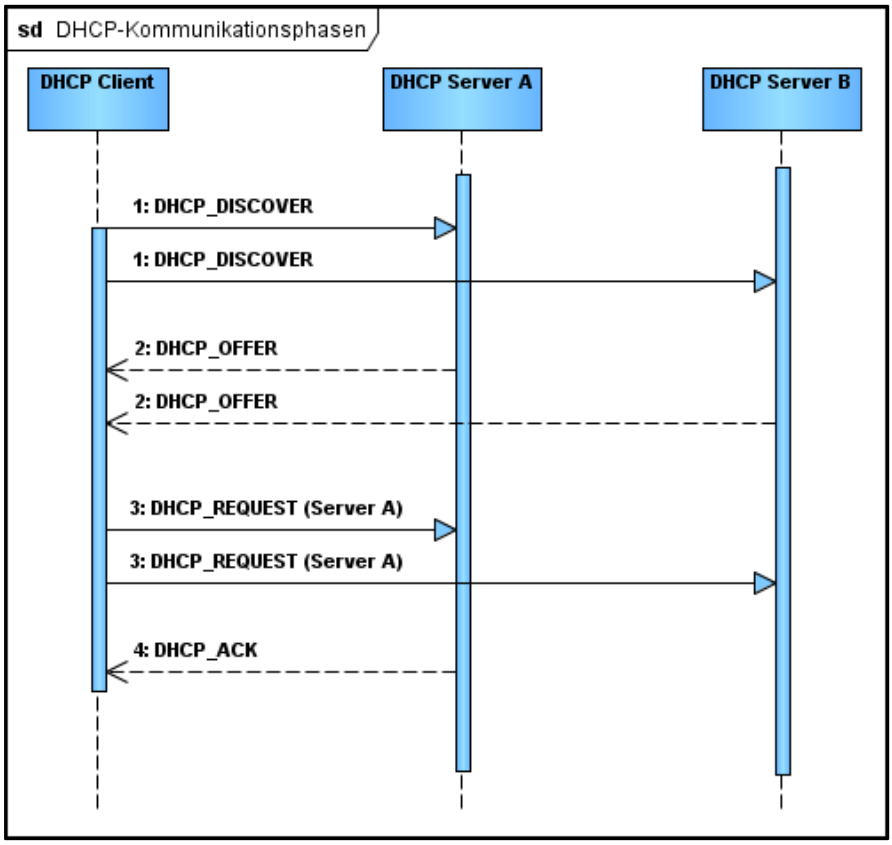


Abbildung 3-2: Kommunikationsphasen zwischen DHCP-Server und DHCP-Client

Um einem Rechner eine IP-Adresse zuzuweisen, sind vier Phasen nötig (Siehe Abbildung 3-2 und vgl. Badach 2007, S. 197):

1. Anforderungsphase (DHCP_DISCOVER)
Der Client sendet die Nachricht DHCP_DISCOVER in einem IP-Broadcast-Paket (Ziel-IP-Adresse = 255.255.255.255) als Anforderung, um von einem Server die benötigten Konfigurationsparameter (IP-Adresse, Subnet-Mask etc.) zu bekommen. Die Nachricht DHCP_DISCOVER als Broadcast wird normalerweise auf das eigene Subnetzwerk einge-

schränkt. Sie kann aber über eventuell vorhandene DHCP-Relay-Agenten in die anderen Subnetzwerke weitergeleitet werden. Der Einsatz von DHCP-Relay-Agenten hat dann eine große Bedeutung, wenn nicht alle Subnetzwerke über ihre eigenen DHCP-Server verfügen.

2. Angebotsphase (DHCP_OFFER)

Jeder DHCP-Server kann dem Client mit einer Nachricht DHCP_OFFER sein Angebot von Konfigurationsparametern zukommen lassen. Der Server versucht zuerst, dem Client direkt das Angebot zu senden. Aber dies ist nicht immer möglich. Hierbei sind zwei Fälle zu unterscheiden: Der Client wird gerade initialisiert und verfügt deshalb noch nicht über eine IP-Adresse. In diesem Fall sendet der Server sein Angebot als Broadcast-Nachricht (IP-Adresse 255.255.255.255). Diese Nachricht DHCP_OFFER enthält bereits die MAC-Adresse des betreffenden Clients, sodass nur der „richtige" Client diese Nachricht lesen darf. Der Client verfügt bereits über eine IP-Adresse, doch die Lease-Dauer geht zu Ende, sodass er diese Adresse auf die nächste Lease-Periode „verlängern" möchte. In diesem Fall wird das Angebot vom Server direkt an den Client gesendet.

3. Auswahlphase (DHCP_REQUEST)

In dieser Phase wählt der Client die Konfigurationsparameter des ersten von ihm empfangenen Angebots aus und sendet eine Broadcast-Nachricht DHCP_REQUEST, um das ausgewählte Angebot anzufordern. In DHCP_RESQUEST ist die Identifizierung des ausgewählten DHCP-Servers enthalten. Hier kann auch die angebotene IP-Adresse mithilfe der Option Requested IP-Address bestätigt werden. DHCP_REQUEST wird als Broadcast verschickt, um allen übrigen DHCP-Servern, die möglicherweise ihre Angebote für den Client reserviert hatten, mitteilen zu können, dass sich der Client für einen anderen Server entschieden hat. Die übrigen Server können die reservierten Parameter wieder freigeben, um sie anderen Clients anzubieten.

4. Bestätigungsphase (DHCP_ACK)

Der DHCP-Server, der vom Client ausgewählt wurde, antwortet mit der Nachricht DHCP_ACK, die alle Konfigurationsparameter für den Client enthält. Nach dem Empfang von DHCP_ACK und dem Eintragen von Parametern wird beim Client der Konfigurationsvorgang beendet. In dieser Phase können eventuell noch weitere, verspätete Angebote eintreffen. Sie werden nun vom Client einfach ignoriert.

3.1.5 Weitere DHCP-Nachrichten

- **DHCP_NAK:** Diese Nachricht wird in der Bestätigungsphase verwendet und von einem ausgewählten DHCP-Server an einen Client gesendet, um darauf hinzuweisen, dass die in DHCP_REQUEST geforderten Konfigurationsparameter abgelehnt wurden. Dies kann dann erfolgen, wenn ein Client versucht, die Lease-Dauer für seine

bisherige IP-Adresse zu verlängern, diese IP-Adresse aber nicht mehr verfügbar oder ungültig ist, weil der Client in ein anderes Subnetz „umgezogen" ist.

- **DHCP_RELEASE:** Mit ihr teilt ein DHCP-Client einem Server mit, dass einige Parameter (z. B. IP-Adresse) nicht mehr benötigt werden. Damit werden diese Parameter freigegeben und stehen anderen Clients zur Verfügung.
- **DHCP_DECLINE:** Damit teilt ein DHCP-Client dem Server mit, dass die vergebene IP-Adresse schon benutzt wird.
- **DHCP_INFORMATION:** Sie ist nur in der neuen Version von DHCP enthalten (RFC 2131). DHCP_INFORMATION kann ein Client nutzen, dem eine statische IP-Adresse manuell zugeteilt wurde, er jedoch dynamisch zusätzliche Konfigurationsparameter vom DHCP-Server zugeteilt bekommen möchte.
- **DHCP_FORCERENEW:** Nach RFC 2131 kann der Server den Client nicht ansprechen, wenn z. B. neue Konfigurationsdaten vorliegen. Daher wurde im RFC 3203 das FORCERENEW zu DHCP-Nachrichten hinzugefügt, um den Client in einen RENEW-Zustand zu versetzen, in dem er die neuen Konfigurationsparameter anfordern kann.
- **DHCP_DISCOVER (mit rapid commit option):** Die Anforderung der Konfigurationsparameter erfolgt normalerweise über vier Phasen (Abbildung 3-2).Es könnte aber sein, dass der Server dem Client schon bekannt ist. In diesem Fall ist es nicht erforderlich, über die vier Phasen zu kommunizieren. Dafür wurde im RFC 4039 eine neue DHCP-Option definiert. Diese ermöglicht dem Client, seine Konfigurationsparameter mit einer Zwei-Phasen-Kommunikation anzufordern. Dazu sendet er ein DHCP_DISCOVER mit einer „rapid commit Option“ und bekommt unmittelbar einen DHCP_ACK mit den geforderten Konfigurationen zurück.

3.1.6 DHCP-Relay-Agenten

Bei DHCP können auch sog. DHCP-Relay-Agenten implementiert werden. Ein solcher Agent hat die Aufgabe, DHCP-Nachrichten in andere Subnetze weiterzuleiten, die nicht über einen eigenen DHCP-Server verfügen. Ein Relay-Agent wird entweder in einen IP-Router oder in einen für diesen Zweck konfigurierten Rechner implementiert. Der Einsatz von Relay-Agenten hat den Vorteil, dass nicht für jedes Subnetz ein eigener DHCP-Server zur Verfügung gestellt werden muss. Andererseits besteht die Gefahr, dass beim Ausfall eines DHCP-Servers einige Clients nicht in der Lage sind, am Netzwerkbetrieb teilzunehmen. Es ist deshalb notwendig, immer sowohl redundante DHCP-Server als auch redundante DHCP-Relay-Agenten einzuplanen. Aus diesem Grund lässt DHCP mehrere DHCP-Server sowie mehrere Relay-Agenten zu.

3.1.7 Vor und Nachteile von DHCP

Ein laufender DHCP-Dienst bringt sowohl Vor- als auch Nachteile mit sich.

Vorteile

- DHCP vereinfacht die Konfiguration von großen und mittelgroßen Netzwerken. Wenn z. B. die DNS-Adresse oder eine andere Änderung im Netzwerk vorgenommen wurde, ist es für den Administrator nicht nötig, jeden Client manuell einzeln zu rekonfigurieren, da vorgenommene Änderungen oder Einstellungen in einer zentralen Stelle (DHCP-Server) durchgeführt und anschließend auf Anfrage der Clients gesendet werden. Das reduziert Fehler, die bei einer manuellen Konfiguration auftreten könnten.
- Die Anzahl der verschwendeten IP-Adresse wird wesentlich reduziert, da sie nur nach Bedarf zugewiesen werden (leasing).
- Dank der automatischen Konfiguration wird die „Plug-and-Play“-Fähigkeit der Rechner erreicht. Ein neu im Netzwerk angeschlossener Rechner meldet sich beim Server und bekommt seine Konfigurationsparameter automatisch zugewiesen. Ein Rechner, der das Netzwerk verlässt, gibt wieder seine Konfigurationsparameter (IP-Adresse) frei, sofern nicht statisch zugewiesen, damit sie von neu ankommenden Rechnern wiederverwendet werden können.
- Adressenkonflikte werden durch den Einsatz eines DHCP-Servers vermieden. Er überprüft, bevor er einem Client eine Adresse zuweist, ob diese aus dem reservierten Bereich noch nicht vergeben wurde und nicht von einem anderen Rechner zum Zeitpunkt der Zuweisung verwendet wird.

Nachteile

- DHCP kann ein „single point of failure” für das Netzwerk darstellen. Würde das Netzwerk nur aus einem DHCP-Server bestehen und dieser aus irgendeinem Grund nicht verfügbar sein, können neu im Netz ankommende Rechner keine Konfiguration erhalten, und eine abgelaufene Lease-Dauer könnte nicht mehr erneuert werden.
- DHCP-Nachrichten werden per Broadcast realisiert, und diese werden nicht geroutet. Besteht das Netzwerk aus mehreren Subnetzwerken, muss entweder jedes Subnetzwerk einen eigenen DHCP-Server haben, oder es muss im Router ein DHCP-Relay Agent installiert sein, der die Nachrichten weiterleitet. Das Verteilen von Broadcast-Nachrichten wird aber nicht empfohlen, da es einen „Broadcast-Sturm“ verursacht, der das Netz zur Überlastung führen kann.

- Eine DHCP-Transaktion wird meinst nicht authentifiziert durchgeführt, was zur Folge hat, dass „bösartige“ DHCP-Server falsche Konfigurationsparameter an Clients senden können.

Einige der oben genannten Nachteile werden von den Protokollen Authentication for DHCP Messages [RFC 3118] und DHCP-Failover beseitigt.

3.2 Planung redundanter DHCP-Server

Aus Gründen der Ausfallsicherheit empfiehlt es sich, mehrere DHCP-Server in einem Subnetz zu planen. Aufgrund der Art und Weise, wie DHCP-Leases vergeben werden, stellt das gleichzeitige Betreiben mehrerer DHCP-Server kein Problem dar, solange die Server richtig konfiguriert werden. Es ist darauf zu achten, dass sich der IP-Adressen-Bereich auf einem Server nicht mit dem konfigurierten IP-Adressen-Bereich auf einem anderen, redundant vorhandenen Server überlappt. Das ist notwendig, da die DHCP-Server nicht miteinander kommunizieren und somit auch nicht über die Konfiguration eines anderen Servers kennen. Bei Überlappungen könnte es dazu kommen, dass eine bestimmte IP-Adresse mehrmals in einem Subnetz vergeben wird.

Will man aber, dass ein IP-Bereich von zwei Servern verwaltet wird, ist ein Server als primär und der andere als sekundär zu definieren. Die Server müssen dann miteinander synchronisiert werden. Der primäre Server muss den sekundären Server über alle vorgenommenen Aktivitäten informieren (Lease Vergabe, Freigabe, …), damit er seine Konfigurationsdatei bzw. Datenbank aktualisieren kann. Somit kann der sekundäre Server im Falle des Ausfalls des primären Servers die Konfiguration im Netzwerk übernehmen, ohne dass inkonsistente Zustände auftreten (vgl. draft-ietf-dhc-failover-12)

3.3 Authentifizierung von DHCPv4-Nachrichten

DHCP spielt eine sehr wichtige Rolle im Betrieb moderner Netzwerke, bringt jedoch viele Sicherheitslücken mit sich. Die Sicherheitslücken des DHCP-Protokolls und eine mögliche Lösung werden in diesem Abschnitt dargestellt.

3.3.1 Mögliche DHCP Sicherheitslücke

Ein versehentlich falsch konfigurierter DHCP-Server stellt eine Bedrohung für die DHCP-Clients dar, da er die falsche Konfigurationen an den Clients weitergibt.

Ein Angriff auf die DHCP-Clients ist die Einrichtung eines „böswilligen“ DHCP-Servers (Rogue DHCP-Server), welcher falsche Konfigurationen an Clients verteilt. Mit einem solchen Server könnte man die Angriffe „man in the middle“- oder „denial of service“ erreichen.

Eine typische Bedrohung eines DHCP-Servers ist der Dienstdiebstahl (theft of service) durch einen „böswilligen“ DHCP-Client, der sich hinter gültigen Identitäten versteckt. Dabei reserviert er alle Adressen der Server-Konfiguration bzw. des IP-Bereichs.

Um diese Bedrohungen zu vermeiden, wurde eine neue DHCP-Option im RFC 3118 definiert (Erweiterung von RFC 2131). Ihr Einsatz wird nachfolgend beschrieben.

3.3.2 Aufbau der DHCP-Authentifizierungsoption

Der Aufbau der DHCP-Authentifizierungsoption wird durch Abbildung 3-3 dargestellt. Beim Einsatz dieser Option werden nachfolgende Felder benutzt.

- **Code:** Code für die Option, er hat den Wert 90.
- **Length:** Länge im Oktett der Option, also die Länge der restlichen Felder, Protocol, Algorithm, RDM, Replay Detection und Authentication Information.
- **Protocol:** Definiert die benutzte Authentifizierungstechnik.
- **Algorithm:** Algorithmus, der von der Authentifizierungstechnik benutzt wird.
- **RDM** (Replay Detection Method)*:* Erkennungsmethode für Replay-Angriffe. Wird eingesetzt, um zu vermeiden, dass Nachrichten abgehört und wieder gesendet werden.
- **Replay cont.:** Replay-Angriff Erkennungsmethode Inhalt.
- **Athentification Information***:* Authentifizierungsinformation, sie hängt von der benützten Authentifizierungstechnik ab.

<table>
<tr><th>bits 0 - 7</th><th>8 - 15</th><th>16 - 23</th><th>24 - 31</th></tr>
<tr><td>Code</td><td>Length</td><td>Protocol</td><td>Algorithm</td></tr>
<tr><td>RDM</td><td colspan="3">Replay Detection (64 bits)</td></tr>
<tr><td colspan="4">Replay cont.</td></tr>
<tr><td>Replay cont.</td><td colspan="3">Authentication Information</td></tr>
<tr><td colspan="4">Authentication Information</td></tr>
</table>

Abbildung 3-3: Aufbau der DHCP-Authentifizierungsoption [vgl. RFC 3118, S.3]

Zwei Authentifizierungstechniken werden nachfolgend beschrieben.

3.3.3 Authentifizierung durch ein gemeinsam benütztes Token

Die Authentifizierung durch ein gemeinsam benütztes Token erfolgt durch ein im Klartext ausgetauschtes Token zwischen Client und Server, welches ein 8-128 bit Schlüssel ist. Das ist eine sehr „rudimentäre“ Authentifizierungsmethode und reicht nicht zur Absicherung gegen Hacker Angriffe. Deshalb sollten „fortgeschrittene Methoden“ wie das „Delayed Authentication“ angewendet werden.

3.3.4 Delayed Authentification (verzögerte Authentifizierung)

Die verzögerte Authentifizierungstechnik basiert auf dem HMAC (Keyed-Hashing for Message Authentication) Protokoll RFC 2104, welches MD5 (Message-Digest Algorithm) [RFC 1321] als Hash-Funktion einsetzt. Hierbei wird das Protocol-Feld auf 1 gesetzt. Der Client fordert bei einem DHCP_DISCOVER oder DHCP_INFORMATION eine Authentifizierung an (verzögert). Der Server antwortet mit einer DHCP_OFFER-Nachricht, welche die vom Client geforderte Authentifizierung enthält. Ist sie gültig, kann mit dem Nachrichtenaustausch fortgefahren werden andernfalls wird unterbrochen. War sie gültig, wird sie für alle weiteren Nachrichten während der Sitzung verwendet. Wird eine Nachricht als nicht authentifiziert gekennzeichnet, wird die Sitzung ungültig. Die Überprüfung der Authentifizierung ist von beiden Seiten durchzuführen.

Die folgenden Begriffe werden im Rahmen einer verzögerten Authentifizierung verwendet.

- **Replay Detection** (Erkennungsmethode eines Replay-Angriff): Wird eingesetzt, um zu vermeiden, dass Nachrichten abgehört und wieder gesendet werden.
- **K (Key):** Gemeinsam benutzter geheimer Schlüssel zwischen Sender und Empfänger.
- **secret ID:** Eindeutige Identifizierung des geheimen Schlüssels.
- **HMAC-MD5:** Funktion zur Generierung des MAC (Message Authentification Code)

Das Input der HMAC-MD5 Funktion ist die gesamte DHCP-Nachricht, einschließlich der Nachrichten-Header und des Optionen-Feldes. Jeder DHCP-Client hat einen eindeutigen Key. Der Key wird immer durch seine ID ermittelt. Die ID wird im Feld Secret ID gespeichert.

Den Aufbau der DHCP_DISCOVER- oder DHCP_INFORMATION-Authentifizierungsanfrage zeigt Abbildung 3-4.

bits 0 - 7	8 - 15	16 - 23	24 - 31
Code	Length	1	Algorithm
RDM	Replay Detection (64 bits)		
Replay cont.			
Replay cont.			

Abbildung 3-4: Aufbau DHCP_DISCOVER/DHCP_INFORMATION Authentifizierungsanfrage [vgl. RFC 3118, S.6]

Den Aufbau einer DHCP_OFFER-, DHCP_REQUEST- oder DHCP_ACK-Nachricht bei einer Verzögerten Authentifizierung zeigt Abbildung 3-5.

bits 0 - 7	8 - 15	16 - 23	24 - 31
Code	Length	Protocol	Algorithm
RDM	Replay Detection (64 bits)		
Replay cont.			
Replay cont.	Secret ID (32 bits)		
secret id cont	HMAC-MD5 (128 bits)....		

Abbildung 3-5: Verzögerte Authentifizierung Nachrichtaufbau (DHCP-OFFER, -REQUEST oder –ACK) [vgl. RFC 3118, S.6]

3.3.5 Schwachstelle der DHCP-Authentifizierung

Die Gefahr wird durch die Authentifizierung nicht komplett beseitigt. Ein „böswilliger“ Client kann den Server immer noch mit DHCP_DISCOVER-Nachrichten überfluten, was zu einem „denial of service“-Angriff führen kann.

3.4 Zeroconfig (IPv4)

Damit ein Rechner in einem Netzwerk kommunizieren kann, muss er mindestens mit einer IP-Adresse und weiteren Konfigurationsparametern konfiguriert werden. Dies erfolgt entweder manuell durch den Benutzer oder dynamisch bzw. automatisch durch einen DHCP-Server. Es ist immer noch vorteilhaft, wenn sich der Rechner automatisch konfigurieren kann, auch wenn der DHCP-Dienst nicht zur Verfügung steht. Dies ermöglicht das Protokoll Zeroconfig (Zero Configuration Networking), welches in RFC 3927 definiert ist. Zeroconfig ist ein Protokoll, das auf ARP (Abschnitt 3.7) aufbaut und einem Rechner ermöglicht, sich selbst eine IP-Adresse aus einem reservierten Bereich zu geben. Diese "Link-Local-IP-Adressen" im Adressbereich 169.254.0.0/16 wurden von der IANA (Internet Assigned Numbers Authority) zu diesem Zweck in RFC 3330 definiert.

