INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Bachelorarbeit

LAN-Beacon
Ein Protokoll zur authentifizierten Selbstbeschreibung lokaler Netze

Dominik Bitzer

INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Bachelorarbeit

LAN-Beacon
Ein Protokoll zur authentifizierten Selbstbeschreibung lokaler Netze

Dominik Bitzer

Aufgabensteller:	Dr. Vitalian Danciu
Betreuer:	Dr. Vitalian Danciu Tobias Guggemos Annette Kostelezky
Abgabetermin:	26. Juli 2017

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 26. Juli 2017

.....
(Unterschrift des Kandidaten)

Kurzfassung

Lokale physische Netze können in virtuelle Netze aufgeteilt werden, sogenannte VLANs. Hiermit kann der Verkehr von mehreren, an dem gleichen Switch angeschlossenen Geräten auf der Sicherungsschicht getrennt werden. Auf einem Anschluss können ein portbasiertes und eine Vielzahl von tagged VLANs transportiert werden.

Hieraus ergibt sich das Problem, dass man durch bloße Betrachtung der physischen Netztopologie nicht mehr eindeutig auf die an einem Anschluss transportierten Netze schließen kann. Verschiedene Ansätze zur Identifikation existieren, diese sind allerdings teilweise nicht einfach anzuwenden oder schlicht unzuverlässig. Zusätzlich bieten sämtliche dieser Verfahren keine Möglichkeit zur Authentifizierung an.

In der vorliegenden Arbeit wird ein Protokoll zur authentifizierten Selbstbeschreibung von Netzen entwickelt und ein Prototyp implementiert. Hierbei liegt der Schwerpunkt auf der Einfachheit in der Anwendung und dem Betrieb. Das auf LLDP basierende Protokoll bietet eine Möglichkeit zur Verifikation von empfangenen Daten, inklusive Abwehrmaßnahmen gegen einen Angriff durch Wiedereinspielung.

Als Anwendungsbeispiel wird ein in diesem Rahmen zusammengestellter Kleinstrechner mit Bildschirm beschrieben, der als mobile Lösung zur Auswertung der über das Protokoll empfangenen Daten dient. Die entwickelte Lösung kann als Basis für andere Anwendungen der Selbstbeschreibung von Netzen verwendet werden, da sie leicht erweiterbar ist.

Abstract

Local physical networks can be split up into virtual networks, also called VLANs. This allows the separation of traffic on the Data Link Layer, even if devices are connected to the same switch. On each port, one non-tagged and several tagged VLANs can be transported.

Therefore, examination of the physical network topology is not sufficient if one wants to find out which network is transported by one port. Several approaches for identification exist, but these are neither easy to use nor unreliable. Additionally, none of the described approaches offer authentication.

In this thesis, a protocol for the authenticated self-description of local networks is developed and a prototype is implemented. The focus is easy usability and operation. The LLDP-based protocol offers verification of received information and includes counter-measures against replay-attacks.

To demonstrate this, a single-board computer with a display is described, which has been set up with the prototype implementation. It can be used as a compact and mobile solution for the evaluation of data received through the protocol. The implementation can serve as a basis for further applications of the self-description of networks, since it is easily extendable.

Inhaltsverzeichnis

1 Einleitung

Moderne Switches bieten die Trennung von physisch über das gleiche Netz verbundenen Endgeräten in virtuelle Subnetze auf der Sicherungsschicht (VLANs) an. Dies bietet Vorteile bezüglich einfacherer Administration, Sicherheit oder dem Netzverkehrsmanagement, zum Beispiel um Broadcasts von IP-Telefonen vom restlichen Netzverkehr zu trennen.

Beim Anschluss eines neuen Gerätes an ein Rechnernetz ergibt sich allerdings ein Problem: da die Zuordnung von VLANs zu Netzanschlüssen auf Basis der Konfiguration der Switches geschieht, kann man anhand der Verkabelung nicht mehr sicher darauf schließen, welche Netze über eine Ethernet-Buchse transportiert werden. An einem Anschluss können ein portbasiertes VLAN und an sogenannten Trunk-Ports theoretisch bis zu 4094 tagged VLANs nach dem Standard IEEE 802.1Q [IEE14a] transportiert werden. Schließt man in dem Beispiel in Abbildung 1.1 einen Rechner an die Buchse ganz rechts an, kann man ohne Vorwissen über die Konfiguration nicht die transportierten VLANs vorhersagen.

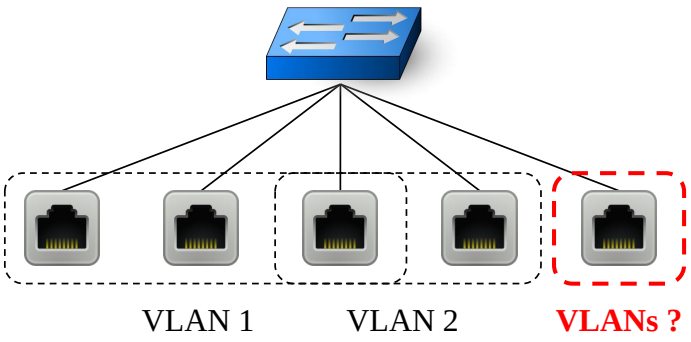

Abbildung 1.1: Trennung von Rechnern in VLANs über die Konfiguration des Switches

Bei der Einbindung von neuen Geräten in ein Netz ist die Information über transportierte Netze aber wichtig, da ein Arbeitsrechner zum Beispiel über das VLAN der IP-Telefone nicht auf sämtliche benötigten Dienste zugreifen kann. Der Anschluss eines Rechners oder eines anderen Gerätes an ein Netz stellt zudem einen sicherheitskritischen Vorgang dar, da sich bei Fehlern hierbei Angriffsmöglichkeiten ergeben. So kann zum Beispiel gewünscht sein, dass die Kommunikation über vertrauliche Informationen nur über ein vertrauenswürdiges Netz erfolgt oder für besonders sicherheitskritische Geräte ein abgetrenntes VLAN verwendet werden soll.

Lösungen wie die Beschriftung der Buchsen sind in großen Netzen aufwändig, schlecht skalierbar und bedeuten bei Änderungen der Netzkonfiguration großen manuellen Aufwand. Gerade beim Anschluss vieler Geräte wäre eine schnelle Identifikation aber hilfreich, da der Zugriff auf die Konfiguration der Switches einerseits umständlich, andererseits aber möglicherweise auch gar nicht erwünscht ist. Wird zum Beispiel ein externer Dienstleister mit der Installation neuer Geräte beauftragt, soll dennoch lediglich der Netzadministrator Zugriff

auf die Konfiguration des restlichen Netzes haben, da diese Informationen intern und zum Beispiel möglicherweise sicherheitsrelevant sind.

Die Auswertung von bereits anfallendem Netzverkehr der Sicherungsschicht und der darin enthaltenen VLAN-Tags setzen voraus, dass in dem betrachteten VLAN zu diesem Zeitpunkt Geräte kommunizieren. Die Verwendung von Informationen aus höheren Schichten des OSI-Modells, wie zum Beispiel DHCP, setzt die Verfügbarkeit solcher Dienste voraus und stellt darüber hinaus einen womöglich unerwünschten Verwaltungsaufwand dar.

Es kommt hinzu, dass unter Umständen die Identifikation mit Einschränkungen möglich, aber keine Möglichkeit zur Authentifizierung vorgesehen ist. Generische Sicherheitslösungen höherer Schichten wie IPsec können für Protokolle der Sicherungsschicht wie dem Link Layer Discovery Protocol nicht als Lösung angewendet werden.

1.1 Zielsetzung und Ergebnisse der Bachelorarbeit

Um eine solche Identifikation von VLANs auf der Sicherungsschicht zu ermöglichen, werden in der vorliegenden Arbeit systematisch ein Protokoll mit grundlegenden Sicherheitsmaßnahmen entwickelt und dieses in einem Prototyp implementiert. Dieser stellt die für Anschluss und Konfiguration von Endgeräten notwendigen Informationen bereit und ermöglicht so die Selbstbeschreibung von lokalen Netzen. Die Authentifizierung ermöglicht die Verifizierung dieser Informationen und beinhaltet Maßnahmen gegen einen Angriff durch Wiedereinspielung. Für das Protokoll wurde der Name **"LAN-Beacon"** gewählt.