Prozedur bei der Auswahl einer Link-Local-IP-Adresse

Wenn ein Rechner sich eine Link-Local-IP-Adresse zuweisen will, wählt er mithilfe eines Zufallszahlengenerators eine IP-Adresse zwischen 169.254.1.0 und 169.254.254.255 aus. Die ersten 256 und die letzten 256 Adressen sind von der IANA für zukünftige Anwendungen reserviert.

Nach der Auswahl einer geeigneten IP-Adresse muss der Rechner unbedingt überprüfen, ob diese nicht schon von einem anderen Rechner verwendet wird, um einen Adressenkonflikt im Netz zu vermeiden. Dies erfolgt mit Hilfe von ARP Probe (siehe Abschnitt 3.7.1 ARP Probe).

Sobald der Rechner bereit ist, mit der Konfliktüberprüfung zu beginnen, wartet er eine zufällige Zeit zwischen 1 und 2 Sekunden ab, sendet dann drei ARP-Proben mit einem zufälligen Zeitabstand von 1 bis 2 Sekunden. Empfängt er zwischen dem Testanfang und zwei Sekunden nach Versenden der letzten ARP Probe ein ARP-Paket, bei dem die Absender-IP-Adresse der zu überprüfenden IP-Adresse entspricht, so wurde ein Konflikt gefunden. Er muss diese Prozedur mit einer anderen zufällig generierten IP-Adresse wiederholen.

Wird in diesem Zeitraum eine andere ARP-Probe empfangen, die als Empfänger-IP-Adresse die zu testende IP-Adresse enthält, und deren Absender-MAC-Adresse keiner der MAC-Adressen der Netzwerkschnittstelle des Rechners entspricht, so muss vom Rechner ebenfalls eine neue IP-Adresse generiert und überprüft werden. Das kann beispielsweise passieren, wenn zwei oder mehrere Rechner gleichzeitig probieren, dieselbe Link-Local-Adresse zu konfigurieren.

Um ARP-Stürme bei mehreren, kurz aufeinander folgenden Konflikten und damit eine Überlastung des lokalen Netzwerks zu vermeiden, muss jeder Rechner nach zehn Fehlversuchen die Geschwindigkeit, mit der er neue Adressen auswählt und überprüft, auf maximal eine Überprüfung pro Minute reduzieren. Stellt der Rechner keinen Konflikt fest, so hat er erfolgreich die generierte IP-Adresse für sich reserviert.

Ein Adressenkonflikt kann auch noch nach dem Einsatz der ausgewählten IP-Adresse stattfinden, wenn nämlich der Rechner ein ARP-Paket empfängt, das von einem anderen Rechner mit gültiger Sender- und Empfänger-IP-Adresse versendet wurde, die aber der eigene IP-Adresse entspricht. Der Rechner hat nun die Möglichkeit, entweder eine neue IP-Adresse auszuwählen oder seine IP-Adresse zu verteidigen.

3.5 NETCONF

Netconf [RFC 4741] ist ein Standard Protokoll zur Konfiguration und zum Management von Netzwerkelementen. Damit können Konfigurationsdaten, Statusinformationen und Systemnachrichten empfangen und versendet werden. Es bietet Geräten eine formale Programmierschnittstelle (API) an, welche zum Senden und Empfangen von Netconf-Nachrichten angesprochen wird.

Es wird ein in XML (Extensible Markup Language) eingebetteter RPC (Remote Procedure Call) benutzt, der im Idealfall über eine verschlüsselte Verbindung von Server zu Client bzw. von Client zu Server gesendet wird. Die jeweilige Antwort ist ebenfalls in XML eingebettet. Die komplette Syntax wird im RFC 4741 festgelegt, sodass alle beteiligten Anwendungen und Geräte problemlos auf die Daten zugreifen können.

Das Ziel von Netconf ist es, eine flexible und automatisierbare Schnittstelle anzubieten, die durch klare Richtlinien spezifiziert ist. In Netconf werden XML basierende Technologien angewendet, um den Zugriff auf verschiedene Netzwerktopologien, Netzwerkkomponenten und Rechner zu gewährleisten.

Netconf benutzt einen RPC-basierten Mechanismus, um die Kommunikation zwischen Client und Server zu vereinfachen.

Netconf wird in 4 Schichten eingeteilt. Diese werden in Abbildung 3-6 veranschaulicht.

(1) **Transport-Layer** wickelt die Kommunikation zwischen Client und Server ab. Die Verbindung sollte möglichst verschlüsselt sein.
(2) **RPC-Layer** stellt einen transportunabhängigen Mechanismus für einen „Remote Procedure Call“ bereit.
(3) **Operations-Layer** definiert eine Basis von Operationen mit XML codierten Parametern.
(4) **Content-Layer** enthält die spezifischen Konfigurationsanfragen und -antworten. Sie werden teilweise durch Schicht 3 organisiert und durch XML strukturiert.

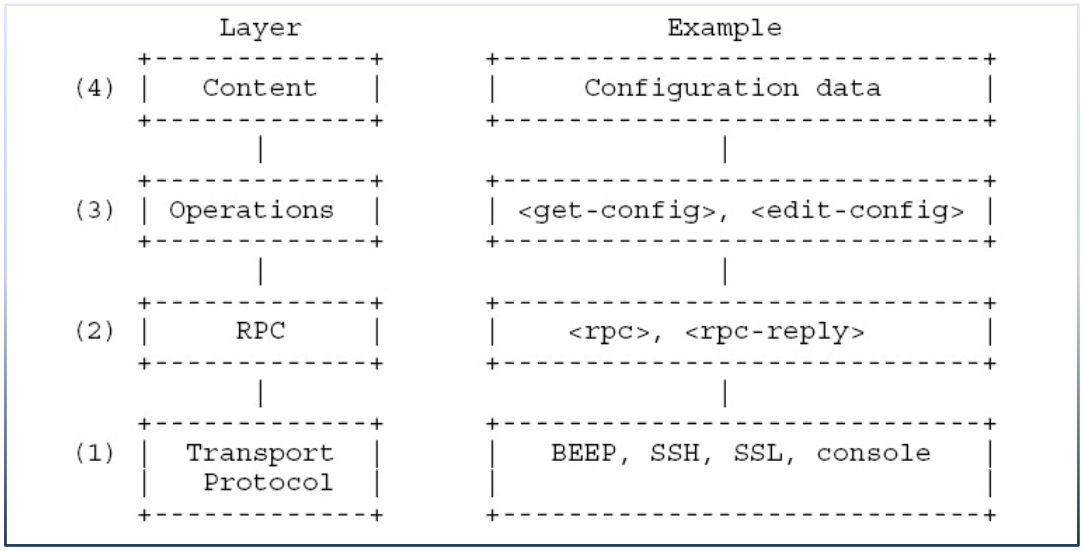
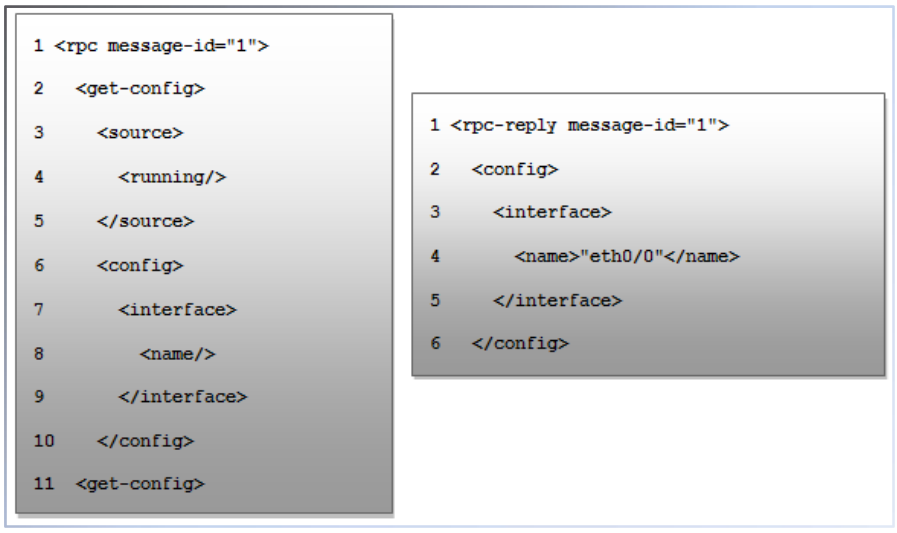


Abbildung 3-6: Netconf Schichten Modell [RFC 4741, S. 6]

Auswahl Netconf Operationen

Operationen werden in Netconf verwendet, um zu definieren, welche Art von Konfiguration vorgenommen werden soll.

get-config (Abbildung 3-7**):** Die `<get-config>` Operation wird verwendet, um die Konfiguration bzw. Teile der Konfiguration eines Gerätes anzufordern.



```
1 <rpc message-id="1">
2   <get-config>
3     <source>
4       <running/>
5     </source>
6     <config>
7       <interface>
8         <name/>
9       </interface>
10    </config>
11  <get-config>
```

```
1 <rpc-reply message-id="1">
2   <config>
3     <interface>
4       <name>"eth0/0"</name>
5     </interface>
6   </config>
```

Abbildung 3-7: Beispiel einer get-config Operation (vgl. Batroff 2007, S. 27)

Get-config unterscheidet dabei zwischen der aktuellen Konfiguration und der Startkonfiguration. Dies wird durch das Element <source> definiert, in dem man ein Kinderelement <running/> oder <startup/> definiert. Zusätzlich können Filter belegt werden, die spezielle Konfigurationsdaten auslesen können. Zum Beispiel die Interface- oder die Benutzerkonfigurationen. Wenn eine Anfrage positiv verlaufen ist, werden innerhalb eines <rpc-reply> Tags die angeforderten Daten gesendet. Bei einem Fehler wird innerhalb des <rpc-reply> ein <rpc-error> Tag mit einer Fehlerbeschreibung gesendet. Ein kurzes Beispiel einer <get-config> Anfrage auf den Namen eines Interfaces aus der „running" Konfiguration und eine mögliche Antwort wird im Abbildung 3-7 dargestellt.

edit-config (Abbildung 3-8**):** <edit-config> lädt einen Teil oder eine komplette Konfiguration auf ein angegebenes Gerät. Dabei kann eine Konfigurationsdatei angegeben werden, die verändert werden soll. Es ist auch möglich, eine URL (Uniform Resource Locator) anzugeben, in der sich die Konfiguration befindet. Ein Beispiel für das Ändern eines Routers von 10.0.0.1 auf 10.0.1.1 wird in Abbildung 3-8 gezeigt.

```
1 <rpc message-id="2">
2   <edit-config>
3     <target>
4       <running/>
5     </target>
6     <config>
7       <route>
8         <destinationtarget = "10.0.0.1">10.0.1.1</destination>
9      </route>
10   </config>
11   <edit-config>
12 </rpc>
```

```
1 <rpc-reply message-id="2">
2    <ok/>
3 </rpc-reply>
```

Abbildung 3-8: Beispiel einer edit-config Operation (vgl. Batroff 2007, S. 28)

Weitere Netconf Operationen wie <get>, <copy-config>, <delete-config>, <lock>, <unlock>, <close-session>, <kill-session> sind im RFC 4741 zu finden.

3.6 SNMP

Das Simple Network Management Protocol (SNMP) ist ein von der IETF-Organisation entwickeltes Protokoll zur zentralen Steuerung und Überwachung von Netzwerkelementen wie z. B. Router, Server, Switches, Drucker, Computer, usw. Es werden auf den Geräten sogenannte Agenten benutzt. Diese erfassen den Zustand des Gerätes und können teilweise Einstellungen vornehmen oder Aktionen auslösen (Abbildung 3-9). Die zentrale Managementstation (Manager) kommuniziert mit den Agenten über das Netzwerk. Der grundsätzliche Aufbau von SNMP wird in Abbildung 3-9 verdeutlicht.

SNMPv3 ist die aktuelle Version des Protokolls. SNMPv3 geht auf die Sicherheitslücken seiner Vorgänger SNMPv1 und SNMPv2 ein.

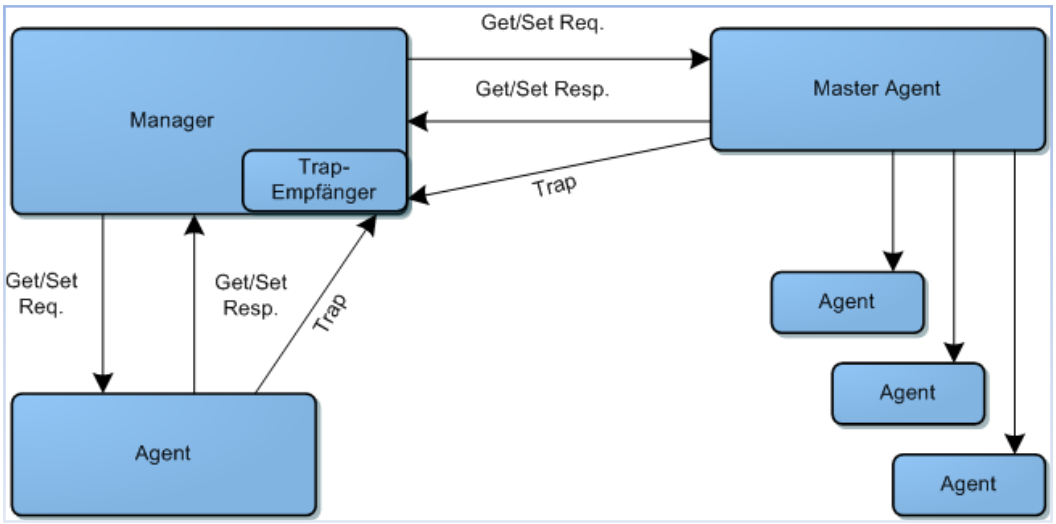


Abbildung 3-9: SNMP Aufbau.

Es gibt sechs verschiedene Pakete, die zum Management oder zur Konfiguration gesendet werden können:

- **GET** fordert ein Managementdatensatz an.
- **GETNEXT** fordert den nachfolgenden Datensatz an.
- **GETBULK** ruft mehrere Datensätze auf einmal ab, wie z. B. mehrere Reihen einer Tabelle (verfügbar ab SNMPv2).
- **SET** verändert einen oder mehrere Datensätze.
- **RESPONSE** antwortet auf eine Anfrage.
- **TRAP** sendet eine unaufgeforderte Nachricht eines Agenten an den Manager.

Management Information Base

Die Werte, die der Manager abruft oder auf die der Agent zugreift, werden in der Management Information Base (MIB) gespeichert. Diese ist in Form einer Baumstruktur aufgebaut, deren Elemente entweder durch eine Zahlennotation oder durch einen DNS-ähnlichen Namensaufbau adressiert werden. Zum Beispiel ist die IP-Gruppe der MIB-II sowohl über „1.3.6.1.2.1.4“, als auch unter dem Namen „iso.org.dod.internet.mgmt.MIB-2.ip“ erreichbar (Abbildung 3-10). Neben MIB-2 [RFC 1213] gibt es noch eine Reihe RFCs von MIBs, die speziell auf die entsprechenden Geräte angepasst sind. Zusätzlich können noch bei der IANA private MIBs reserviert werden, die dann als Private Enterprise Number unter „1.3.6.1.4.1“ (iso.org.dod.internet.privat.enterprise) zu finden sind. Diese Adressierung durch die Zeichenkette aus Zahlen und Punkten wird als Object Identifier (OID) bezeichnet und identifiziert eindeutig die Elemente innerhalb eines MIBs. Die Namen werden auf diese OIDs abgebildet.

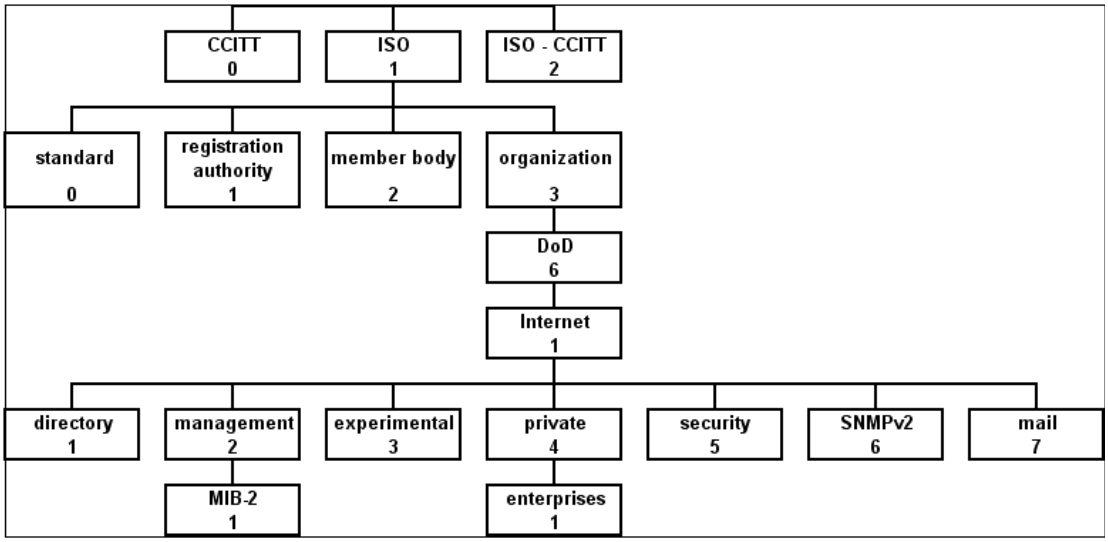


Abbildung 3-10: SNMP MIB-II Tree (Wikipedia1).

3.7 ARP

ARP [RFC 826] ist ein Hilfsprotokoll zur Ermittlung einer physikalischen Adresse (MAC-Adresse) für das Protokoll IP, d. h., es ist für die Zuordnung von MAC-Adressen zu IP-Adressen verantwortlich.

Wenn das Protokoll IP die Anforderung erhält, ein Paket an eine IP-Adresse im gleichen Subnetz zu senden, sucht es zuerst im ARP-Cache nach der korrespondierenden MAC-Adresse. Falls kein Eintrag vorhanden ist, wird versucht, mithilfe von ARP die gesuchte MAC-Adresse zu ermitteln. Hierfür wird ein ARP-Request verschickt (Broadcast). In dieser

ARP-Nachricht werden die restlichen Endsysteme im selben Subnetz gebeten, die gesuchte Adresszuordnung IP-Adresse der MAC-Adresse zukommen zu lassen. Ein Endsystem schickt immer eine Antwort als ARP-Reply (Unicast) mit der gesuchten Zuordnung zurück.

Die Ermittlung einer MAC-Adresse im Endsystem A nach ARP zeigt Abbildung 3-11. Die Broadcast-Nachricht ARP-Request enthält die IP-Adresse der angeforderten MAC-Adresse und wird von allen Endsystemen im LAN gelesen. Sobald ein Endsystem die eigene IP-Adresse im ARP-Request erkennt (hier Endsystem B), antwortet es mit einem ARP-Reply.

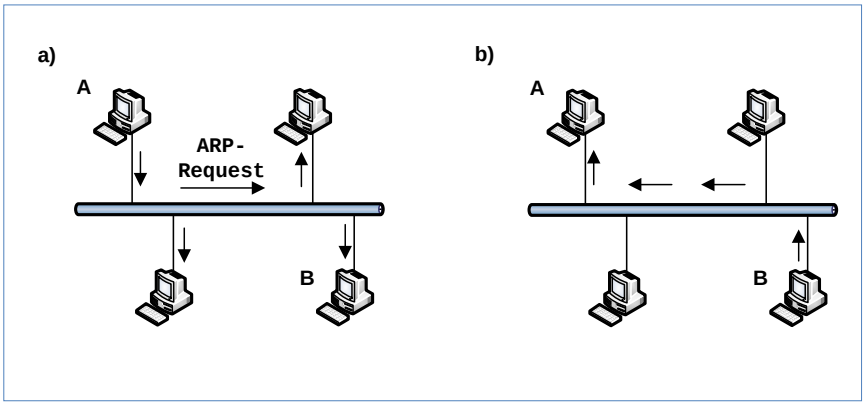


Abbildung 3-11: Ermittlung einer MAC-Adresse nach dem ARP: a) Broadcast-Nachricht ARP-Request, b) Antwort ARP-Reply (vgl. Badach 2007, S. 87)

Die ARP-Nachrichten haben folgende Felder (Abbildung 3-12):

- **Hardware Typ (HTYPE)** gibt an, von welchem LAN-Typ (z. B. Ethernet) die Nachricht generiert wurde.
- **Protocol Typ (PTYPE)** gibt an, von welchem Netzwerkprotokoll die Operation angefordert wurde. Das Protokoll IP hat den Wert 0x0800 (2048). ARP unterstützt auch andere Netzwerkprotokolle.
- **Hardware length (HLEN)** gibt die Länge der Hardwareadresse (d. h. MAC-Adresse) in Oktetten an. Normalerweise ist HLEN = 6 bei einer MAC-Adresse.
- **Protocol length (PLEN)** gibt die Länge der Protokolladresse, d. h. der IP-Adresse, in Oktetten an. Bei der IP-Adresse ist PLEN = 4.
- **Operation (OPER)** enthält die Zahl, die angibt, welche Operation ausgeführt werden soll (1 für ARP-Anforderung, 2 für ARP-Antwort, 3 für RARP-Anforderung, 4 für RARP-Antwort).

- **Sender hardware address (SHA)** enthält die MAC-Adresse des Absenders.
- **Sender protocol address (SPA)** enthält die IP-Adresse des Absenders.
- **Target hardware address (THA)** enthält die gesuchte MAC-Adresse (in ARP-Reply).
- **Target protocol address (TPA)** enthält die IP-Adresse, für die die MAC-Adresse ermittelt wird.

+	bits 0 - 7	8 - 15	16 - 31
0	Hardware type (HTYPE)		Protocol type (PTYPE)
4	Hardware length (HLEN)	Protocol length (PLEN)	Operation (OPER)
8	Sender hardware address (SHA) (first 32 bits)		
12	Sender hardware address (SHA) (last 16 bits)		Sender protocol address (SPA) (first 16 bits)
16	Sender protocol address (SPA) (last 16 bits)		Target hardware address (THA) (first 16 bits)
20	Target hardware address (THA) (last 32 bits)		
24	Target protocol address (TPA)		

Abbildung 3-12: Aufbau der Nachricht ARP-Request und –Reply (vgl. Wikipedia2)

3.7.1 ARP Probe

Dieser Begriff wird in der IPv4-Adressenkonflikt-Erkennung (RFC 5227) benutzt. Vor Beginn der Verwendung einer IPv4 Adresse (ob empfangen durch manuelle Konfiguration, DHCP, oder andere Mittel), muss ein Rechner, der RFC 5227 umsetzt, einen Test durchführen, um herauszufinden, ob die Adresse bereits benutzt wird, indem er eine ARP-Probe broadcastet. Ein ARP-Request, in dem die IP-Adresse des Senders aus Nullen besteht, wird als ARP-Probe interpretiert.

3.8 RARP

Das Protokoll RARP (Reverse Address Resolution Protocol) ist für Stationen gedacht, die ihre IP-Adresse nicht selbst speichern können (z. B. Remote-Boot-Stationen ohne Festplatte). RARP ist das Gegenstück zu ARP. Es ermöglicht, bei einer bekannten MAC-Adresse die zugehörige IP-Adresse zu finden [RFC 903]. Bei RARP ist es notwendig, einen speziellen Server festzulegen, in dem eine RARP-Tabelle enthalten ist. Der Server sucht in dieser Tabelle nach der IP-Adresse, die mit der angeforderten MAC-Adresse übereinstimmt und gibt die gesuchte IP-Adresse als RARP-Reply bekannt.

Der Aufbau von RARP-Nachrichten ist identisch mit dem von ARP-Nachrichten [vgl. Abbildung 3-12]. Bei RARP werden im Feld Operation die Werte 3 für RARP-Request und 4 für RARP-Reply verwendet.

3.9 ICMPv4

In jedem Netz treten von Zeit zu Zeit Fehler auf, die an die Verursacher oder davon Betroffenen gemeldet werden müssen. Diese Aufgabe wird vom Protokoll ICMP (Internet Control Message Protocol) übernommen. Hierfür stellt das ICMP eine Vielzahl von sog. ICMP-Nachrichten zur Verfügung. ICMP wurde bereits im Jahr 1981 im RFC 792 spezifiziert. In den RFCs 950, 1256, 1393, 1812 und 4884 wurde der Funktionsumfang erweitert (vgl. Badach 2007 94-100).

Zu den wichtigsten Aufgaben des Protokolls ICMP gehören u. a.:

- Unterstützung der Diagnose durch Hilfsprogramme
  - Hilfsprogramm „Ping“, wird üblicherweise in Netzwerken zum Feststellen der Erreichbarkeit des Kommunikationspartners verwendet. Es sendet die ICMP-Nachricht Echo Request an eine IP-Adresse und wartet auf die ICMP-Nachricht Echo Reply als Antwort. Es meldet die Anzahl der empfangenen Antworten und die Zeitspanne zwischen Senden der Anfrage und Eingang der Antwort.
  - Hilfsprogramm „traceroute“, wird zum Verfolgen von Routen eingesetzt
- Unterstützung der Aufzeichnung von Zeitmarken (Timestamps) sowie Ausgabe von Fehlermeldungen bei abgelaufenem Timestamps von IP-Paketen
- Verwaltung von Routingtabellen
- Unterstützung der Flusskontrolle, um eine Überlastung des Routers bzw. des Zielrechners zu vermeiden (ICMP Source Quench)
- Mitwirken bei der Feststellung der maximalen Länge von IP-Paketen, d. h. von MTU (Maximum Transfer Unit)

3.9.1 ICMP-Nachrichten

Den Aufbau von ICMP-Nachrichten zeigt Abbildung 3-13. Da ICMP unterschiedliche Informationen zu transportieren hat, enthalten die ICMP-Nachrichten einen Header, der in allen Nachrichten immer gleich ist. Die Bedeutung von ICMP-Angaben, die direkt nach dem Header folgen, ist von einzelnen Fehlern bzw. Diagnosesituationen abhängig. Für die Struktur des Teils ICMP Data bei den einzelnen ICMP-Nachrichtentypen sei auf die Dokumente RFC 792 und RFC 1256 verwiesen.

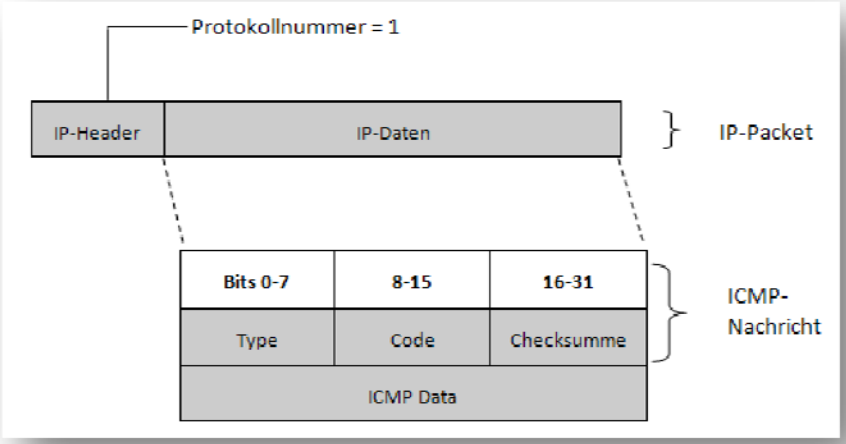


Abbildung 3-13: Aufbau von ICMP Nachrichten (vgl. Badach 2007, S. 95)

Die einzelnen Angaben im ICMP-Header sind:

- **Type** dient als Unterscheidung der Bedeutung einzelner ICMP-Nachrichten (Abbildung 3-14)
- **Code** ist eine weitere Unterteilung der Bedeutung der Nachricht innerhalb eines Typs. Beispielsweise in der Nachricht „Destination Unreachable“ wird dem Absender eines IP-Pakets mitgeteilt, warum es nicht übermittelt werden konnte: 0 = Netz nicht erreichbar, 1 = Rechner nicht erreichbar, 2 = Protokoll nicht erreichbar 3 = Port nicht erreichbar und 4 = Fragmentierung erforderlich und: DF-Bit gesetzt
- **checksum (Prüfsumme)** enthält eine Prüfsumme, die nur die ICMP-Daten auf Fehler überprüft.

Type	Bedeutung der ICMP-Nachricht
0	*Echo Reply* (Echo-Antwort)
3	*Destination* Unreachable
4	*Source Quench* (Senderate reduzieren)
5	*Redirect* (Route ändern)
8	*Echo Request* (Echo-Anforderung)
9	*Router Advertisement* (Router-Bekanntmachung)
10	*Router Solicitation* (Suche nach einem Router)
11	*Time Exceeded* (Lebenszeit des IP-Pakets ist überschritten)
12	*Parameter Problem* (Parameterfehler im IP-Paket)
13	*Time Stamp Request* (Uhrzeitangabe-Anforderung)
14	*Time stamp Reply* (Uhrzeitangabe-Antwort)
15	*information Request* (Anforderung der Information)
16	*Information Reply* (Antwort auf Informationsanforderung)
17	*Adress Mask* Request (Abfrage der Subnetzmaske)
18	*Adress Mask* Response (Antwort auf Abfrage der Subnetzmaske)

Abbildung 3-14: Typen von ICMP-Nachrichten, für eine vollständige Liste wird auf [IANA] verwiesen (vgl. Badach 2007 S. 96).

3.9.2 ICMP-Anfragen

Nachfolgende Anfragen können ins Netz mit ICMP gestellt werden, um Informationen zu gewinnen.

- **Echo Request/Reply (Echo-Funktion)** ist die häufigste Anfragemeldung. Hierbei sendet das Ping-Programm ein Echo-Request zu einem bestimmten Ziel (Rechner bzw. Router). Das Ziel muss darauf mit einem Echo Reply antworten. Jeder IP-fähige Rechner muss auf ein Echo-Request antworten können.
- **Timestamp Request/Reply (Zeitmarkenanfrage):** Ein Rechner gibt eine Zeitmarkenanfrage mit Timestamp Request ab, um von einem anderen Rechner eine Zeitmarke zu erhalten, die das aktuelle Datum und die Uhrzeit angibt. Ein Rechner, der eine Zeitmarkenanfrage in Timestamp Request empfängt, antwortet mit Timestamp Reply, um die Laufzeit eines IP-Pakets über das Netz zu ermitteln.

- **Information Request/Reply (Informationsanfrage)** ermöglicht einem Rechner, seine IP-Adresse (z. B. von einem Adress-Server) abzufragen.
- **Address-Mask Request/Response (Abfrage der Subnetzmaske)** ermöglicht einem Rechner, die zu verwendende Subnetzmaske abzufragen. In einem Subnetzwerk, in dem diese Funktion unterstützt wird, sind ein oder mehrere Rechner als Subnetzmasken-Server gekennzeichnet.
- **Router Solicitation (Suche nach einem Router) und Router Advertisement (Router-Bekanntmachung)** [RFC 1256]**:** Ein Rechner kann während seiner Konfigurationsphase eine Nachricht Router Solicitation an alle Systeme (Rechner, Router) in demselben Subnetz verschicken. Diese Nachricht bedeutet, ich suche einen Router. Der Router antwortet mit der Nachricht Router Advertisement.
- **Path MTU Discovery (MTU Ermittlung)** [RFC 1191] ist ein Verfahren zum dynamischen Erkennen der maximalen Paketgröße für einen bestimmten Pfad im Netzwerk.

4 Auswahl der Protokolle

Nachdem die untersuchten Protokolle beschrieben worden sind, wird in diesem Kapitel festgelegt, welche davon am besten auf die gestellten Anforderungen zur Inbetriebnahme/Konfiguration des Netzwerks zugeschnitten werden können.

Es stellt sich die Frage, welche Anforderung ein Protokoll benötigt und welches zu ihrer Erfüllung eingesetzt werden kann.

Netzwerk automatisch in Betrieb nehmen/konfigurieren

Man braucht ein Protokoll, um die Anforderung „Netzwerk automatisch in Betrieb nehmen/konfigurieren“ zu erfüllen. Nachfolgende Kriterien müssen dafür erfüllt werden.

- Konfigurieren von Geräten als Netzwerkteilnehmer
- Senden und Empfangen von Standard-Broadcast-Nachrichten
- Auffinden von Geräten
- Authentifizierung von Nachrichten
- Übermittlung von herstellerspezifischen Daten
- Zentrale Verwaltung und Speicherung der Netzwerkkonfiguration
- Zentrale Abfrage der Netzwerkkonfigurationsdaten.

Nachfolgende Protokolle sind daraufhin untersucht worden (siehe Abbildung 4-1)

ARP kann eingesetzt werden, um ein Subnetzwerk zu scannen (ARPPing). Dazu ist eine ARP-Request-Nachricht (Broadcast) an jedes Gerät zu senden. Der wesentliche Vorteil von ARP ist, dass seine Nachrichten nicht von Firewalls geblockt werden. Es hat aber mehrere Nachteile. Ein ARP-Request kann von jedem Host im Subnetzwerk beantwortet werden, was eine Sicherheitslücke darstellt. Der MAC-Frame ist so festgelegt, dass keine weiteren Informationen (Gerätzustand, Authentifizierung, …) mit ARP übermittelt werden können. Daher kann es zur Erfüllung dieser Anforderung nicht eingesetzt werden.

RARP kann nur IP-Adressen übermitteln. Die Konfiguration der Netzwerkschnittstelle eines Geräts mit nur der IP-Adresse ist aber nicht ausreichend. Außerdem hat RARP einen festen Nachrichtenrahmen (siehe ARP), der verhindert, dass zusätzliche Informationen übertragen werden können. Es ist daher ungeeignet.

Zeroconfig kann zwar zur Konfiguration eines Geräts als Netzwerkteilnehmer eingesetzt werden, basiert aber auf ARP und ist daher nicht geeignet.

ICMP wird eingesetzt, um die Erreichbarkeit eines Hosts zu überprüfen (Ping). Will man Ping einsetzen, um das Netzwerk zu scannen, muss für jeden Host ein ICMP-Echo-Request

(parallel) gesendet und eine entsprechende Technik zum Empfang von ICMP-Echo-Reply bereitgestellt werden (vgl. fping). Mit viel Geschicklichkeit (Multithreading und Synchronisation) kann das Scannen des Netzwerks mit Ping sehr effizient sein. Das einzige Problem dabei ist, dass die Geräte bei der Erstinbetriebnahme keine gültige IP-Adresse haben und nur mit Broadcast-Nachrichten erreicht werden können. Man kann ICMP benutzen, um Konfigurationsparameter anzufragen, die zur Konfiguration der Netzwerkschnittstelle benötigt werden. Das Gerät muss für die IP-Adresse eine Information-Request-Anfrage senden, für die Subnet-Mask eine Address-Mask-Request-Anfrage und für die Gateway-Adresse eine Router-Solicitation-Anfrage. Grundsätzlich kann die Netzwerkkarte eines Geräts mit diesen drei Parametern konfiguriert werden. Das ist aber nicht ausreichend, da einige Netzwerkapplikationen noch weitere Informationen (wie DNS Server IP, Log-Server-IP …) benötigen, um einwandfrei funktionieren zu können. Diese können mit ICMP nicht ermittelt werden. Aus diesen Gründen ist ICMP auch nicht geeignet.

SNMP ist ein Protokoll, das grundsätzlich zum Management und Monitoring von Netzwerk-Geräten konzipiert worden ist. Es kann auch zur Konfiguration der Netzwerkschnittstelle der Geräte eingesetzt werden. Hierfür wird das SET-Paket mit den entsprechenden Konfigurationsparametern verwendet. Es kann außerdem zum Scannen des Netzwerks eingesetzt werden, vorausgesetzt, jedes Gerät hat einen funktionierenden SNMP-Agenten. SNMP hat aber den Nachteil, dass es keine Standard Broadcast-Nachrichten definiert. Darum ist es auch auszuschließen.

NETCONF ist dank XML-RPC ein sehr gut strukturiertes Protokoll zur Konfiguration und zum Management von Netzwerkelementen. Da die Operationen aber über RPC erfolgen, ist das Broadcasten von Nachrichten nicht möglich. Des Weiteren wird für RPCs eine Middleware benötigt, die von jedem Embedded-Gerät nicht bereitgestellt werden kann. Daher passt NETCONF nicht.

DHCP ist ein Protokoll zur dynamischen Konfiguration von Netzwerkelementen. Alle Konfigurationen des Netzwerks werden an einer zentralen Stelle (DHCP-Server) verwaltet und gespeichert. Es ist möglich mit DHCP Standard-Broadcast-Nachrichten zu senden. Die DHCP-Nachrichten können authentifiziert werden. DHCP definiert auch eine Option zum Transportieren von herstellerspezifischen Informationen. Es hat aber auch einige Nachteile. So ist es nicht möglich, die im Netzwerk angeschlossenen Client-Geräte aufzufinden, wenn deren IP-Adressen nicht bekannt sind und die im DHCP Server gespeicherten Informationen von einem anderen Rechner abzufragen. DHCP beschreibt keinen Mechanismus zur Konfiguration des Netzwerks von einem entfernten Rechner (Konfig-Tool) aus. Aus diesen Gründen eignet sich DHCP auch nicht.

Fazit: Wie in Abbildung 4-1 zu sehen ist, hat sich keines der untersuchten Protokolle zur automatischen Inbetriebnahme/Konfiguration des Netzwerks als geeignet erwiesen. Es soll aber ein Protokoll als Basis ausgewählt werden. Die in Frage kommenden Protokolle sind NETKONF, SNMP und DHCP. DHCP kann, im Gegensatz zu NETCONF und SNMP Standard-Broadcast-Nachrichten senden und empfangen. Aus diesem Grund wird DHCP als

Basis Protokoll ausgewählt. Deshalb werden im Folgenden die Komponenten des Systems umbenannt in DHCP-Konfigurationstool, DHCP-Server und DHCP-Client.

	ARP	RARP	Zerocon-fig	ICMP	SNMP	NETCONF	DHCP
Konfigurieren von Geräten als Netzwerkteilnehmer	✗	✓	✓	✓	✓	✓	✓
Senden und Empfangen von Standard-Broadcast-Nachrichten	✓	✓	✓	✗	✗	✗	✓
Authentifizierung von Nachrichten	✗	✗	✗	✓	✓	✓	✓
Übermitteln von herstellerspezifischen Daten	✗	✗	✗	✗	✓	✓	✓
Auffindung von Geräten	✗	✗	✗	✗	✗	✗	✗
Zentrale Verwaltung und Speicherung der Netzwerkkonfig.	✗	✗	✗	✗	✓	✓	✓
Zentrale Abfrage der Netzwerkkonfig.	✗	✗	✗	✗	✗	✓	✗

Legende: geeignet: ✓ nicht geeignet: ✗

Abbildung 4-1: Eignung der Protokolle für die automatische Inbetriebnahme/Konfiguration des Netzwerks

Generell gilt für weitere Anforderungen, die ein Protokoll benötigen, dass ihnen jeweils nur eines zugeordnet werden kann. Für die automatische Ermittlung der Netzwerk-Eigenschaften wird ICMP (Abschnitt 3.9) und für die Erkennung von Adressenkonflikten ARP (Abschnitt 3.7) oder Ping (Abschnitt 3.9) eingesetzt.

5 Entwurf

Dieses Kapitel beschäftigt sich mit dem Entwurf des Systems. Dabei wird seine Architektur dargestellt und die Spezifikation der Komponenten festgelegt. Dabei wird auf die ausgewählten Protokolle verwiesen und nur auf die an diesen Protokollen vorgenommen Änderungen näher eingegangen.

5.1 Architektur des Systems

Die Systemarchitektur (Abbildung 5-1) stellt eine Übersicht über die Komponenten des Systems und die Art und Weise dar, wie diese miteinander verbunden sind.

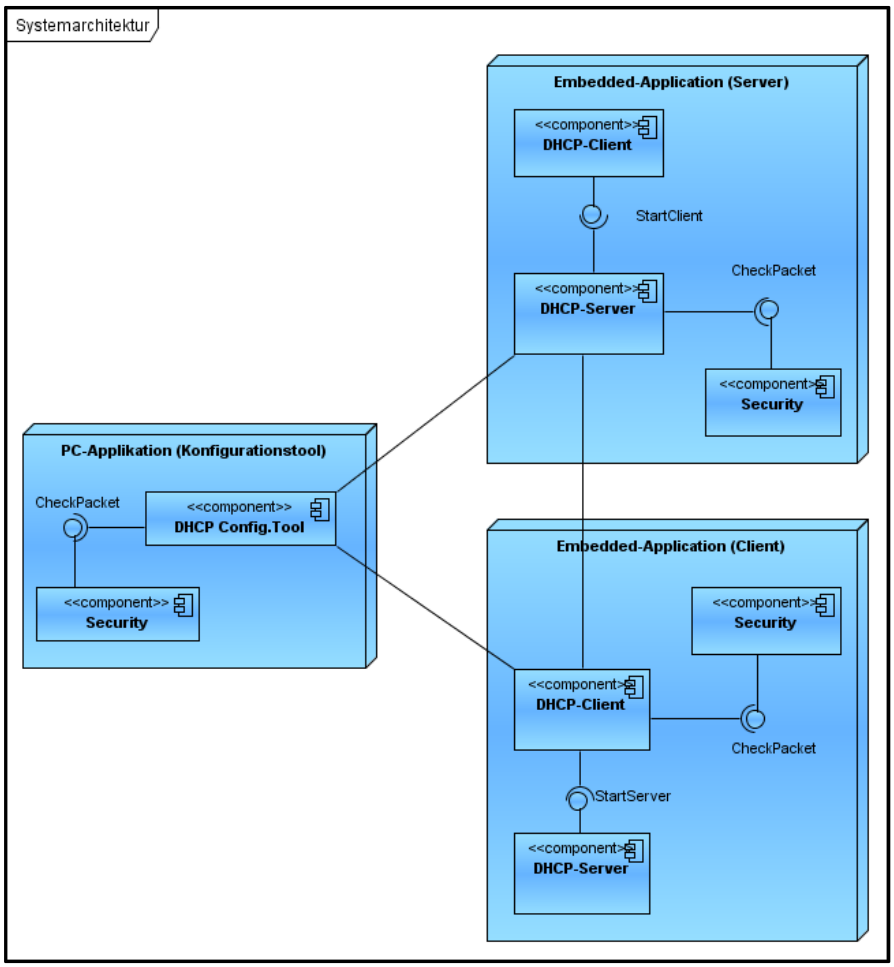


Abbildung 5-1: Architektur des Systems

Das System zur Inbetriebnahme/Konfiguration des Netzwerks besteht aus vier Komponenten (DHCP-Server, DHCP-Client, Konfigurationstool und Security), die auf zwei verteilte Applikationen (Embedded- und PC-Applikation) verteilt sind. Die Embedded-Applikation läuft auf einem Embedded Gerät und kann sowohl im Servermodus als auch im Clientmodus gestartet werden. Die PC-Applikation läuft auf einem Desktop-PC bzw. Notebook und enthält das DHCP-Konfigurationstool.