Das *konzeptuelle Ergebnis* der Arbeit ist die vorliegende Dokumentation des Vorgehens mit einer Anforderungsanalyse, einer Auswertung des Standes der Technik, dem Protokolldesign, der Implementierungsdokumentation, einer Evaluierung und eines Ausblicks auf weitere mögliche Forschung zu dem Thema.

Die *praktischen Ergebnisse* sind das Protokoll selbst und der Prototyp. Dieses Programm kann im Sender- oder Empfängermodus betrieben werden und ermöglicht die Kommunikation untereinander in einem Rechnernetz.

Um die Ergebnisse anhand eines realistischen Anwendungsfalles zu präsentieren, wird der Prototyp auf einem Einplatinencomputer installiert und bietet die Ausgabe der Informationen auf einem integrierten Bildschirm an. Diese Lösung kann zum Beispiel in Verbindung mit einer mobilen Stromversorgung dazu verwendet werden, um bei Anschluss an einem Netz die Rahmen des entwickelten Protokolls zu empfangen und anzuzeigen. Anhand der Ausgabe sollen die an einem Anschluss verfügbaren Netze identifiziert werden können.

1.2 Aufbau und Herangehensweise der Arbeit

Um eine erfolgreiche Implementierung zu ermöglichen, die sowohl den Ansprüchen aus der Praxis gerecht wird als auch einen wissenschaftlichen Mehrwert bietet, wird für die Abschlussarbeit folgende Methodik angewendet:

1. Zur Definition des Funktionsumfangs und der Gestaltung des Protokolls werden die gegebenen Anforderungen durch Expertenbefragungen ergänzt und anschließend in Kapitel 2 systematisch eingeordnet und dargestellt.

   Das Ergebnis ist eine Liste sowohl funktionaler als auch nicht-funktionaler Kriterien, die sich für die verschiedenen Interessengruppen wie Nutzer, Systemadministratoren

und Entwickler ergeben. Zudem werden die verschiedenen Anforderungen priorisiert und in notwendige und optionale Ziele eingeordnet. Die Anforderungen dienen sowohl der Evaluierung bestehender Lösungen in Kapitel 3 als auch der späteren Entwicklung des Protokolls in Kapitel 4.

2. Um redundante Entwicklungen zu vermeiden und für die Forschung neue Erkenntnisse zu gewinnen, werden anhand der zuvor in Kapitel 2 identifizierten Anforderungen in Kapitel 3 der Stand der Technik und bestehende Lösungsansätze betrachtet. Diese Lösungen werden aufgeführt, zusammengefasst und auf Basis des zuvor entstandenen Kriterienkataloges evaluiert.

   Die Ergebnisse werden verglichen, und es wird auf dieser Basis ein Protokoll als Ausgangspunkt für das Protokolldesign in Kapitel 4 ausgewählt. Als das Protokoll, das den größten Anteil der Anforderungen erfüllt, wird das Link Layer Discovery Protocol (LLDP) identifiziert. Hiermit besteht bereits ein etablierter und leicht erweiterbarer Standard.

3. Mit den gewonnenen Erkenntnissen wird anschließend in Kapitel 4 der Entwurf des Protokolls durchgeführt. Dieses basiert auf der Syntax von LLDP-Rahmen und wird erweitert, um zusätzliche Informationen transportieren zu können. Zudem ist eine Möglichkeit zur Authentifizierung mit Signaturen, Timestamps und ein Challenge-Response Verfahren vorgesehen.

4. Die Implementierung des in Kapitel 4 beschriebenen LAN-Beacon-Protokolls als Prototyp wird in C durchgeführt und in Kapitel 5 dokumentiert. Es werden die bei der Entwicklung getroffenen Implementierungsentscheidungen erklärt und für zukünftige Entwicklungen nützliche Erkenntnisse dargestellt. Die Anwendung des Protokolls und der Implementierung wird anhand der integrierten Lösung auf einem Einplatinencomputer in dem realistischen Anwendungsfalls des Anschlusses neuer Geräte in einem Hochschulnetz dargestellt. Mit Hilfe der in Kapitel 2 gewonnenen Anforderungen wird diese Lösung dann evaluiert.

5. Auf Basis des in Kapitel 5 implementierten und evaluierten Funktionsumfangs werden in Kapitel 6 mögliche Anwendungsfälle und zukünftige Erweiterungen skizziert und Fragestellungen für die weitere Forschung aufgeworfen, wie zum Beispiel bezüglich der Sicherheit der entwickelten Lösung.