Die verteilten Komponenten kommunizieren über Nachrichtenaustausch. Die Beziehungen zwischen ihnen werden durch die Assoziationslinien in Abbildung 5-1 dargestellt. Da es sich um DHCP handelt, wird UDP als Transportprotokoll für den Nachrichtenaustausch eingesetzt.

Die Komponenten aus derselben Applikation kommunizieren über die jeweiligen zur Verfügung stehenden Schnittstellen. DHCP-Server, DHCP-Client und das Konfigurationstool müssen z. B. die Authentizität und Integrität aller empfangenen Nachrichten durch die Security (Sicherheit) überprüfen lassen. Daraufhin geschieht der Transfer über die Check-Packet-Schnittstelle.

5.2 Spezifikationen der Komponenten des Systems

Die Spezifikationen der Komponenten des Systems entsprechen den für DHCP festgelegten Spezifikationen und dem SDeDi-DHCP (Secure Device Discovery DHCP), das in dieser Arbeit entworfen worden ist und eine Erweiterung von DHCP ist. SDeDi-DHCP wird im Abschnitt 5.3 ausführlich beschrieben. Die Systemkomponenten und deren Spezifikationen werden nachfolgend aufgelistet.

- Sicherheitskomponente: RFC 3118
- DHCP-Server und -Client: RFC 2131 + SDeDi-DHCP
- DHCP-Konfigurationstool: RFC 2131 + SDeDi-DHCP

5.3 SDeDi-DHCP

Im Kapitel 4 ist festgestellt worden, dass keines der untersuchten Protokolle zur automatischen Inbetriebnahme/Konfiguration des Netzwerks eingesetzt werden kann. DHCP wurde aber als Basis-Protokoll ausgewählt, weil es die meisten gewünschten Kriterien erfüllte, insbesondere das Senden/Empfangen von Broadcast-Nachrichten und die Übermittlung von herstellerspezifischen Daten. Um geeignet zu sein, musste DHCP erweitert werden. Die Entwicklung mit den nötigen Erweiterungen erfolgte in dieser Bachelorarbeit und wurde SDeDi-DHCP (Secure Device Discovery Dynamic Host Configuration Protocol) genannt. Durch das SDeDi-DHCP wird es möglich, die Geräte unabhängig von deren IP-Adressen zu finden. Es findet nur die bekannten Gerätetypen (selektiv), unbekannte werden automatisch herausgefiltert. Diese selektive Fähigkeit wird durch den Einsatz des Sicherheitsprotokolls realisiert. Es ist nicht nur für das selektive Auffindung der Geräte zuständig, sondern auch für die Überprüfung der Authentizität und der Integrität der Nachrichten. SDeDi ist aber nicht nur dafür konzipiert worden, sondern es füllt auch alle Lücken des DHCP-Protokolls bezüglich der Anforderungen für die automatische Inbetriebnahme/Konfiguration des Embedded-TCP/IP-Netzwerks.

5.3.1 Lösungsansatz

Laut DHCP dürfen keine herstellerspezifischen Nachrichten verwendet werden. Daher werden die Nachrichten für das SDeDi-DHCP in „vendor specific information options“ [RFC 2132] eingekapselt. Um konform zu DHCP zu bleiben, werden für die SDeDi-DHCP-Nachrichten Standard-DHCP-Nachrichten verwendet, jedoch mit einer anderen Semantik. Die passende Semantik der Nachricht wird in der „vendor specific information options“ codiert. Alle Nachrichten werden nach „Authentication vor DHCP Messages“ [RFC 3118] authentifiziert und nicht authentifizierte Nachrichten verworfen.

Die im SDeDi-DHCP neu definierten Nachrichten werden anhand der drei nachfolgenden Kommunikationsmodelle beschrieben.

5.3.2 Erstinbetriebnahme-Kommunikationsmodell

Bei einer Erstinbetriebnahme sind alle Geräte Client. Ein Server wurde noch nicht festgelegt. Alle Geräte haben entweder die werkseingestellte IP-Adresse oder keine und müssen aufgefunden und konfiguriert werden. Die bei einer Erstinbetriebnahme zu verwendenden Nachrichten werden in Abbildung 5-2 dargestellt und nachfolgend beschrieben.

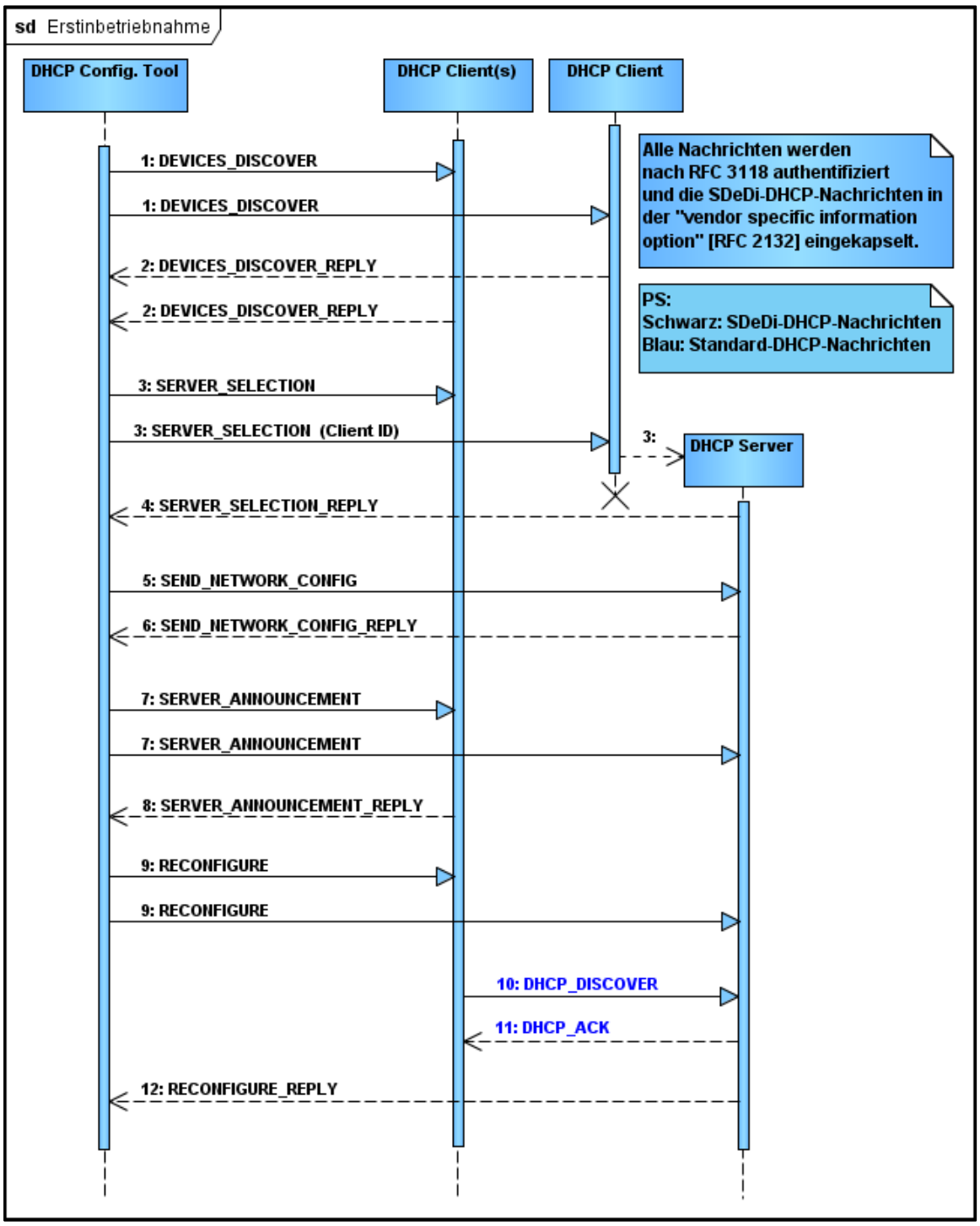


Abbildung 5-2: Sequenzdiagramm für die Erstinbetriebnahme des Netzwerks

DEVICES_DISCOVER (DHCP_DISCOVER): DHCP Configuration Tool to Devices

Eine DEVICES_DISCOVER-Nachricht wird vom DHCP-Konfigurationstool an das ihm zugehörige Subnetzwerk gebroadcastet. Zweck dieser Operation ist, das Netzwerk zu scannen, um die zu konfigurierenden Geräte zu finden und Informationen über deren aktuelle Konfigurationen (IP-Adresse, Subnetzwerk etc.) bzw. Zustand zu gewinnen. Da keine DEVICES_DISCOVER-Nachricht von DHCP festgelegt wurde, wird sie durch eine normale DHCP_DISCOVER-Nachricht versendet. Hierbei operiert das Konfigurationstool als ein DHCP-Client, der nach einem DHCP-Server sucht. Da die DHCP_DISCOVER-Nachricht nur für DHCP-Server bestimmt ist, muss jeder DHCP-Client zusätzlich den Server-Port 67 abhören. Bekommt ein Client eine Nachricht vom DHCP-Server Port, überprüft er, ob die Nachricht eine SDeDi-DHCP-Nachricht ist (enthält eine passende „vendor specific information option“). Ist das der Fall, wird die Nachricht entsprechend interpretiert. Andernfalls wird sie ignoriert.

DEVICES_DISCOVER_REPLY (DHCP_ACK): Devices to DHCP Configuration Tool.

Nach einer DEVICES_DISCOVER-Nachricht wartet das DHCP-Konfigurationstool auf alle DEVICES_DISCOVER_REPLYs-Nachrichten für eine maximal definierte Zeit (MAX_DEVICES_DISCOVER_TIME). Die Wartezeit hängt von der geschätzten Anzahl der am Netz angeschlossenen Geräte (bekannte und unbekannte) ab. Unbekannte Geräte werden dazugezählt, weil auch deren Nachrichten überprüft werden, bevor sie ignoriert werden. Analog zu DEVICES_DISCOVER wird eine DEVICES_DISCOVER_REPLY Nachricht durch eine DHCP_ACK versendet.

Da DHCP das UDP als Transport-Protokoll verwendet, können Nachrichten verloren gehen, und das Konfigurationstool wird davon nie erfahren. Deswegen sollte die DEVICES_DISCOVER-Nachricht mehrfach versendet werden. Die maximale Anzahl der DEVICES_DISCOVER Nachrichten, die versendet werden können, wird durch die Konstante MAX_DEVICES_DISCOVER festgelegt (typischerweise 4). Neue Geräte werden der Liste der schon gefundenen Geräte hinzugefügt, doppelte Einträge müssen gelöscht werden.

Wird der Vorgang DEVICES_DISCOVER als „fertig“ gekennzeichnet, werden alle gefundenen Geräte in einer angemessenen Form dargestellt. Anschließend kann die Konfiguration des Netzwerks beginnen. Der für die gefundenen Geräte reservierte IP-Bereich wird eingetragen, ein Gerät als Server ausgewählt, konfiguriert und die restlichen Geräte als Client konfiguriert. Schließlich kann das Senden weiterer Nachrichten fortgesetzt werden.

SERVER_SELECTION (DHCP_DISCOVER): DHCP Configuration Tool to selected Device

Bei einer Erstinbetriebnahme sind, wie bereits erwähnt, alle Geräte Client. Es muss ein Server eingerichtet werden, bevor die Konfiguration der Clients gestartet wird. Dazu ist eine SERVER_SELECTION-Nachricht an einen Client aus der Liste der gefundenen Geräte zu senden. Die Auswahl des Clients wird manuell über das Tool vom Errichter durchge-

führt. Eine SERVER_SELECTION-Nachricht enthält außerdem alle notwendigen Parameter (IP-Adresse, Subnetzwerk-Maske, …) zur Konfiguration der Netzwerkkarte des Servers.

Empfängt ein Client eine SERVER_SELECTION Nachricht, richtet er seine Netzwerkschnittstelle mit den empfangenen Parametern ein und wechselt in den Server-Modus. Danach antwortet er auf die SERVER_SELECTION-Nachricht mit einer SEVER_SELECTION_REPLY-Nachricht.

SERVER_SELECTION_REPLY (DHCP_ACK): DHCP Server to DHCP Configuration Tool

Mit einer SERVER_SELECTION_REPLY-Nachricht teilt das Gerät dem Konfigurationstool mit, dass er die SERVER_SELECTION-Nachricht erfolgreich empfangen hat und nun als DHCP-Server operiert.

SEND_NETWORK_CONFIG (DHCP_DISCOVER): DHCP Configuration Tool to DHCP Server

Mit dieser Nachricht teilt das Konfigurationstool dem DHCP-Server die Konfiguration des Netzwerks mit. Die Mitteilung kann auf mehrere Arten erfolgen. Die Art der Mitteilung wird mit dem „SDeDi-option **NETWORK_CONFIG_LOCATION**“ festgelegt.

Die NETWORK_CONFIG_LOCATION Option kann folgende Inhalte haben.

- **DHCP:** Die Konfiguration des Netzwerks wird in der Nachricht SEND_NETWORK_CONFIG eingebettet. Da UDP-Nachrichten nicht fragmentiert werden können, ist die MTU (üblich 1500) zu beachten. Passt die Konfiguration nicht in ein Paket, ist eine geeignete Sequenznummer-Technik zur Segmentierung einzusetzen.
- **TFPT** gibt die Adresse der Konfigurationsdatei in einem TFTP-Server an.
- **FTP** gibt die Adresse der Konfigurationsdatei in einem FTP-Server an.

SEND_NETWORK_CONFIG_REPLY (DHCP_ACK): DHCP Server to DHCP Configuration Tool

Mit dieser Nachricht teilt der Server dem Konfigurationstool mit, dass er die Konfiguration erfolgreich bekommen bzw. heruntergeladen hat. Wurden mehrere Pakete über DHCP versendet, ist der Empfang einer SEND_NETWORK_CONFIG_REPLY die Bestätigung eines Pakets mit der jeweiligen Sequenznummer.

SERVER_ANNOUNCEMENT (DHCP_DISCOVER): -optional- DHCP Configuration Tool to DHCP Clients

Das Konfigurationstool kann die Clients über den ausgewählten Server informieren. Hierbei broadcastet er eine SERVER_ANNOUNCEMENT-Nachricht. Erhält ein Client diese Nachricht, speichert er die IP-Adresse des Servers und kann seine Konfigurationsdaten mit ei-

ner Unicast-Nachricht anfordern. Der Client kann nach erfolgreichem Empfang der Nachricht eine Antwort an das Konfigurationstool senden.

SERVER_ANNOUNCEMENT_REPLY (DHCP_ACK): -optional- DHCP Client to DHCP Configuration Tool

Diese Nachricht dient der Bestätigung des Empfangs einer SERVER_ANNOUNCEMENT-Nachricht.

RECONFIGURE (DHCP_DISCOVER): DHCP Configuration Tool to DHCP Clients

Mit dieser Nachricht werden die Clients darüber informiert, dass deren Konfigurationen bzw. die des Netzwerks sich geändert haben. Eine RECONFIGURE-Nachricht kann als Unicast an einen Client oder als Broadcast an alle Clients gesendet werden. Erhält ein Client eine RECONFIGURE-Nachricht, fordert er vom DHCP Server mit einem Standard DHCP_DISCOVER die neuen Konfigurationen per Unicast oder Broadcast an. Anschließend antwortet der Server mit einer Standard DHCP_ACK-Nachricht. Hat der Client die RECONFIGURE-Nachricht per Broadcast empfangen, muss er das Senden der DHCP_DISCOVER-Nachricht mit einer zufälligen generierten Zeit verzögern. Die Verzögerung wird mit der Konstante **RECONFIGURE_DELAY** festgelegt. Damit die RECONFIGURE_DELAYs möglichst unterschiedlich sind, können sie mithilfe der Client-ID (z. B. MAC-Adresse) berechnet werden. Ziel der Verzögerung ist es, die Überlastung des Netzwerks sowie die Überflutung des Servers mit DHCP_DISCOVER Nachrichten zu vermeiden.

RECONFIGURE_REPLY (DHCP_ACK): DHCP Server to DHCP Configuration Tool

Ist der RECONFIGURE-Vorgang vom DHCP-Server als abgeschlossen gekennzeichnet, sendet er eine RECONFIGURE_REPLY an das Konfigurationstool. Der Vorgang wird als abgeschlossen gekennzeichnet, wenn alle über die RECONFIGURE-Nachricht getriggerten Clients reagiert haben oder die **MAX_RECONFIGURE_TIME** abgelaufen ist**.** Diese hängt von der Anzahl der Clients ab. Das Konfigurationstool verwendet **MAX_RECONFIGURE_TIME + ca. 2 s** um den Vorgang als abgeschlossen zu kennzeichnen, wenn er keine RECONFIGURE_REPLY-Nachricht vom Server empfängt. Nach Ablauf des Konfigurationstool-Timers oder Erhalt einer RECONFIGURE_REPLY wird das Netzwerk erneut gescannt, um die Änderungen anzuzeigen oder entsprechende Maßnahmen zur Beseitigung aufgetretener Probleme bei dem Vorgang zu treffen.

5.3.3 Konfiguration-Kommunikationsmodell (Server wird behalten)

Wurde das Netzwerk erfolgreich in Betrieb genommen, kann es jederzeit rekonfiguriert werden. Viele Gründe können zur Rekonfiguration des Netzwerks gegeben sein. Nachfolgend werden einige aufgelistet.

- Server wird wegen Wartung gewechselt.
- Neue Geräte sind im Netzwerk angeschlossen worden.
- Die Struktur des Netzwerks hat sich geändert.
- Der reservierte Bereich hat sich geändert.
- Geräte werden aus dem Netzwerk entfernt.

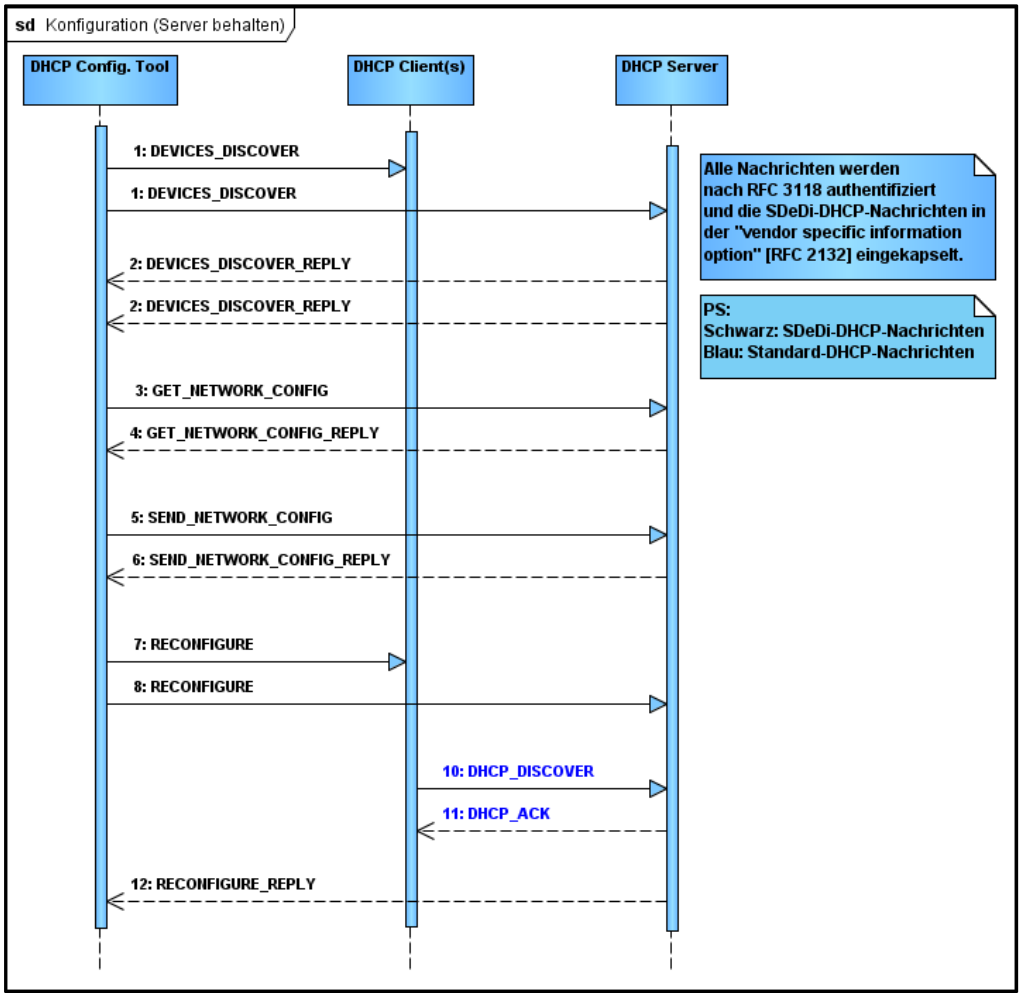


Abbildung 5-3: Sequenzdiagramm für eine Konfiguration mit „Server behalten“

In Abbildung 5-3 wird ein Kommunikationsmodell dargestellt, in dem der Server nicht gewechselt wird. Hierbei wird auf zwei neue, noch nicht behandelte Nachrichten hingewiesen (GET_NETWORK_CONFIG, GET_NETWORK_CONFIG_REPLY).

GET_NETWORK_CONFIG (DHCP_DISCOVER): DHCP Configuration Tool to DHCP

Server (vgl. SEND_NETWORK_CONFIG im Abschnitt 5.3.2).

Das Konfigurationstool fordert mit dieser Nachricht die aktuellen Netzwerkkonfigurationsdaten vom DHCP Server an.

GET_NETWORK_CONFIG_REPLY (DHCP_ACK) DHCP Server to DHCP Configuration Tool (vgl. SEND_NETWORK_CONFIG_REPLY in Abschnitt 5.3.2).

Mit dieser Nachricht teilt der DHCP-Server dem Konfigurationstool den Ort der aktuellen Konfiguration mit.

5.3.4 Konfiguration-Kommunikationsmodell (Server wechseln)

Ein Kommunikationsmodell, in dem der Server gewechselt wird, wird in Abbildung 5-4 dargestellt.

Wie bereits im Abschnitt 5.3.3 erwähnt, kann bei einer Rekonfiguration des Netzwerks der Server gewechselt werden. Soll der Server gewechselt werden, sind folgende Schritte erforderlich (vgl. Abbildung 5-4).

1. Aktuelle Netzwerkkonfiguration vom aktuellen Server anfordern und sichern (**GET_NETWORK_CONFIG**).
2. Aktuellen Server zurücksetzen, dann wird er Client mit IP-Adresse 0.0.0.0. (**CHANGE_TO_CLIENT**).
   Erhält der Server die Nachricht CHANGE_TO_CLIENT, setzt er seine Netzwerkschnittstelle-Konfiguration zurück und wechselt in den Client Modus. Danach antwortet er mit einem CHANGE_TO_CLIENT_REPLY.
3. Neuen Server auswählen (SERVER_SELECTION).
4. Weiterer Verlauf siehe Abbildung 5-4 und vgl. Abschnitt 5.3.2.

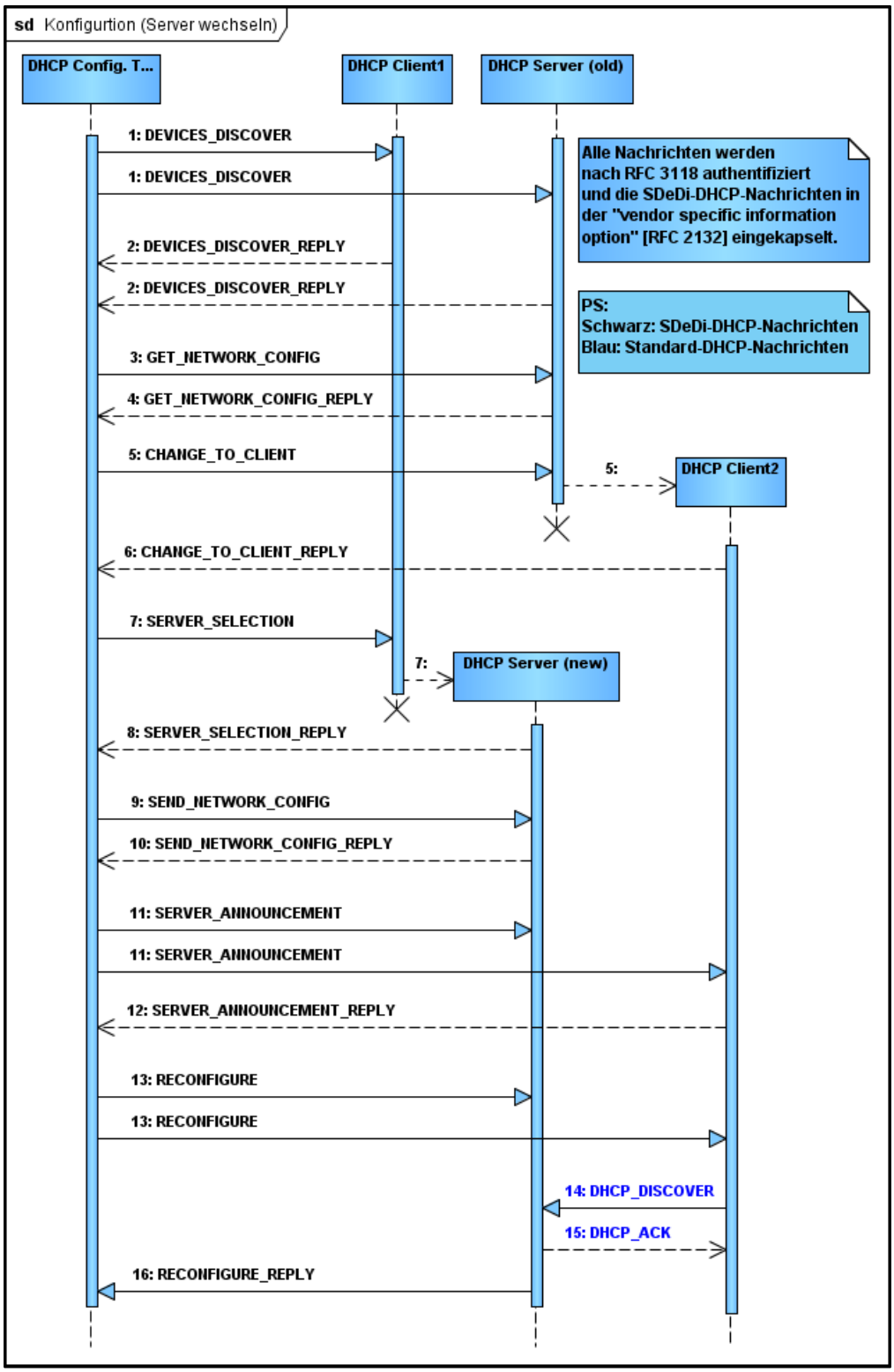


Abbildung 5-4: Sequenzdiagramm für eine Konfiguration mit „Server wechseln“

5.4 Verhaltensmodelle

In diesem Abschnitt werden die Verhaltensmodelle der Komponenten des Systems dargestellt und beschrieben. Diese werden durch Zustandsautomaten modelliert und zeigen, welche Zustände die Komponenten annehmen können und wie sie auf welche Nachrichten bzw. Ereignisse in welchen Zuständen reagieren.

5.4.1 DHCP-Client Verhaltensmodelle

In diesem Abschnitt werden die Verhaltensmodelle eines DHCP-Client dargestellt.

5.4.1.1 DHCP-Client Verhaltensmodell (allgemein)

Das allgemeine Verhalten eines DHCP-Clients wird durch Abbildung 5-5 dargestellt und nachfolgend beschrieben (vgl. RFC 2132). Wie in Abbildung 5-5 zu ersehen ist, kann ein DHCP-Client in zwei verschiedenen Zuständen starten, entweder im INIT oder im INIT-REBOOT.

Start im INIT-Zustand

Er startet im INIT-Zustand, wenn die IP-Adresse ungültig (Lease-Zeit abgelaufen) oder unbekannt (Erststart, Netzwerkkarte noch nicht konfiguriert) ist. Hierbei broadcastet er ein DHCP_DISCOVER und wechselt in den SELECTING-Zustand, wo er DHCP_OFFERs von Servern sammelt. Danach wählt er einen Server aus und sendet eine DHCP_REQUEST Nachricht mit der ausgewählten Server-ID, um die Konfigurationsparameter anzufordern. Danach wechselt er in den REQUESTING-Zustand, in dem er eine DHCP_NAK oder DHCP_ACK empfangen kann. Empfängt er ein DHCP_NAK, konnte der Server die geforderten Konfigurationsparameter nicht zur Verfügung stellen. Dann kehrt er in den INIT-Zustand zurück. Empfängt er jedoch ein DHCP_ACK, überprüft er, ob die erhaltene IP-Adresse konfliktfrei ist. Ist dies der Fall, setzt er die empfangenen Konfigurationsparameter ein, speichert die Lease-Zeit und startet die Timer T1 = ½* (Lease-Zeit) und T2 = ¾* (Lease-Zeit) im Fall einer dynamischen IP-Adresse ab und wechselt in den BOUND-Zustand. Steht die IP-Adresse im Konflikt, sendet er dem Server eine DHCP_DECLINE Nachricht und kehrt in den INIT-Zustand zurück.

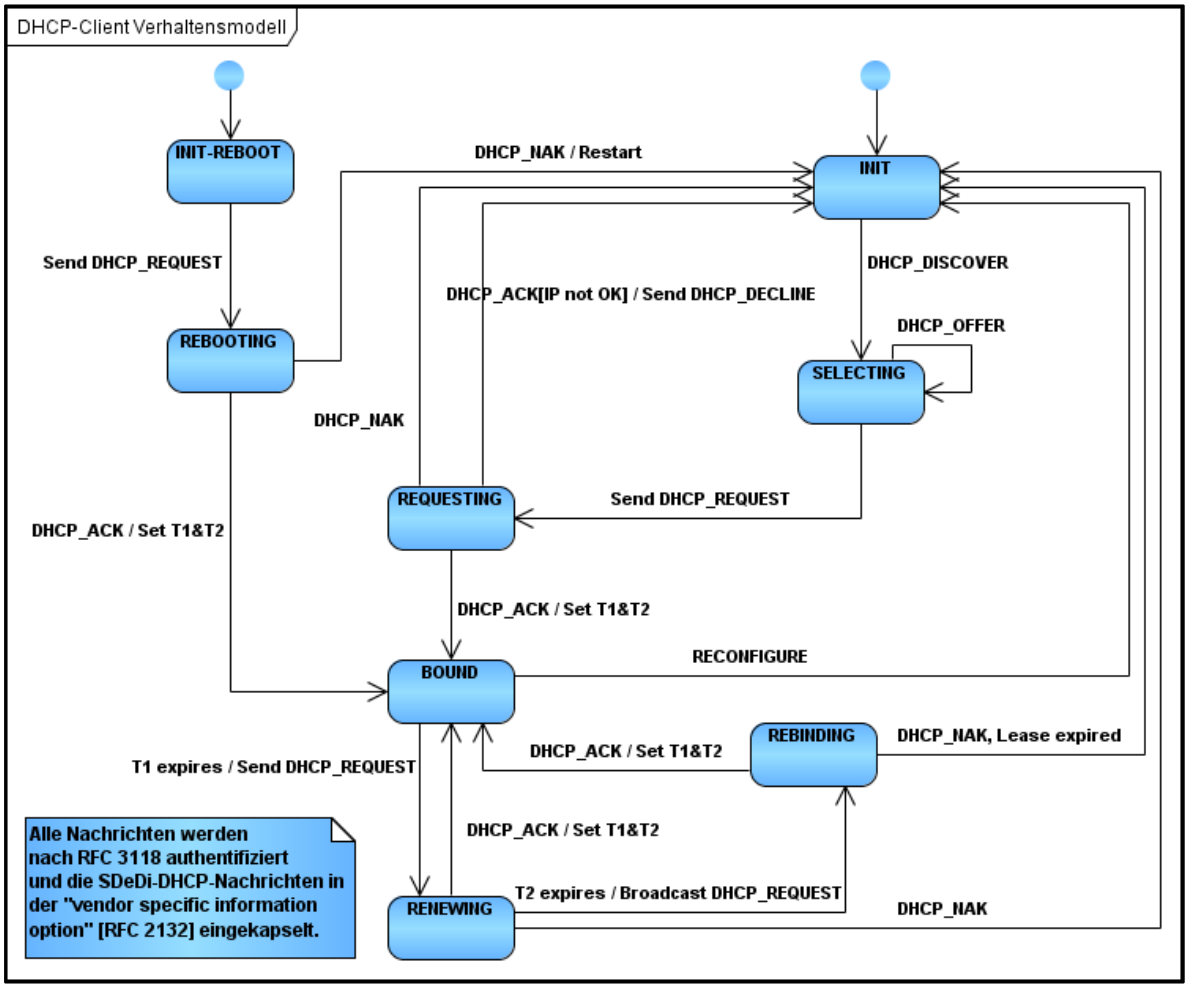


Abbildung 5-5: DHCP-Client Verhaltensmodell [vgl. RFC 2131, RFC 3203]

Befindet sich der Client im BOUND-Zustand, ist er konfiguriert und kann auf folgende Nachrichten bzw. Ereignisse reagieren.

- Erhalt einer RECONFIGURE Nachricht: Hierbei wechselt er in den INIT-Zustand und startet den Prozess neu.
- Ablauf T1: Hiebei sendet er ein DHCP_REQUEST, um die IP-Adresse zu erneuen und wechselt in den RENEWING Zustand.

Befindet er sich im RENEWING-Zustand, kann er auf nachfolgende Nachrichten bzw. Ereignisse reagieren.

- Empfang eines DHCP_ACK ==> Neue Lease-Zeit speichern, T1 und T2 starten und in den BOUND-Zustand wechseln.

- Empfang eines DHCP_NAK ==> Wechseln in den INIT-Zustand.
- Ablauf von T2 ==> Broadcasten einer DHCP_REQUEST Nachricht und wechseln in den REBINDING-Zustand.

Wurde er wegen des Ablaufs von T2 in den REBINDING-Zustand überführt, reagiert er auf nachfolgende Ereignisse bzw. Nachrichten folgendermaßen:

- Empfang eines DHCP_ACK ==> Neue Lease-Zeit speichern, T1 und T2 starten und in den BOUND-Zustand wechseln.
- Empfang eines DHCP_NAK ==> Wechseln zum INIT-Zustand.
- Ablauf der Lease-Zeit ==> Wechseln zum INIT-Zustand.

Start im INIT-REBOOT-Zustand

Der DHCP-Client startet im INIT-REBOOT-Zustand, wenn die Netzwerkkarte des Geräts bereits konfiguriert wurde und die aktuellen Konfigurationsparameter noch gültig sind. In diesem Zustand sendet er ein DHCP_REQUEST an den Server, um seine Konfigurationsparameter zu erneuen und wechselt in den REBOOTING-Zustand. Empfängt er im REBOOTING-Zustand ein DHCP_ACK, überprüft er, ob die erhaltene IP-Adresse konfliktfrei ist. Ist dies der Fall, setzt er die empfangenen Konfigurationsparameter ein, speichert die Lease-Zeit und wechselt in den BOUND-Zustand. Empfängt er jedoch ein DHCP_NAK vom Server, sind die gespeicherten Konfigurationsparameter nicht mehr gültig. Er muss in den INIT-Zustand wechseln und den Prozess neu starten.

5.4.1.2 SDeDi-DHCP-Client Verhaltensmodell (mit einer statischen IP-Adresse)

Das Verhalten eines SDeDi-DHCP-Client bei einer Konfiguration mit einer statischen IP-Adresse zeigt Abbildung 5-6. Diese Konfigurationsart wird für statische Geräte (BMZs, Server, …) eingesetzt. Dabei sind die Zustände RENEWING und REBINDING, die zur Verwaltung der Lease-Zeit vorgesehen sind, wegzulassen. Außerdem wurde die „rapid commit option“ eingesetzt, um die vier Nachrichtenaustauschphasen (vgl. Abschnitt 3.1.4) zu vermeiden und nur mit zwei Phasen (vgl. Abschnitt 3.1.5) für die Konfiguration auszukommen. Mit dem Einsatz dieser Option fällt der Zustand REQUESTING weg.

In Abbildung 5-6 ist außerdem zu sehen, in welchem Zustand der Client auf SDeDi-DHCP-Nachrichten reagiert. Im INIT-Zustand kann er auf die DEVICES_DISCOVER und RECONFIGURE und im BOUND auf die RECONFIGURE und CHANGE_TO_SERVER-Nachrichten reagieren.

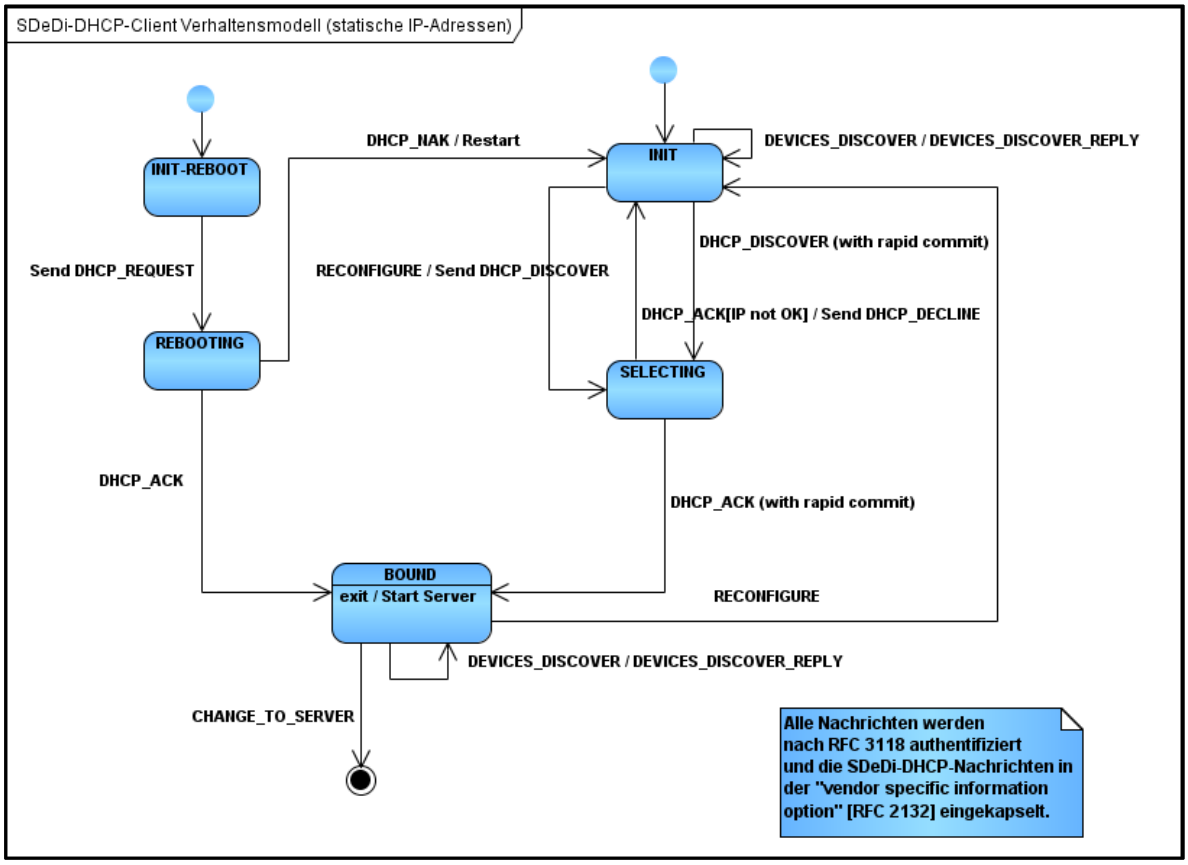


Abbildung 5-6: SDeDi-DHCP-Client Verhaltensmodell (statische IP-Adressen) [vgl. RFC 2131, RFC 4039 und SDeDi-DHCP]

5.4.2 SDeDi-DHCP-Server Verhaltensmodell

Das Verhalten eines SDeDi-DHCP-Servers zeigt Abbildung 5-7 (vgl. RFC 2131 und SDeDi-DHCP).

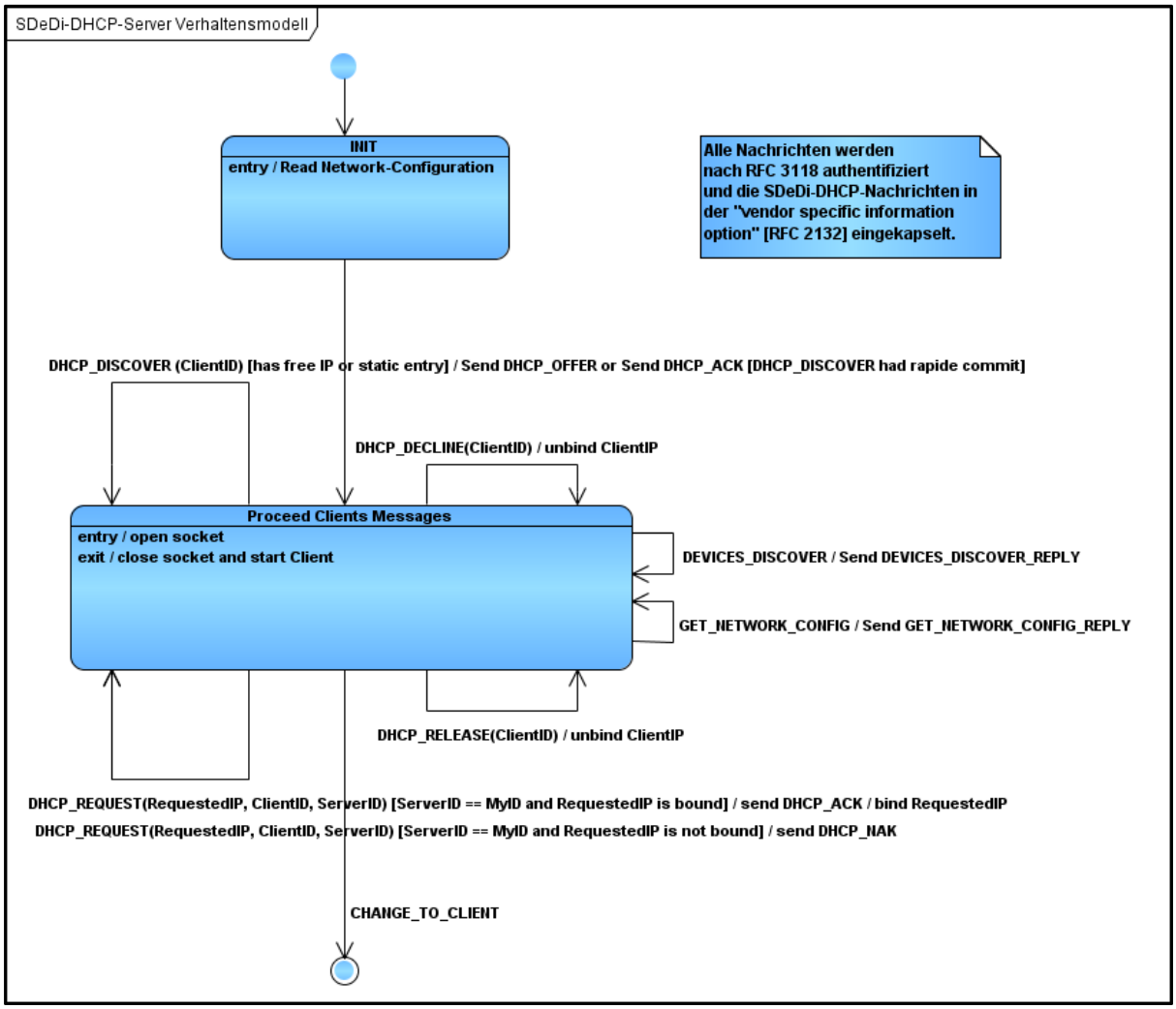


Abbildung 5-7: SDeDi-DHCP-Server Verhaltensmodell [vgl. RFC 2131 und SDeDi-DHCP].