2 Anforderungsanalyse

Die in Kapitel 1 genannte Problemstellung beschreibt abstrakt ein Protokoll und ein dazu gehöriges Programm für die authentifizierte Selbstbeschreibung lokaler Netze. Nun werden der Funktionsumfang und die Eigenschaften des Protokolls genauer beschrieben, um eine erfolgreiche Entwicklung zu gewährleisten. Dieser Schritt wird im Kontext der Softwareentwicklung Anforderungsmanagement genannt. Hierbei werden systematisch die gewünschten Eigenschaften von zu entwickelnden Artefakten definiert.

Für die vorliegende Arbeit ergibt sich durch die Problemstellung in Kapitel 1 bereits ein Großteil der Anforderungen. Diese werden in Abschnitt 2.1 durch Anwendungsszenarien genauer spezifiziert und durch Experteninterviews ergänzt.

Hieraus wird in Abschnitt 2.2 ein strukturierter Kriterienkatalog mit den genannten Dimensionen "funktional/nicht-funktional" und "notwendig/optional" entwickelt. Die gewonnenen Anforderungen dienen sowohl der Evaluierung bestehender Lösungen in Kapitel 3 als auch der späteren Ausarbeitung des erweiterten Protokolls in Kapitel 4.

2.1 Anwendungsszenarien

Angelehnt an die Form der User-Story, wie sie zum Beispiel aus der agilen Softwareentwicklung bekannt ist, können verschiedene Anwendergruppen und ihre Erwartungen an das skizzierte Protokoll und die entwickelte Software beschrieben werden. Diese informellen Anwendungsfälle werden anschließend verwendet, um in Abschnitt 2.2 Anforderungen in Form eines formellen Kriterienkataloges für das Protokoll auszuarbeiten.

Um die Anforderungen der Anwendergruppen zu identifizieren, kann man sich in deren Rollen hineinversetzen und aus dem jeweiligen Blickwinkel durchgeführte Tätigkeiten überlegen. Zusätzlich wurden in der Form von Experteninterviews [HD04] zusätzliche Anwendungsfälle und Anforderungen von zwei potentiellen Anwendern identifiziert, die in der Netz- und Systemadministration an einer großen Universität tätig sind.

2.1.1 Anschluss neuer Geräte an ein VLAN (Nutzer):

- Ein Nutzer will bei Anschluss eines Endgerätes sicherstellen, dass es an das gewünschte VLAN angeschlossen wurde. Dies soll möglich sein, obwohl die Zuordnung von VLANs zu Netzanschlüssen anhand der Konfiguration und nicht der Verkabelung geschieht.
- Der Nutzer will sämtliche an einem Anschluss transportierten VLANs zuverlässig herausfinden.
- Der Nutzer will hierfür keine speziellen Programme benötigen, sondern Standardprogramme wie TCPdump oder Wireshark verwenden.
- Der Nutzer will die gewünschten Informationen durch bloßes Betrachten der Ausgaben von TCPdump oder Wireshark finden, erkennen und lesen können. Hierfür sollen keine

spezielle Software oder komplizierte Filterregeln notwendig sein.

- Es sollen weitere Informationen, wie zum Beispiel die E-Mail-Adresse der Systemadministratorin aus dem Rahmen erkenntlich sein. Eine vollständige Liste der gewünschten Informationen befindet sich im Abschnitt 2.2.2.
- Für die Verwendung erweiterter Funktionalitäten soll ein Client-Programm einsetzbar sein, das LAN-Beacon Rahmen empfangen und auswerten kann.
- Da der Anschluss eines Rechners in ein Netz einen sicherheitskritischen Vorgang darstellt, möchte der Nutzer die über das entwickelte Protokoll empfangenen Daten optional verifizieren und somit die Authentizität des Senders und der Informationen sicherstellen.

2.1.2 Änderung der Netzkonfiguration (Systemadministration):

- Eine Systemadministratorin möchte bei Änderungen an der Netzkonfiguration (z.B. Hinzufügen neuer VLANs) die aktualisierten Informationen leicht einpflegen können, um mit dem Protokoll den aktuellen Zustand des Netzes korrekt widerzuspiegeln.
- Die Systemadministratorin will den Dienst auf Server-Seite dauerhaft und stabil betreiben und bei Bedarf nicht erst aktivieren müssen.
- Die Systemadministratorin will das Programm mit möglichst geringem manuellen Aufwand auf neuen Servern installieren und konfigurieren können.

2.1.3 Erweiterung des Protokolls und der Implementierung (Entwickler):

- Ein Entwickler will das Protokoll mit geringem Aufwand um neue Funktionen erweitern und dabei auf möglichst generische und weit verbreitete Ansätze zurückgreifen.
- Der Entwickler will die Änderungen vornehmen, ohne dass dabei die Kompatibilität zwischen verschiedenen Versionen und Implementierungen verloren geht.
- Um die Entwicklung und Einarbeitung zu erleichtern, sollte sich das Protokoll nah an etablierte Verfahren und verbreitete Industriestandards anlehnen.
- Es sollen möglichst geringe Abhängigkeiten bestehen, um die Portabilität zu gewährleisten.

2.2 Kriterienkatalog

Aus diesen Anwendungsszenarien wird nun ein Kriterienkatalog von Anforderungen gewonnen. Um eine Strukturierung zu ermöglichen, werden die Kriterien hierbei in die folgenden unterschiedlichen Dimensionen eingeordnet:

Nutzer-, betreiber- oder entwicklerseitig teilt nach den verschiedenen Personengruppen ein, von denen die Anforderungen gestellt werden. Die Beteiligung dieser Gruppen ist wichtig, da neben den Endanwendern auch Systemadministratoren im Betrieb und Entwickler im Lebenszyklus von Software eine zentrale Rolle spielen.

Funktional oder nicht-funktional teilen in Funktionsumfang, wie zum Beispiel eine Möglichkeit zur Authentifizierung, und gewünschte Rahmenbedingungen, wie zum Beispiel den für den Betrieb möglichst niedrigen Wartungsaufwand oder die für Nutzer möglicherweise wichtigen Reaktionszeiten, ein.

Notwendig oder optional beschreiben die Wichtigkeit einer Anforderung für die erfolgreiche Entwicklung. Hieraus kann eine Priorisierung abgeleitet werden, welche aufgrund der Steuerung des Entwicklungsaufwandes - oder falls sich Anforderungen widersprechen - notwendig sein kann. Ein Beispiel hierfür ist, wenn einerseits optional möglichst kleine Rahmen gewünscht sind, gleichzeitig aber Menschenlesbarkeit unbedingt notwendig ist.

2.2.1 Authentifizierung

In Abschnitt 2.1.1 und 1 wurde die Notwendigkeit einer Möglichkeit zur Verifizierung der empfangenen Informationen dargestellt. Um dies auf der Empfängerseite zu ermöglichen, muss das Protokoll also eine Form von Authentifizierung anbieten.

Dieser Nachweis darf nicht wiederverwendbar sein, also abgetrennt und mit anderslautenden Informationen neu zusammengefügt werden können. Diese Möglichkeit besteht zum Beispiel, wenn lediglich ein Passwort oder andersartiger Identitätsnachweis mitgesendet wird. Es muss ein Verfahren gewählt werden, das lediglich für einen einzelnen Rahmen gültig ist.

Schließlich muss der Missbrauch von Rahmen verhindert werden, welche einmal durch eine legitime Quelle gesendeten wurden. Bei einem solchen Angriff durch Wiedereinspielung könnte der Angreifer einen Rahmen mitschneiden und auf einem anderen Netzsegment zu einem späteren Zeitpunkt wieder senden. Jeder Rahmen darf daher nur für einen bestimmten Zeitraum gültig sein.

Neben Sicherheitsrisiken durch falsch eingestellte Systemzeiten bleibt für den Angreifer eine kurze Zeitspanne, innerhalb derer er einen Rahmen in einem Netz abfangen und an ein anderes Netzsegment weiterleiten kann. Hierbei würde der Angreifer in dem Ursprungsnetz sogar nur einen passiven Angriff durchführen und könnte für den Sender unerkannt bleiben. Daher muss eine Möglichkeit zur aktiven Authentifizierung der Senderseite vorgesehen sein.