5.4.3 SDeDi-DHCP-Konfigurationstool Verhaltensmodell (automatische Konfiguration)

Das Verhalten des Konfigurationstools bei einer automatischen Inbetriebnahme /Konfiguration wird in diesem Abschnitt beschrieben und mit Abbildung 5-8 dargestellt.

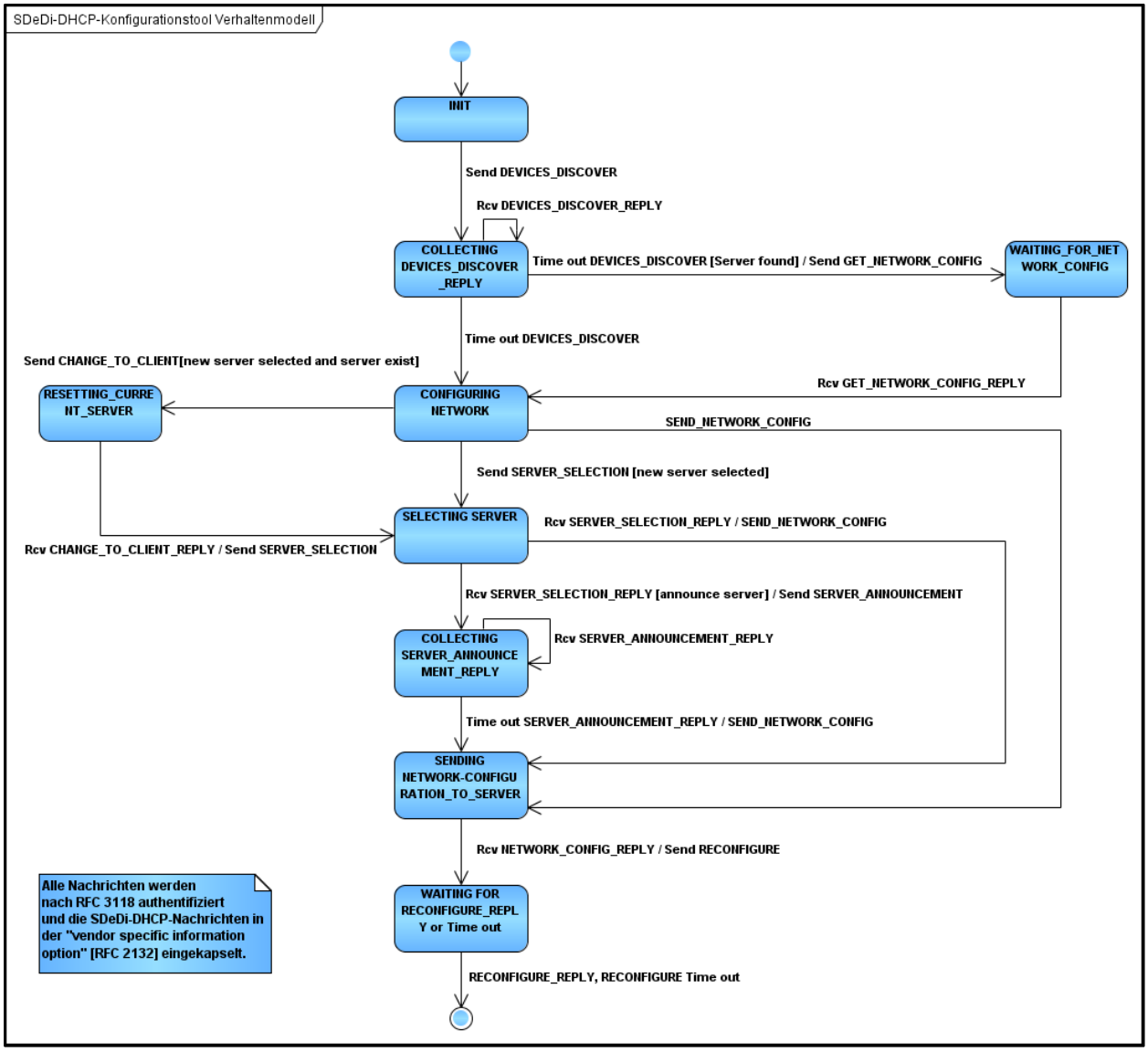


Abbildung 5-8: SDeDi-DHCP-Konfigurationstool Verhaltensmodell [vgl. SDeDi-DHCP].

Voraussetzung zur Durchführung einer automatischen Inbetriebnahme/Konfiguration durch das Konfigurationstool ist das Eintragen des reservierten IP-Adressenbereichs. Der Automat in Abbildung 5-8 wird beim Start einer automatischen Inbetriebnahme/Konfiguration ausgeführt. Dabei wechselt das Konfigurationstool in den INIT-Zustand

und broadcastet ein DEVICES_DISCOVER. Damit startet er den Scan-Vorgang, um alle Geräte aufzufinden. Haben alle Geräte auf das DEVICES_DISCOVER geantwortet oder wenn die MAX_DEVICES_DISCOVER_TIME abgelaufen ist (time out), wird überprüft, ob ein Server gefunden worden ist. Ist dies der Fall, werden die aktuellen Netzwerk-Konfigurationsdaten von ihm gefordert (GET_NETWORK_CONFIG), und es wechselt in den WAITING_FOR_NETWORK_CONFIG-Zustand, wo die Netzwerkkonfigurationsdaten empfangen werden. Nach Empfang der Netzwerkkonfigurationsdaten wird in den CONFIGURING_NETWORK-Zustand gewechselt. In diesem Zustand wird das Netzwerk automatisch konfiguriert.

Netzwerkkonfigurieren ist das Eintragen der Netzwerkeigenschaften sowie der spezifischen Konfigurationsparameter aller Geräte in einer Konfigurationsdatei bzw. Datenbank. Bei einer automatischen Netzwerkkonfiguration werden alle Einträge automatisch vom Konfigurationstool durchgeführt. Voraussetzung ist, dass der IP-Adressenbereich „eingetragen“ und der Server „ausgewählt“ wurde. Das Konfigurationstool führt Nachfolgendes bei einer automatischen Konfiguration zur Erstinbetriebnahme durch: Automatische Ermittlung und Eintrag der Netzwerkeigenschaften, Reservierung der IP-Adressen. Dafür könnte folgendes Beispielszenario betrachtet werden:

1. Reservierung der letzten IP-Adresse des Bereichs für das Konfigurationstool.
2. Reservierung der IP-Adressen der Server-Geräte von der vorletzten Adresse absteigend: Ist der IP-Adressen-Bereich z. B. 192.168.1.100-192.168.1.254, wird für den ersten Server die IP-Adresse 192.168.1.253 reserviert, 192.168.1.252 für den Zweiten usw.
3. Reservierung der IP-Adressen der Client-Geräte von der ersten Adresse aufsteigend: Die 192.168.1.100 wird für das erste Client-Gerät reserviert, die 192.168.1.101 für das zweite usw.

Handelt es sich nicht um eine Erstinbetriebnahme, wird die aktuelle Netzwerkkonfiguration, bis auf die Reservierung der IP-Adressen der neu gefundenen Geräte übernommen. Für die Reservierung der IP-Adresse eines neuen Geräts wird einfach die nächste freie übernommen.

Ist die Netzwerkkonfiguration erfolgreich durchgeführt worden, kann der nächste Zustand ausgeführt werden. Der KONFIGURING_NETWORK-Zustand führt zu drei verschiedenen Zuständen: SELECTING_SERVER, RESETTING_CURRENT_SERVER und SENDING_NETWORK_CONFIG_TO_SERVER.

Wechseln in den SELECTING_SERVER-Zustand: Der CONFIGURING_NETWORK-Zustand führt direkt zu diesem Zustand, meist bei einer Erstinbetriebnahme, da noch kein Server existiert und erst ausgewählt werden muss. Der SELECTING_SERVER-Zustand kann zu zwei verschiedenen Zuständen führen, COLECTING_SERVER_ANNOUCEMENT_REPLY und SEN-

DING_NETWORK_CONFIGURATION_TO_SERVER. Dies hängt davon ab, ob der ausgewählte Server dem Client bekannt gegeben werden soll (optional).

Soll der Server bekannt gegeben werden, wird ein SERVER_ANNOUNCEMENT beim Empfang einer SERVER_SELECTION-Nachricht an alle Clients gesendet und in den COLECTING_SERVER_ANNOUNCEMENT_REPLY-Zustand gewechselt. In diesem Zustand werden die Antworten der Clients gesammelt. Haben alle Clients geantwortet oder wenn der „time out“ zum Empfangen der SERVER_ANNOUNCEMENT_REPLY von Clients abgelaufen ist, werden die Netzwerkkonfigurationsdaten an den Server gesendet und in den SENDING_NETWORK_CONFIGURATION_TO_SERVER-Zustand gewechselt. Hat der Server die Netzwerkkonfigurationsdaten empfangen, sendet er eine NETWORK_CONFIG_REPLY zur Bestätigung. Dann teilt das Konfigurationstool den Clients mit, dass die neuen Konfigurationsparameter beim Server liegen und angefordert werden müssen. Dafür sendet er eine RECONFIGURE-Nachricht an alle Clients und wechselt in den WAITING_FOR_RECONFIGURE_REPLY-Zustand. Empfängt ein Client eine RECONFIGURE-Nachricht, muss er seine neuen Konfigurationsparameter vom Server fordern. Haben alle Clients ihre neuen Konfigurationsparameter anfordern können, sendet der Server dem Konfigurationstool eine RECONFIGURE_REPLY zur Bestätigung. Damit ist die Konfiguration/Inbetriebnahme abgeschlossen.

Soll der Server nicht bekannt gegeben werden, werden mit dem Empfang der SERVER_SELECTION_REPLY die Netzwerkkonfigurationsdaten an ihn gesendet und in den SENDING_NETWORK_CONFIGURATION_TO_SERVER-Zustand gewechselt.

Wechseln in den RESETTING_CURRENT_SERVER-Zustand: Der KONFIGURING_NETWORK-Zustand führt zum RESETTING_CURRENT_SERVER-Zustand, wenn der aktuelle Server gewechselt werden muss. Um den Server zu wechseln, muss zuerst der aktuelle Server zurückgesetzt werden, um inkonsistente Zustände im Netzwerk zu vermeiden. Dazu sendet das Konfigurationstool eine CHANGE_TO_CLIENT-Nachricht an den aktuellen Server und wechselt in den RESETTING_CURRENT_SERVER-Zustand. In diesem Zustand wartet es auf die Bestätigung des aktuellen Servers mit der Nachricht CHANGE_TO_CLIENT_REPLY. Nach Empfang dieser Nachricht kann das Konfigurationstool mit der SERVER_SELECTION Nachricht den neue Server auswählen und in den SELECTING_SERVER-Zustand wechseln.

Wechseln in den SENDING_NETWORK_CONFIGURATION_TO_SERVER-Zustand: Der KONFIGURING_NETWORK-Zustand führt zum SENDING_NETWORK_CONFIGURATION-TO_SERVER-Zustand, wenn ein Server vorhanden ist und kein neuer Server ausgewählt wurde.

6 Implementierung

Dieses Kapitel beschäftigt sich mit der Implementierung der im Kapitel 5 festgelegten Spezifikation des Systems. Das System besteht aus einer verteilten Applikation, die aus einer Embedded- und einer PC-Applikation besteht.

6.1 Embedded-Applikation

Dieser Abschnitt beschreibt die Implementierung der Embedded-Applikation (DHCP-Server und -Client). Sie wurde mit der Programmiersprache C unter eCos auf Basis des LPC2468 OEM Test-Bords entwickelt. Als IDE (Integrated Development Environment) wurde „Eclipse IDE for C/C++, Version 3.3.0“ mit einem Pluging für eCos verwendet.

6.1.1 Überblick Embedded RTOS eCos

eCos steht für „embedded configurable operating system” und ist ein Open Source, ein konfigurierbares, portierbares und freies Embedded Echtzeitbetriebssystem. Nachfolgend werden seine wesentlichen Merkmale beschrieben.

eCos wird als Open-Source-Laufzeitsystem zur Verfügung gestellt und durch das „GNU open source development tool“ unterstützt. Entwickler haben vollen und freien Zugang zu allen Aspekten des Laufzeitsystems. Keine Teile davon sind proprietär oder versteckt, und der Code kann nach eigenen Bedürfnissen frei überprüft, ergänzt und geändert werden. Diese Rechte werden durch die GNU GPL (GNU General Public License) gewährt und geschützt. Eine Ausnahmeklausel ist der eCos Lizenz hinzugefügt worden. Diese ermöglicht, Anwendungen auf der Basis von eCos zu entwickeln, die nicht unbedingt frei sein müssen, vorausgesetzt der Code dieser Anwendungen stammt nicht vom eCos Code.

Eine der wichtigsten technologischen Innovationen in eCos ist die Möglichkeit, das System zu konfigurieren (Systemkonfiguration). Da ermöglicht eCos Entwicklern, ein eigenes applikationsspezifisches Betriebssystem zu erstellen. Dadurch kann die „resource footprint“ von eCos minimal gehalten werden, indem alle unbenötigten Funktionalitäten und Features entfernt werden. Alle diese Merkmale machen eCos geeignet für Embedded-Systeme.

eCos hat eine auf Komponenten basierende Architektur, was möglich macht, die Funktionalitäten von eCos durch Komponenten zu erweitern. Dadurch können die Features von Applikationen durch Laufzeit-Komponenten (optional) erweitert werden.

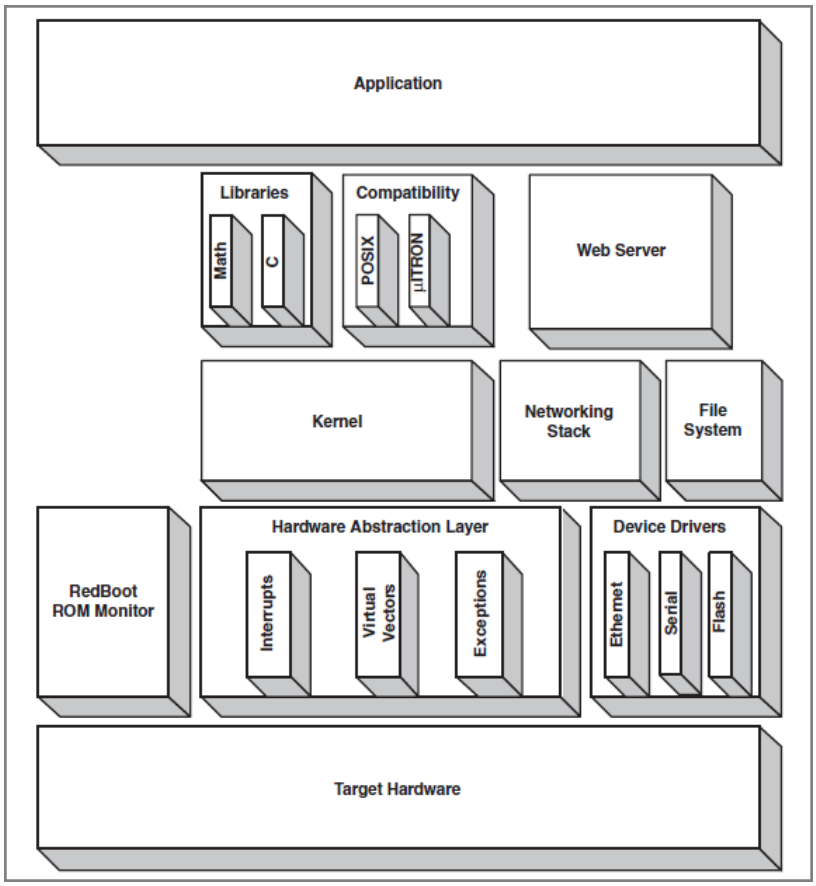


Abbildung 6-1: Beispiel eines Embedded Softwaresystems unter eCos (Massa 2003, S. 9).

eCos wurde so designed, dass es auf eine breite Menge von Zielplattformen bzw. - Architekturen einschließlich 16, 32, 64 bit-Architekturen, MPUs (Microprocessor unit), MCUs (Microcontroller unit) und DSPs (Digital Signal Processor) portiert werden kann.

Nachfolgende Architekturen werden zurzeit von eCos unterstützt: ARM, Intel StrongARM, Xscale, Fujitsu FR-V, Hitachi SH2/3/4, Hitachi H8/300H, Intel x86, MIPS, Matsushita AM3x, Motorola PowerPC, Motorola 68k/Coldfire, NEC V850 und Sun SPARC.

eCos wurde konzipiert, um die Entwicklung von Anwendungen mit Echtzeit-Anforderungen zu unterstützen, indem es Mechanismen wie Thread-Unterbrechbarkeit (preemptability), minimale Interrupt-Wartezeiten (interrupt latencies), Synchronisationsmechanismen, „scheduling policies“ und Interruptbearbeitungsmechanismen (interrupt handling mechanisms) zu Verfügung stellt. eCos stellt Funktionalitäten bereit, die von allgemeinen Embedded Applikationen benötigt werden, einschließlich Gerätetreiber, Speicherverwaltung, Ausnahmebehandlung, C, Mathe-Bibliotheken, TCP-IP Stack usw. (Abbildung 6-1). Außerdem werden neben der Laufzeitunterstützung alle benötigten Tools für die Entwicklung von Embedded-Applikationen zur Verfügung gestellt wie das eCos-Software-Konfiguration- and Build-Tool, der GNU basierte Kompilers, Assemblers, Linker, Debuggers, Simulators, etc. (vgl. eCos, I.2).

6.1.2 Embedded Artists LPC2468-16 OEM Board

Das LCP2468-16 Board wird in Abbildung 6-2 dargestellt und weist nachfolgende Features auf (vgl. Embedded Artists).

- NXP's ARM7TDMI LPC2468 microcontroller in BGA package, with 512 Kbyte program FLASH and 96 Kbyte SRAM
- External FLASH memories: 128 MB NAND FLASH and 4 MB NOR FLASH
- External data memory: 32 MB SDRAM
- 12.0000 MHz crystal for maximum execution speed and standard serial bit rates, including CAN and USB requirements
- 32.768kHz RTC crystal
- 100/10M Ethernet PHY/interface based on Micrel KSZ8001L (16-bit data bus version) and National DP83848 (32-bit data bus version)
- USB-OTG support on USB-A channel via external ISP1301 chip
- 256 Kbit I2C E2PROM for storing non-volatile parameters
- Buffered 16- or 32-bit data bus for external expansion
- Connector: Two 100 pos connector (FX8C-100 from Hirose), 0.6mm pitch
    - All LPC2468 pins available (except a few used for Ethernet and USB-OTG interface)
- uSD/transflash interface and connector
- +3.3V only powering
- Onboard reset generation
- Compact dimension: 80 x 65 mm (16-bit data bus version), 75 x 40 mm (32-bit data bus version)

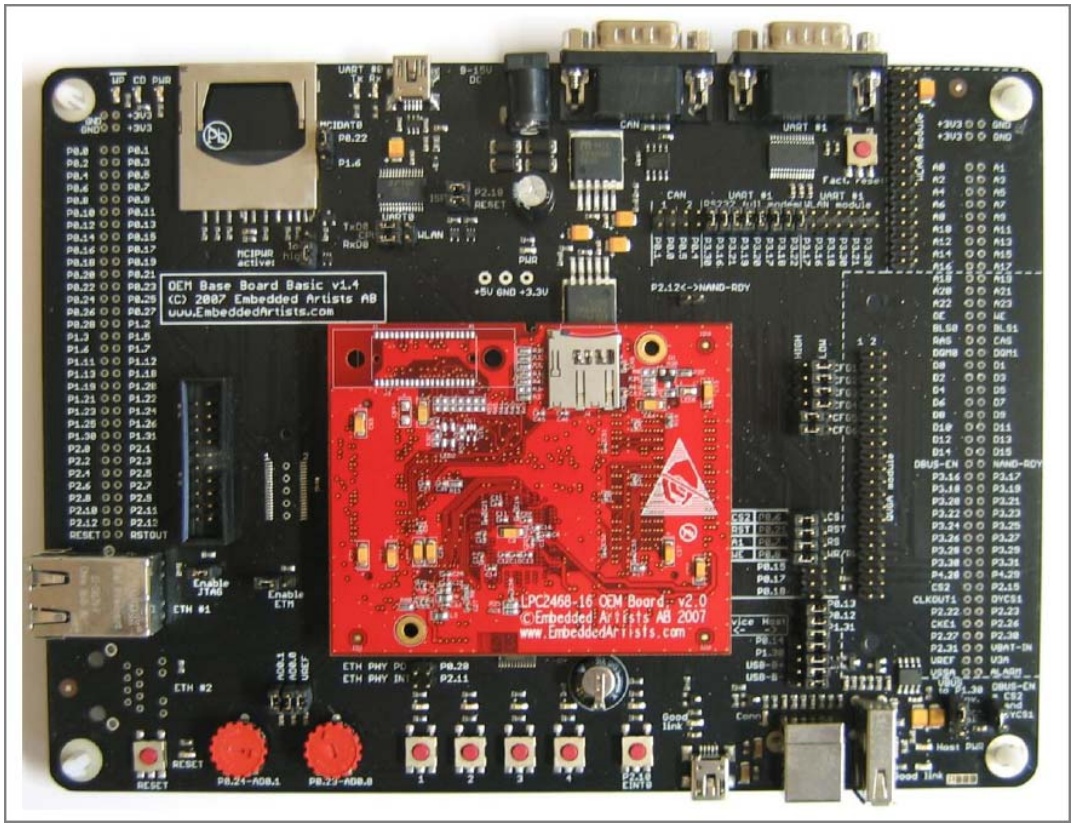


Abbildung 6-2: LPC2468-16 OEM Board (Embedded Artists)

6.1.3 DHCP-Client und -Server

Für die Implementierung des DHCP-Clients und -Servers unter eCos wurde das Standard „udhcp-packet“ unter eCos portiert und entsprechend dem SDeDi-DHCP-Protokoll (Abschnitt 5.3) erweitert. Dafür wurde der Name „SDeDi_udhcp_eCos“ verwendet.

„udhcp“ ist ein Softwarepaket, welches die wesentlichen Funktionalitäten von DHCP-Client und -Server nach RFC2131 implementiert. „udhcp“ wurde so implementiert, dass es für Embedded Systems geeignet ist. „udhcp“ steht unter der GNU GPL lizenziert und ist im busybox-Projekt zu finden (siehe BUSYBOX).

6.1.3.1 Hauptprogrammfluss von SDeDi_udhcp_eCos

Der Hauptprogrammfluss wird in Abbildung 6-3 dargestellt und nachfolgend beschrieben. Das SDeDi_udhcp_eCos startet mit einer Initialisierungsphase, um zu bestimmen in welchem Modus es gestartet werden soll. Es kann sowohl im Client- als auch im Server-Modus starten, und dieser kann auch während der Laufzeit geändert werden.

Start

Die notwendigen Informationen über den Operationsmodus (COMPONENT_TYPE), den Konfigurationszustand (CONFIG_STATE) und die aktuelle Konfiguration der Netzwerkkarte (INTERFACE_CONFIG) eines Gerätes werden in einer Konfigurationsdatei (z. B. …/ SDeDi_udhcp_eCos_if.conf) gespeichert. Bei einem Neustart des Geräts werden zuerst folgende Informationen gelesen, deren Bedeutung hier beschrieben wird.

COMPONENT_TYPE bestimmt, wie die Applikation gestartet werden soll. Ist das COMPONENT_TYPE ein COMPONENT_TYPE _SERVER, wird die Applikation im Server-Modus gestartet. Ist COMPONENT_TYPE ein COMPONENT_TYPE_CLIENT, wird die Applikation im Client-Modus gestartet.

CONFIG_STATE stellt fest, ob die Netzwerkkarte konfiguriert ist. Ist sie konfiguriert, enthält die oben genannte Konfigurationsdatei außerdem die Konfigurationsparameter der Netzwerkkarte (IP-Adresse, Subnetzwerk-Maske, Broadcast Adresse, und Default Gateway oder Router Adresse).

Client-Modus

Wird die Applikation im Client-Modus gestartet, wird der DHCP-Client-Prozess (siehe 5.4.1 DHCP-Client Verhaltensmodelle) gestartet, um in Kontakt mit dem DHCP-Server zu kommen. Dabei werden die Konfigurationsparameter des Geräts angefordert. Ist der Server erreichbar, setzt der Client die vom Server zugewiesenen Konfigurationsparameter ein und wechselt in den konfigurierten Zustand. Ist der Server jedoch nicht erreichbar, überprüft er, ob die lokal gespeicherten Konfigurationsparameter noch gültig sind (IP-Adresse steht nicht im Konflikt). Ist dies der Fall, setzt er diese ein und gibt eine Warnung aus, dass der Server nicht erreicht werden kann und wechselt in den konfigurierten Zustand. Sind die lokalen Konfigurationsparameter jedoch nicht gültig, wechselt er in den nicht konfigurierten Zustand.

Bekommt der Client die Nachricht „RECONFIGURE“ vom Konfigurationstool, ist das ein Signal, dass der Server wieder erreichbar ist, bzw. neue Konfigurationen beim Server liegen. Dann startet er den Client-DHCP-Prozess wieder. Ist der Prozess erfolgreich durchgeführt worden, wechselt er in den konfigurierten Zustand.

Bekommt die Applikation die Nachricht SERVER_SELECTION, speichert sie alle notwendigen Informationen, die für den Server-Modus notwendig sind, und wechselt zum Server-Modus.

Server-Modus

Wird die Applikation im Server-Modus gestartet, liest sie die NIC-Konfiguration aus der SDeDi_udhcp_eCos_if.conf-Datei. Im Server-Modus verhält sich die Applikation wie ein SDeDi-DHCP-Server (5.4.2 SDeDi-DHCP-Server Verhaltensmodell). Erhält die Applikation in

diesem Modus die Nachricht CHANGE_TO_CLIENT, setzt sie Ihre aktuelle Konfiguration zurück, wechselt in den Client-Modus und wartet auf eine RECONFIGURE Nachricht.

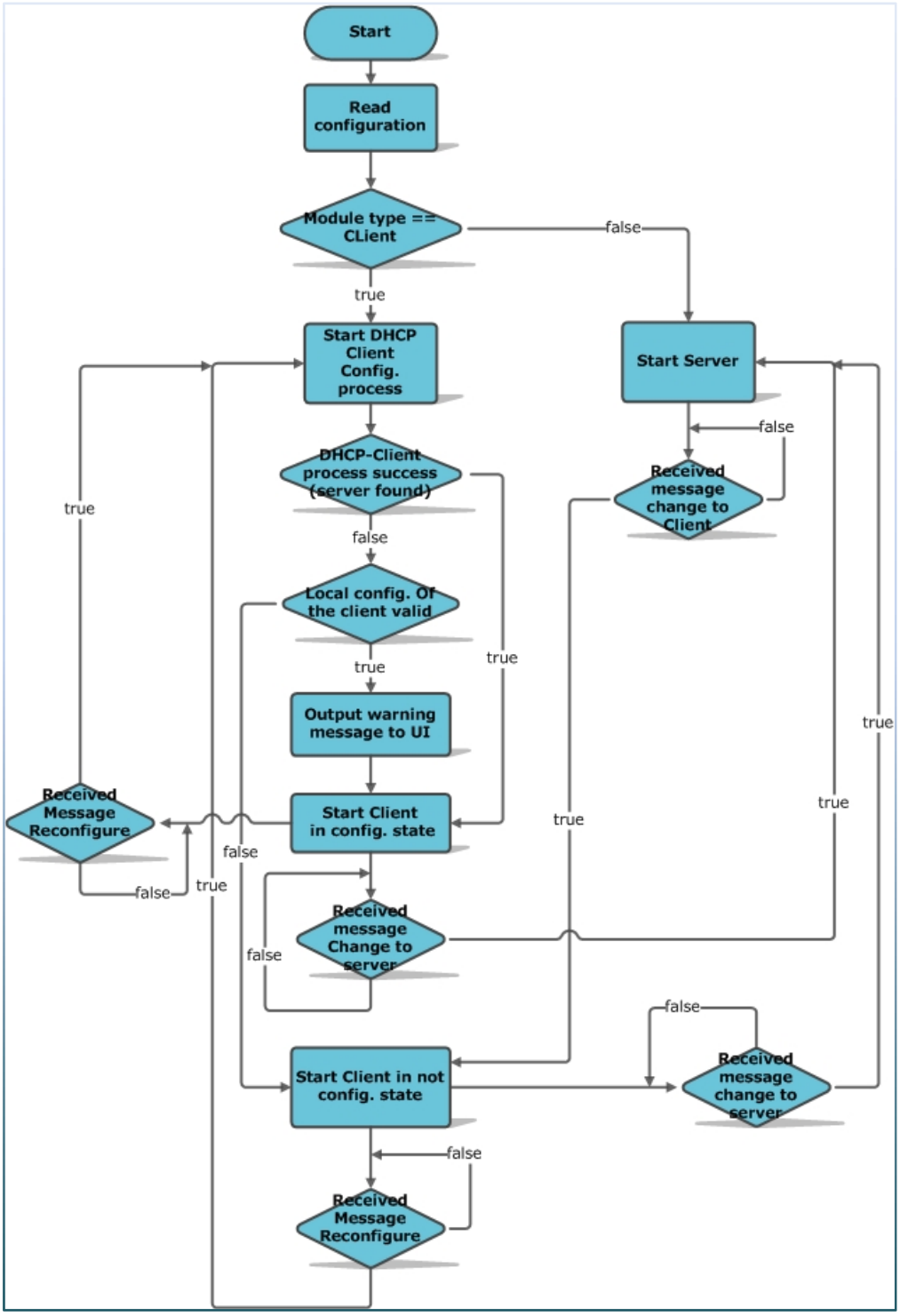


Abbildung 6-3: SDeDi_udhcp_eCos Hauptprogramm Flussdiagramm

6.2 PC-Applikation

In diesem Abschnitt wird die PC-Applikation (DHCP-Konfigurationstool) beschrieben. Sie wurde mit der objektorientierten Programmiersprache Visual c# unter dem .Net Framework 3.5 unter Visual Studio 2008 entwickelt.

6.2.1 Überblick Microsoft .Net und Visual c#

Microsoft .NET ist eine Plattform, die die notwendigen Werkzeuge und Technologien bereitstellt, um Anwendungen für Microsoft Windows und andere Plattformen wie mobile Geräte (Handys, PDAs etc.) oder Web-Server zu entwickeln. Visual c# ist eine der von Microsoft entwickelten Programmiersprache zur Erstellung dieser Anwendungen. Diese Anwendungen erlauben die Kommunikation mit vielen (entsprechend ausgerüsteten) Geräten wie Windows-Rechnern, Handys, PDAs etc. Letztendlich bietet Microsoft .NET eine Architektur, die sich aus verschiedenen Bausteinen zusammensetzt.

Visual Studio 2008 ist die neueste Microsoft-Entwicklerplattform, mit der sich Anwendungen in verschiedenen Sprachen (Visual Basic, Visual C#, C++ etc.) erstellen lassen. Visual Studio 2008 stellt eine recht mächtige Entwicklungsumgebung dar, die von der Erstellung des Quellcodes über das Testen des Codes und die Anbindung an Datenbanken bis hin zur Erstellung von Anwendungspaketen alle Schritte unterstützt.

Das .NET Framework (Abbildung 6-4) stellt die Infrastruktur für .Net Anwendungen dar. Es besteht aus: Der CLR (Common Language Runtime), die die Laufzeitumgebung, in der .Net-Anwendungen ausgeführt werden.

Die „Klassenbibliothek“ beinhaltet die gesamte Funktionalität des .Net-Frameworks, somit auch die von Visual c#. C# bringt, wie alle anderen .Net Sprachen, lediglich die Syntax mit. Diese „Klassenbibliothek“ wird in sogenannte Namespaces aufgeteilt und bietet Funktionen zum Zugriff auf das Betriebssystem (z. B. Windows), auf das Dateisystem, auf den Netzwerk-Stack und vieles mehr.

Das UI (User Interface), welches der Entwicklung von Benutzeroberflächen dient, wird in Web Form (Web UI, unter ASP.Net bekannt), Windows Form (Desktop UI) und Windows Presentation Foundation (Web UI und Desktop UI) aufgeteilt.

Die CLS (Common Language Specification) und das CTS (Common Type System) stellen die Basis von .Net Sprachunabhängigkeit dar. Die Sprachen des .Net Frameworks müssen bestimmten Regeln folgen, die sicherstellen, dass Codes verschiedener Sprachen kombiniert werden können.

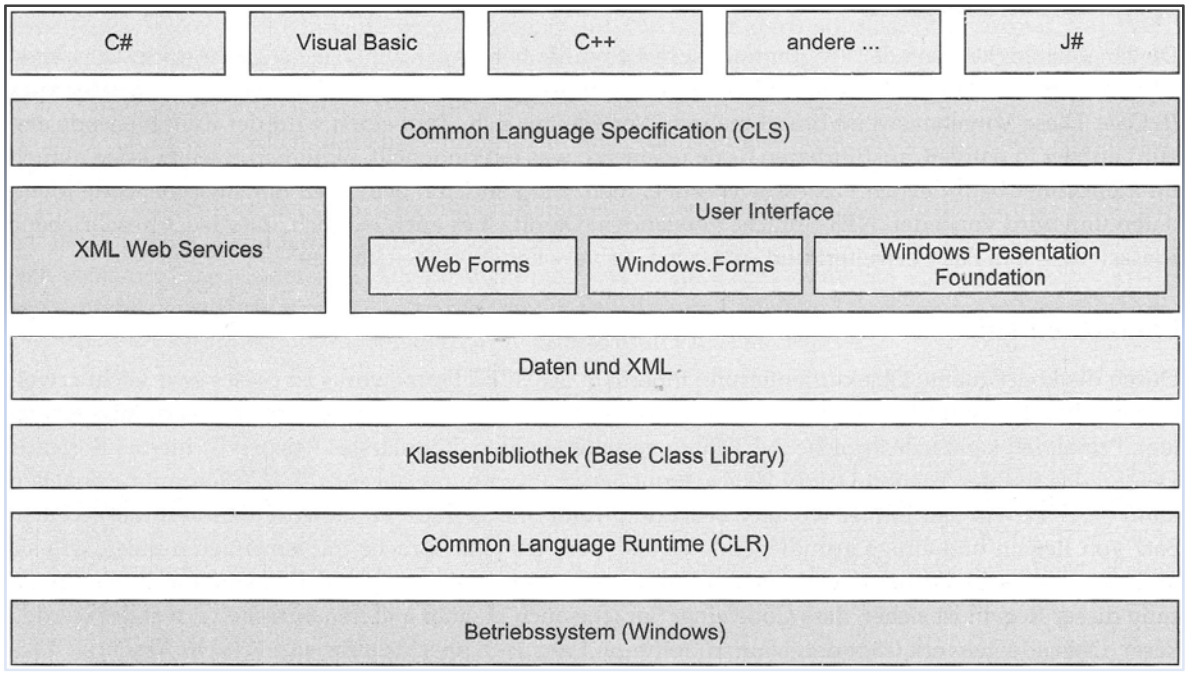


Abbildung 6-4: Übersicht Aufbau des .Net Frameworks (Eller 2008, S. 36)

Zu erwähnen ist, dass die Konzeption der .NET-Architektur nicht ausschließlich auf Windows abgestimmt wurde. .NET-Anwendungen können auf weiteren Geräten wie der XBox, auf Mobiltelefonen, auf PDAs etc. laufen. Microsoft entwickelt für die unterschiedlichen Geräte entsprechende Clients (z. B. .NET Framework für Windows, .NET Compact Framework als Untermenge des .NET Framework für Embedded Windows XP etc.). Darüber hinaus können .Net Anwendungen teilweise dank Projekten wie mono auf „unixoide"-Betriebssystemen laufen (vgl. Eller 2008, S. 36-40).

6.2.2 Konfigurationstool

In diesem Abschnitt werden die Oberflächen und das Klassendiagramm des Konfigurationstools beschrieben und dargestellt.

6.2.2.1 Konfigurationstool Oberfläche (Easy-Modus)

Im Easy-Modus (Abbildung 6-5), besteht die Konfigurationstool-Oberfläche nur aus drei Textboxen und einem Button. Zwei Textboxen (IP-Start und IP-End) für das Eintragen des reservierten IP-Bereichs und eine Textbox (Server ID) für das Eintragen der Server-ID. Die ID des Servers kann z. B. die MAC-Adresse oder die Seriennummer des Server-Geräts sein. Nachdem die benötigten Informationen eingetragen worden sind, kann auf den „Start

Auto Config“-Button geklickt werden. Nach dem Klick wird das Netzwerk automatisch in Betrieb genommen/konfiguriert (Automat in 5.4.3 wird ausgeführt). Es kann jederzeit in den Profi-Modus gewechselt werden, um fortgeschrittene Änderungen vorzunehmen. Dieses Ergebnis zeigt, dass die Konfiguration erheblich einfacher geworden ist: „Mit nur einem Klick“!

Abbildung 6-5: Konfigurationstool Oberfläche (Easy-Modus)

6.2.2.2 Konfigurationstool Oberfläche (Profi-Modus)

Im Profi-Modus (Abbildung 6-6), besteht das Konfigurationstool aus: einer Toolbar mit zwei ToolStripButtons, “Scan Network” und “Start Client Config”, sie dienen jeweils zum Scannen des Netzwerks und zum Starten der Client-Konfiguration.

Ein TreeView (Configuration Explorer) dient nicht nur zum Navigieren durch die aktuelle Konfiguration (Network Configuration), sondern zeigt auch das Scan-Ergebnis (Network Scan Result). Nach der Auswahl eines Elements aus dem „Configuration Explorer“ wird die damit verbundene View (Anzeige oben links) angezeigt. Wird z. B. das „Network Scan Result“-Element ausgewählt, wird die View zum Anzeigen des Ergebnisses des Netzwerk-Scans aktiviert, usw.

Ein Output-Fenster (Output) dient zur Anzeige der aktuellen Aktion, deren Beendigung und es gibt Hinweise zur Durchführung der nächten Aktion.

Ein Message-Logs-Fenster (Message Logs) zeigt alle gesendeten und empfangenen (Broadcast-) Nachrichten an.

In diesem Modus können alle weiterführenden Konfigurationen vorgenommen werden. Diese sind z. B. Wechseln der Server, Löschen und Hinzufügen von Clients, Löschen und Hinzufügen von DHCP-Optionen, etc. Dabei werden keine manuellen Einträge gemacht. Alle Informationen (IP- Adressen, MAC-Adressen, etc.) für die Konfiguration stehen zur Verfügung und werden nur per Mausklick ausgewählt.

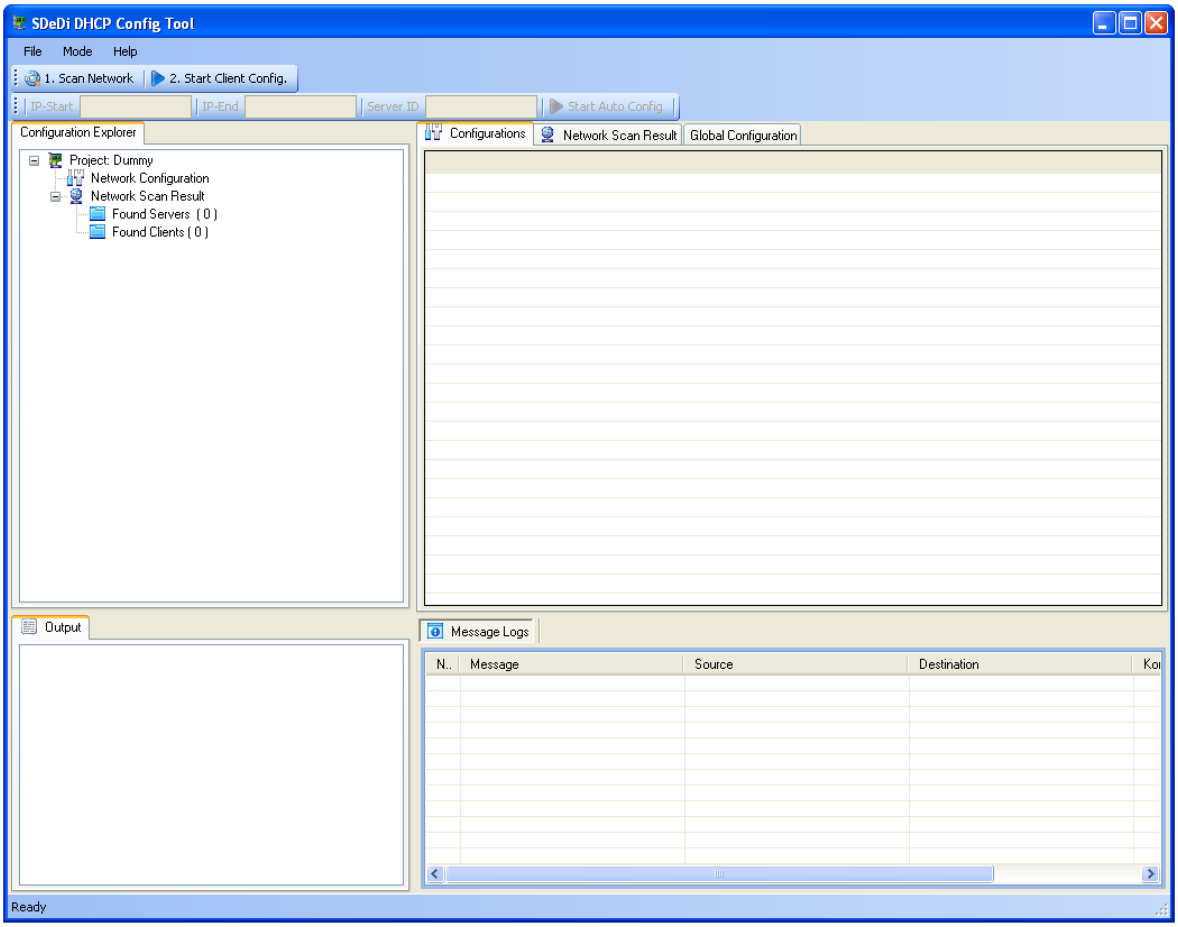


Abbildung 6-6: DHCP-Konfigurationstool (Profi-Modus)

6.2.2.3 DHCP-Konfigurationstool Klassendiagramm

Wie in Abbildung 6-7 zu sehen ist, hat eine DHCP-Konfiguratiostool-Klasse nachfolgende Eigenschaften: Den aktuellen DHCP-Server (currentDHCPServer), eine Liste der gefundenen Server (foundServers), eine Liste der gefundenen Clients (foundClients), eine Klasse zum Empfangen, Versenden und Abhören von UDP Nachrichten (MyUDPSocket), ein Thread zum Abhören von UDP-Verbindungen, einen Timer zur Bestimmung der Abhörzeit und ein configToolState zur Speicherung des aktuellen Zustands des Konfigurationstools.