2.2.2 Übertragene Informationstypen

In dem entwickelten Protokoll sollen abgesehen von der VLAN-ID und dem VLAN-Namen weitere Informationen übertragen werden, die für eine möglichst vollständige Selbstbeschreibung des Netzes und seiner Konfiguration notwendig sind. Hierbei wird ein Teil der Informationen in maschinenlesbarer Form übertragen, wie zum Beispiel die IP-Netze. Andere Informationen werden direkt als Text in menschenlesbarer Form übertragen. Zusätzlich sollen noch einmal sämtliche Inhalte vollständig in einem Feld in menschenlesbarer Form übertragen werden, damit diese durch bloßes Betrachten des Netzverkehres mit Standardprogrammen herausgefunden werden können.

Im Rahmen der Anforderungsanalyse wurden die folgenden Informationen als nützlich identifiziert:

- Eine Textdarstellung sämtlicher versandter Informationen (Mischung aus Validiertem Text und Freitext)
- VLAN-ID (Validierte, maschinenlesbare Repräsentation)
- VLAN-Name (Menschenlesbarer Freitext)
- Benutzerdefinierter Text (Menschenlesbarer Freitext)
- IPv4-Netze (Validierte, maschinenlesbare Repräsentation)
- IPv6-Netze (Validierte, maschinenlesbare Repräsentation)
- Email-Adresse der Kontaktperson (Menschenlesbarer, validierter Text)
- Informationen über den DHCP-Server (Menschenlesbarer Freitext)
- Informationen über den Router (Menschenlesbarer Freitext)
- Authentifizierungsinformationen, wie in Abschnitt 4.3 beschrieben: Zeitstempel, Challenge und Signatur über die verifizierten Informationen (Maschinenlesbare Repräsentation)

2.2.3 Weitere Anforderungen

Aus den in Abschnitt 2.1 beschrieben Anwendungsszenarien ergeben sich noch einige weitere Anforderungen der jeweiligen Gruppen:

- Nutzer:
  - Die gewünschten Informationen sollen mit möglichst geringem Aufwand und durch bloßes Betrachten der Ausgaben von TCPdump oder Wireshark zu finden, erkennen und lesen sein. Daher müssen die Informationen lesbar formatiert in Klartext übertragen werden.
  - Es sollen Informationen über sämtliche an dem Anschluss transportierten VLANs - und nicht nur über einen Teil davon - gewonnen werden.
- Systemadministration:
  - Informationen über Netze sollen leicht hinzufügbar, änderbar und entfernbar sein.
  - Der Dienst soll sich durch hohe Verfügbarkeit und geringen Wartungsaufwand auszeichnen.
  - Neue Instanzen des LAN-Beacon Servers sollen leicht aufzusetzen sein.
- Entwickler:
  - Das Protokoll soll einfach erweiterbar sein, ohne dabei Inkompatibilität zu älteren Versionen zu verursachen.

- Um die Entwicklung und Einarbeitung zu erleichtern, sollte sich das Protokoll - sofern möglich - nah an etablierten Verfahren und verbreiteten Industriestandards anlehnen.
- Es sollen möglichst geringe Abhängigkeiten bestehen, um die Portabilität zu gewährleisten.

2.3 Zusammenfassung und Einordnung der Anforderungen

Die identifizierten Anforderungen werden in der Tabelle 2.1 zusammengefasst und in die in Abschnitt 2.2 definierten Kategorien eingeordnet. Hierbei werden eindeutige Kürzel vergeben, die bei der Evaluierung des Standes der Technik in Kapitel 3, dem Protokolldesign in Kapitel 4 und bei der Implementierung und Evaluierung in Kapitel 5 referenziert werden.