Eine Gerät-Klasse stellt ein Gerät dar und hat als Eigenschaften die Parameter zur Konfiguration seiner Netzwerkschnittstelle, nämlich Namen (name), physikalische IP-Adresse (physicalAddress), IP-Adresse des Subnetzwerks des Geräts usw.

Eine DHCPClient-Klasse stellt ein Client-Gerät dar. Eine DHCPServer-Klasse stellt ein Server-Gerät dar und hat im Gegensatz zu einer DHCPClient-Klasse eine zusätzliche Eigenschaft, welche die Konfigurationsdaten des Netzwerks sind (networkConfigurationData).

Eine NetworkKonfigurationData-Klasse stellt die Konfiguration des Netzwerks dar. Sie hat als Eigenschaften den Dateipfad der Konfigurationsdaten (filePath), den Namen der Netzwerkschnittstelle des Servers zum Empfangen der Client-Anfragen (interfaceName - z. B. „eth0“-), ein Subnetz und eine Liste aller globalen Optionen.

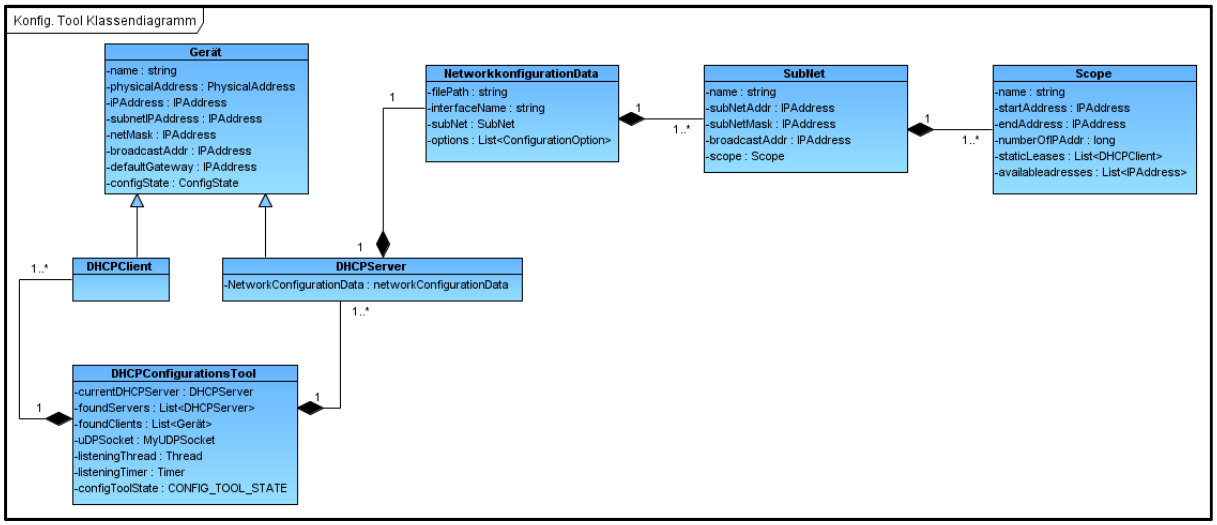


Abbildung 6-7: DHCP-Konfigurationstool Klassendiagramm

Die SubNet-Klasse stellt die Konfiguration eines Subnetzwerks dar. Ihre Eigenschaften sind Name (name) und IP-Adresse des Subnetzwerks (SubnetAddr), die Subnetz-Maske, die Broadcast-Adresse und der reservierte IP-Bereich (scope). In einem Subnetz können ein oder mehrere scopes definiert werden. Hier wird nur ein benutzt.

Die Scope-Klasse stellt einen reservierten IP-Bereich in einem Subnetzwerk dar. Ihre Eigenschaften sind der Name des scopes (name), die Anfang- und Start-IP-Adresse, die Anzahl der IP-Adressen, die in diesem vergeben werden können, eine Liste von schon vergebenen IP-Adressen (staticLeases) und eine Liste der zu vergebenen IP-Adressen (availableAdresses).

7 Test

Nachfolgende Tests sind erfolgreich durchgeführt worden.

- Automatische Inbetriebnahme des Netzwerks mit einem Server und fünf Clients. Dabei sind alle Geräte mit einer DEVICES_DISCOVER-Nachricht in weniger als 2s aufgefunden worden.
- Integration von neuen Geräten ins Netzwerk.
- Adressenkonflikte-Erkennung und -Auflösung.
- Erreichbarkeit der in Betrieb genommenen Geräte. Dies ist durch „anpingen“ der Geräte mit den neu konfigurierten IP-Adressen erfolgt.
- Kommunikation zwischen den Geräten. Die erfolgte durch das Web-Interfache der Geräte, mit der man andere Geräte „anpingen“ könnte.
- Erkennung des Ausfalls eines konfigurierten Geräts.
- Korrekte Funktionalität von Broadcast-Nachrichten. Obwohl ungekannte Geräte auf die gesendeten Nachrichten reagiert haben, sind nur Nachrichten von bekannten Geräten bearbeitet worden. Dies ist anhand des DHCP-Servers des Subnetz-Routers getestet worden.

Dies sind Tests, die normalerweise automatisch während der Entwicklung durchgeführt werden, aber nicht ausreichend zum Testen des Systems sind. Man sollte dafür eher systematische Testansätze durchführen. Es sollten zuerst einzelne Komponenten getestet werden, indem z. B. „Unit Test“ bzw. „Testsuite“ für jede Komponente anhand entsprechender Tools geschrieben und durchgeführt werden. Anschließend sollte ein Integrationstest gemacht werden. Da man sich in einem „shared-subnet“ befindet, ist das Testen des Systems mit seiner Umgebung sehr wichtig. Hierfür können „Etikal Hacker“-Verfahren wie z. B. ein „Denial of Service“-Angriffsversuch eingesetzt werden. Angesichts des Aufwands eines systematischen Testverfahrens, ist es im Rahmen dieser Arbeit leider zeitlich nicht möglich gewesen, dies durchzuführen.

8 Fazit und Ausblick

Das Ziel, ein System für die automatische Inbetriebnahme/Konfiguration des BMZ-TCP/IP-Netzwerks zu entwickeln wurde erreicht. Da das Ergebnis die Erwartungen der Fa. detectomat erfüllt hat, wird das System eingesetzt.

8.1 Fazit

In der Einführung wurde die Aufgabenstellung beschrieben. Diese beinhaltet nicht nur die Hintergründe der Arbeit, sondern auch die Anforderungen des zu entwickelnden Systems aus Sicht des Kunden. Anschließend wurden die Ziele der Arbeit gesetzt. Der Zielsetzung folgte die Motivation. Vor der Definition der Anforderungen an das System, wurden bereits existierende Lösungen zur Inbetriebnahme/Konfiguration von TCP/IP Netzen vorgestellt.

Zur Definition der Anforderungen wurden zunächst wichtige Informationen über das System gegeben, nämlich der vorausgesetzte Zustand, in dem sich die Geräte vor einer Inbetriebnahme/Konfiguration befinden müssen, die Beschreibung der Umgebung („shared-subnet“) des Systems, die Definition der wichtigen Begriffe und die Spezifikation der Systemprozesse. Die Spezifikation der Systemprozesse wurde anhand von Use-Cases und deren tabellarischer Beschreibungen dargelegt. Diese stellten die Grundlagen dar zur Festlegung der Systembeteiligten und Systemkomponenten, sowie zur Definition der funktionalen und nicht funktionalen Anforderungen. Zur Erfüllung einiger davon wurden Protokolle benötigt.

Von der IETF sind bereits mehrere Protokolle in RFCs definiert worden, die zur Inbetriebnahme/Konfiguration von TCP/IP-Netzen eingesetzt werden können. Besonders schwierig bei der Untersuchung und Auswahl war das Lesen der RFCs, da sie schwer zu verstehen sind. Um die beste Wahl angesichts der gestellten Anforderungen treffen zu können, wurden zuerst alle potenziellen Protokolle untersucht. Dabei wurden Protokolle mit gleicher Funktionalität gegenübergestellt und anschließend eine Auswahl getroffen. Als Basis-Protokoll für die automatische Inbetriebnahme/Konfiguration wurde DHCP, für die Sicherheit DHCP-Sicherheit, für die automatische Ermittlung der Netzwerk-Eigenschaften ICMP und für die Erkennung von Adressenkonflikten Ping und ARP ausgewählt.

Im Entwurf wurde zuerst die Architektur des Systems festgelegt. Sie besteht aus zwei verteilte Applikationen: Eine PC-Applikation (DHCP-Konfigurationstool) und eine Embedded Applikation (DHCP-Server und -Client). Danach wurde für die Spezifikationen der Komponenten des Systems die DHCP-Spezifikation vollständig übernommen und im Rahmen dieser Arbeit erweitert zum SDeDi-DHCP. Die Erweiterungen ermöglichten DHCP, die gestellten Anforderungen für die automatische Inbetriebnahme/Konfiguration des Netzwerks vollständig zu erfüllen.

Die Implementierung der Embedded-Applikation erfolgte mit der Programmiersprache c unter eCos auf Basis des Bords „LPC2468-16“. Die Implementierung der PC-Applikation wurde mit der objektorientierten Programmiersprache c# des .Net Frameworks unter Visual Studio 2008 realisiert. Besonders zeitintensiv und komplex war die Implementierung der Embedded-Applikation, da eCos ein „neues“ Betriebssystem war. Besonders zu erwähnen ist das rudimentäre Filesystem (ROMFS), das in der Entwicklungszeit zur Verfügung stand. Mittlerweile gibt es ein besseres, das JFFS (Journalling Flash File System). Obwohl eCos ein Linux-ähnliches OS ist, hat es seine Besonderheiten, die beachtet werden müssen. Es kann z. B. unter eCos kein RAW-Socket angelegt werden, was unter Linux kein Problem darstellt.

Das Netzwerk konnte erfolgreich automatisch in Betrieb genommen werden mit einem Server und fünf Clients. Weitere Funktionalitäten wie Adressenkonflikterkennung, Erreichbarkeit der Geräte, Kommunikation zwischen den Geräten, etc. konnten erfolgreich getestet werden. Das Testen dieser Funktionalitäten war zwar wichtig, aber für ein „vollständiges“ Testen des Systems wäre noch der Einsatz systematischer Testverfahren erforderlich. Das Konnte aufgrund des Aufwands im Rahmen dieser Arbeit leider zeitlich nicht mehr realisiert werden.

8.2 Ausblick

Bevor das System zum Einsatz kommt, sollten noch einige wichtige Faktoren beachtet werden. Es wurde hier mehr Wert auf die funktionalen Anforderungen des Systems gelegt. Die nicht funktionalen Anforderungen, sind aber auch von großer Bedeutung, besonders wenn es sich um ein Echtzeitsystem der Sicherheitstechnik handelt. Zu betrachten sind z. B. nicht funktionale Anforderungen wie die Verfügbarkeit der Server, die Zuverlässigkeit der Nachrichten, die Effizienz des Konzepts und der Implementierungen, etc. Es wurde bereits ein Konzept für die Sicherheit entworfen, aber keines für die Schlüssel-Verwaltung. Was eine sehr wichtige Rolle für die Sicherheit spielt. Wie bereits erwähnt, sollte das System systematisch getestet werden.

Ein weiterer Ausblick wäre der Einsatz des Systems in IPv6-Netzen. Dafür werden die ausgewählten Protokolle durch entsprechende IPv6-Protokolle ersetzt. DHCP wird DHCPv6, ARP wird NDP (Neighbor Discovery Protocol), ICMP wird ICMPv6, etc. Das im Rahmen dieser Arbeit entwickelte Protokoll SDeDi-DHCP wird beibehalten.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

Literatur

Badach 2007 — BADACH, Anatol; HOFFMAN, Erwin: Technik der IP-Netze: Funktionsweise Protokolle und Dienste. München : Hanser, 2007. -ISBN 978-3-446-21935-9

Batroff 2007 — BATROFF, Philipp : Entwicklung eines Netconf Interface zur Verwaltung von SNMP gemanagten Netzelementen. Fachhochschule Köln; Fakultät für Informations-, Medien- und Elektrotechnik, Studiengang Information Engineering; Bachelorarbeit.

Eller 2008 — ELLER, Frank: Visual C# 2008 : Grundlagen, Programmiertechniken, Datenbanken ; [Windows-Programmierung mit WPF und Windows Forms ; O/R-Mapping mit LINQ to SQL]. München: Addison-Wesley, 2008. -ISBN 978-3-8273-2641-6

EN 54-1 1996 — EN 54-1 1996: Brandmeldeanlagen Teil 1: Einleitung.

Hruschka 2002 — HRUSCHKA, Peter ; RUPP, Chris: Agile Softwareentwicklung für embedded real-time systems mit der UML. München : Hanser, 2002. -ISBN 3-446-21997-8

Massa 2003 — MASSA, Anthony J.: Embedded software development with eCos. Upper Saddle River, NJ : Prentice Hall Professional Technical Reference, 2003. -ISBN 0-13-035473-2

Richard 2003 — RICHARD, Blum: C# Network Programming. Alameda: SYBEX, 2003. -ISBN 0-7821-4176-5

Walter 2007 — WALTER, Doberenz; THOMAS, Gewinnus: Visual C# 2005 : Kochbuch. München: Hanser, 2007. -ISBN 3-446-40652-2

Will 2008 — WILL, Panek; TYLOR, Wentworth; JAMES, Chellis: MCTS Windows Server® 2008 Network Infrastructure Configuration. Indianapolis: Wiley Publishing, 2008. -ISBN 978-0-470-26169-9

Request for Comments and Drafts

draft-ietf-dhc-failover-12	DHCP Failover Protocol
RFC 1321	The MD5 Message-Digest Algorithm
RFC 2104	HMAC: Keyed-Hashing for Message Authentication
RFC 2131	Dynamic Host Configuration Protocol
RFC 2132	DHCP Options and BOOTP Vendor Extensions
RFC 3046	DHCP Relay Agent Information Option
RFC 3118	Authentication for DHCP Messages
RFC 3315	Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
RFC 3330	Special-Use IPv4 Addresses
RFC 3927	Dynamic Configuration of IPv4 Link-Local Addresses
RFC 4039	Rapid Commit Option for the Dynamic Host Configuration Protocol version 4 (DHCPv4)
RFC 4741	NETCONF Configuration Protocol
RFC 4742	Using the NETCONF Configuration Protocol over Secure SHell (SSH)
RFC 5227	IPv4 Address Conflict Detection
RFC 792	INTERNET CONTROL MESSAGE PROTOCOL
RFC 826	An Ethernet Address Resolution Protocol
RFC 903	A Reverse Address Resolution Protocol

Internet Links

BUSYBOX — BUSYBOX: BusyBox project
http://busybox.net
Abgerufen am 15.03.2009

Echelon — Echelon: IP-852 Channel User's Guide.
http://www.echelon.com/support/documentation/manuals/cis/078-0312-01A.pdf
Abgerufen am 03.03.2009

eCos — eCos: eCos User Guide
http://ecos.sourceware.org/docs-3.0/user-guide/ecos-user-guide.html
Abgerufen am 21.03.2009

Embedded Artists — Embedded Artists: LPC2468 OEM Board
http://www.embeddedartists.com/products/uclinux/oem_lpc2468.php?PHPSESSID=4d01e4c9b76
Abgerufen am 12.02.2009

FIT-IT — FIT-IT: FIT-IT Embedded Systems.
http://www.fit-it.at/index.html
Abgerufen am 18.04.2009

fping — fping: fping
http://fping.sourceforge.net/
Abgerufen am 25.05.2009

IANA — IANA: ICMP TYPE NUMBERS
http://www.iana.org/assignments/icmp-parameters
Abgerufen am 18.04.2009

ISC — ISC: ISC-DHCP.
https://www.isc.org/software/dhcp
Abgerufen am 02.03.2009

Wikipedia1 — Wikipedia: Simple Network Management Protocol
http://de.wikipedia.org/wiki/SNMP
Abgerufen am 10.04.2009

Wikipedia2 — Wikipedia: Address Resolution Protocol
http://en.wikipedia.org/wiki/Address_Resolution_Protocol
Abgerufen am 13.04.2009

GLOSSAR

Brandmelderzentrale

Eine Brandmelderzentrale (BMZ) ist der Hauptbestandteil eines Brandmeldesystems, durch die andere Bestandteile mit Energie versorgt werden können (z. B. Melder) und die dazu dient die Signale der angeschlossenen Melder aufzunehmen und festzustellen, ob diese Signale eine Brandmeldung bedeuten, diese akustisch und optisch, sowie den Ort der Gefahr anzuzeigen und möglichst jede dieser Informationen zu registrieren. Sie überwacht das Brandmeldesystem auf ordnungsgemäße Funktion, gibt sichtbare und hörbare Anzeigen bei jeder Störung (z. B. Kurzschluss, Drahtbruch der Primärleitung oder Störungen in der Stromversorgung), leitet wenn gefordert, die Brandmeldesignale weiter, z. B. zu Alarmierungseinrichtungen für hörbare oder sichtbare Alarmsignale, zur Übertragungseinrichtung für Brandmeldungen der Feuerwehr, zur Steuereinrichtung für automatische Brandschutzeinrichtungen oder zu einer automatischen Feuerlöschanlage (vgl. EN 54-1 1996, S. 4).

Client

Applikation, die die Konfigurationsparameter der Netzwerkschnittstellen eines Geräts vom Server anfordert und die empfangenen Konfigurationsparameter entsprechend einstellt.

Client-Konfigurieren

Vorgang, bei dem der Server die vom Client geforderten Konfigurationsparameter aus der Konfigurationsdatei ausliest und dem Client sendet.

Errichter

Eine Person, die für die Inbetriebnahme/Konfiguration des Netzwerks zuständig ist, also auch für die Verwaltung des reservierten IP-Bereichs.

Konfigurationsparameter

Alle Informationen, die zur Konfiguration der Netzwerkschnittstelle eines Geräts benötigt werden. Sie bestehen aus den allgemeinen Netzwerkeigenschaften und den spezifischen Konfigurationsparametern des Geräts (IP-

	Adresse, MAC-Adresse, Name, …).
Konfigurationsserver (Server)	Applikation, die die Netzwerkkonfiguration speichert und verwaltet. Eine weitere Funktionalität des Konfigurationsservers ist die Client-Konfiguration.
Netzwerk konfigurieren	Eintragen der Netzwerkeigenschaften sowie der spezifischen Konfigurationsparameter aller Geräte in einer Konfigurationsdatei bzw. Datenbank
Netzwerkadministrator	Eine Person, die für die Bereitstellung des für die zu konfigurierenden Geräte reservierten IP-Bereichs (scope) sowie die damit gebundenen Netzwerkeigenschaften zuständig ist. Er stellt außerdem sicher, dass keine IP-Adresse aus diesem Bereich einem anderen Gerät zugewiesen wird. Er ist zwar zuständig für das gesamte Netzwerk, hat aber keinen direkten Zugriff auf den reservierten IP-Bereich.
Netzwerkeigenschaften	Allgemeine Informationen (z. B.: scope (IP-Bereich), Subnet-Mask, Gateways, DNS-Server, Domain-Name …) über das Netzwerk bzw. Subnetzwerk, die für alle Geräte gelten.
Netzwerkkonfigurationsdaten	Alle Informationen über die Konfiguration des Netzwerks (Netzwerkeigenschaften und gerätespezifische Konfigurationsparameter), die sich in einer Konfigurationsdatei bzw. Datenbank befinden.

Abkürzungsverzeichnis

API	Application Programming Interface
ARP	Address Resolution Protocol
BMS	Brandmeldesystem
BMZ	Brandmeldezentrale
CAN	Controller Area Network
CLI	Command Line Interface
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DSP	Digital Signal Processor
eCos	Embedded Configurable Operating System
FQDN	Fully Qualified Domain Name
FTP	File Transfer Protocol
GPL	GNU General Public License
GUI	Graphical User Interface
HMAC oder KHMAC	keyed-Hash Message Authentication Code
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISC	Internet Systems Consortium
LAN	Local Area Network
LON	Local Operating Network
MAC1	Media Access Control
MAC2	Message Authentification Code
MCTS	Microsoft Certified Technology Specialist
MCU	Microcontroller unit

MIB	Management Information Base
MPU	Microprocessor unit
MTU	Maximum Transmission Unit
PC	Personal Computer
PDA	Personal Digital Assistant
RARP	Reverse Address Resolution Protocol
RDM	Replay Detection Method
RFC	Request for Comments
RPC	Remote Procedure Call
RTOS	Real Time Operating System
SDeDi-DHCP	Secure Device Discovery Dynamic Host Configuration Protocol
SNMP	Simple Network Management Protocol
SSH	Secure Shell
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UI	User Interface
WAN	Wide Area Network
WLAN	Wireless Local Area Network
XML	Extensible Markup Language
YaST	Yet another Setup Tool

Versicherung über die Selbständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §22(4) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 30. Juni 2009 _____

Ort, Datum

_____

Unterschrift