		Kürzel	Anforderung
Nutzerseitig	Funktional	*Notwendig*	
		N-1)	Für die Identifikation der an einem Netzanschluss transportierten VLANs soll ein möglichst geringer Aufwand notwendig sein.
		N-2)	Die empfangenen Informationen sollen auf Basis von Standardprogrammen wie TCPdump oder Wireshark ausgelesen werden können.
		N-3)	Die Informationen sollen von den in Anforderung N-2) genannten Standardprogrammen in menschenlesbarer Form ausgegeben werden.
		N-4)	Die Authentizität des Servers und der empfangenen Informationen sollen verifiziert werden können.
		Optional	
		N-5)	Die in Abschnitt 2.2.2 aufgezählten weiteren Informationen sollen empfangen werden können.
	Nicht-funktional	*Notwendig*	
		N-6)	Die Identifikation sämtlicher an einem Netzanschluss transportierten virtuellen LANs soll zuverlässig möglich sein.
		N-7)	Es sollen Schutzmaßnahmen gegen Angriff durch Wiedereinspielung vorgesehen sein.
		Optional	
		N-8)	Die Ergebnisse der Identifikation der virtuellen Netze sollen nach möglichst kurzer Zeit zur Verfügung stehen.
Betreiberseitig	Funktional	*Notwendig*	
		B-1)	Bei der Änderung der Netztopologie soll für die Einpflegung der Informationen ein möglichst geringer Aufwand notwendig sein.
	Nicht-funktional	*Notwendig*	
		B-2)	Der dauerhafte Betrieb des Protokolls soll mit einer geringen Auslastung des Netzes und der Ressourcen von Client und Server möglich sein.
		Optional	
		B-3)	Für die Integration des Protokolls in ein virtuelles LAN soll ein möglichst geringer Aufwand notwendig sein.
Entwicklerseitig	Nicht-funktional	*Notwendig*	
		E-1)	Die Integration neuer Funktionen in das Protokoll soll bei gleichzeitigem Erhalt der Kompatibilität möglich sein.
		Optional	
		E-2)	Für die Integration von neuen Funktionen in das Protokoll soll möglichst wenig Aufwand notwendig sein.
		E-3)	Das Protokoll soll auf möglichst weit verbreiteten Standards aufbauen.
		E-4)	Es sollen möglichst wenige Abhängigkeiten von Drittprogrammen und -bibliotheken bestehen.

Tabelle 2.1: Gesammelte Anforderungen zusammengefasst und eingeordnet

3 Stand der Technik

Da die in Kapitel 2 ausgearbeiteten Anforderungen Problemen aus dem Alltag entspringen, bestehen hierzu oder zu ähnlichen Problemen bereits Vorarbeiten oder Lösungen. Nach einer kurzen Beschreibung der jeweiligen Funktionsweise werden diese anhand des zuvor entwickelten Kriterienkatalogs auf Anwendbarkeit in dem skizzierten Fall überprüft. Damit sollen redundante Arbeit verhindert und neue Erkenntnisse für die Wissenschaft sichergestellt werden.

Selbst wenn die genannten Anforderungen noch von keinem Protokoll vollständig erfüllt werden, sind Teilaspekte bereits gelöst. Die Identifikation dieser Lösungen bietet den Vorteil, die Arbeit an etablierten Standards und Vorgehensweisen anlehnen zu können. Dadurch werden bereits bekannte Fehlerquellen vermieden und eine für Nutzer, Betreiber und Entwickler des Protokolls intuitive Funktionsweise erreicht. Diese Analyse dient dann als Ausgangspunkt für das in Kapitel 4 folgende Protokolldesign.

3.1 Direktes Auslesen von VLAN-Tags

In tagged VLANs nach dem Standard IEEE 802.1Q [IEE14a] wird die VLAN-ID direkt in den Ethernet-Rahmen mittransportiert. Wie in Abbildung 3.1 dargestellt wird hierfür ein 32 Bit Feld zwischen die Quelladresse und Typ-Feld eingefügt. Switches entfernen diesen Tag im Fall von sogenannten Access Ports. Diese sind für den Anschluss von Endgeräten vorgesehen, welche daher Rahmen ohne Tags erhalten.

Der Tag selbst beinhaltet den 16 Bit langen Tag Protocol Identifier (TPID), welcher durch den EtherType das Vorhandensein des VLAN-Tags kennzeichnet. Die verbleibenden 16 Bit werden für die Tag Control Information (TCI) verwendet, welche sich folgendermaßen aufteilt: der 3 Bit lange Priority Code Point (PCP) definiert die Priorität des Rahmens, der 1 Bit lange Drop Eligible Indicator (DEI) dient der Stausteuerung und der 12 Bit lange VLAN Identifier (VID) enthält die ID an sich.

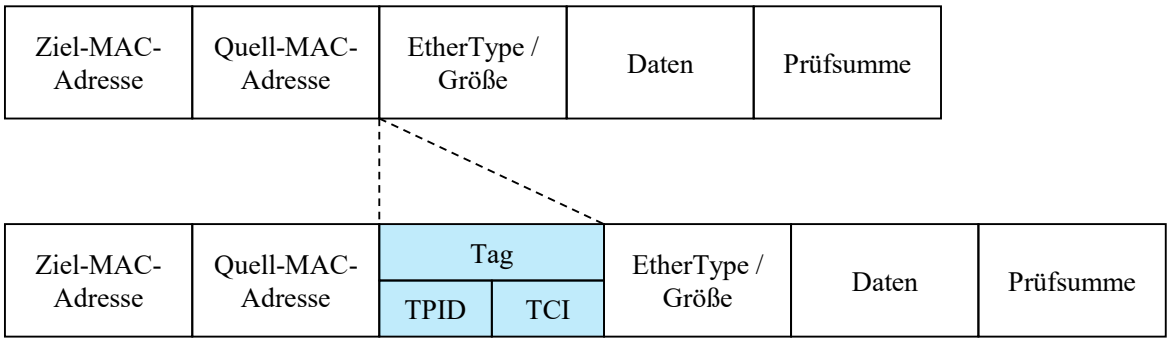

Abbildung 3.1: Aufbau eines 802.1Q tagged VLAN Rahmens [CS06]

Sobald man an einem Netzanschluss nach 802.1Q getaggten Verkehr empfängt, kann man die TCI auslesen. Anhand des hierin enthaltenen VLAN Identifier (VID) kann man anschließend die VLAN-ID herausfinden, an die der Rahmen gesendet wurde. Sofern man aus sämtlichen verfügbaren VLANs Rahmen empfangen hat, kann man auf diese Weise die auf einem Anschluss transportierten VLANs herausfinden.

Tagged VLANs stehen im Gegensatz zu portbasierten VLANs, bei welchen die Weiterleitung von Verkehr lediglich auf Basis der Konfiguration der Anschlüsse durchgeführt wird. Daher sind in den Ethernet-Rahmen keine Tags enthalten.

Bewertung der Anwendbarkeit

Von den zuvor herausgearbeiteten Anforderungen werden die folgenden erfüllt:

N-2): Programme wie TCPdump können die VLAN-IDs anzeigen, zum Beispiel mit dem folgendem Befehl: [1]

```
tcpdump -n -i eth4 host 10.10.10.10 -e
```

B-1): Sobald ein neues VLAN konfiguriert wurde, wird der Tag automatisch in die Rahmen eingefügt. Sobald aus diesem neuen VLAN Rahmen empfangen werden, kann die VLAN-ID aus den Tags ausgelesen werden. Weitere Konfigurationen sind daher nicht notwendig.

B-2): Da die Informationen in den Tags bereits mitgesendet werden, ergibt sich keine zusätzliche Belastung der Ressourcen durch das Auslesen.

B-3): Abgesehen von der Konfiguration der VLANs ist keine weitere Konfiguration notwendig.

E-3): VLAN-Tags nach IEEE 802.1Q sind standardisiert und werden von vielen Produkten unterstützt.

E-4): Das beschriebene Verfahren kann ohne Abhängigkeiten von spezifischer Drittsoftware angewandt werden.

Folgende Anforderungen werden nicht erfüllt oder sind nicht zutreffend:

N-1): Die Methode funktioniert grundsätzlich nur bei tagged VLANs, bei den ebenfalls noch immer verbreiteten portbasierten VLANs jedoch nicht. Falls Tags den Rechner erreichen, können sie aber ebenfalls nicht zuverlässig ausgelesen werden, da sie teilweise vom Treiber des Netzkarte entfernt werden. [Wir]

N-3): Die Information sind in dem Netzverkehr enthalten, allerdings als Zahl in Binärformat und nicht als Text. Da lediglich die ID des VLANs in dem Tag gesendet wird, kann der Name nicht direkt ausgelesen werden.

N-4) und N-7): Nicht erfüllt, da keine Sicherheitsmaßnahmen oder Authentifizierungsmöglichkeiten vorgesehen sind. Durch verschiedene VLAN-Hopping Angriffe (zum Beispiel Double 802.1q Encapsulation VLAN Hopping Angriff) [YB05] können an einen Rechner

[1] https://nicolas.vion.science/network-analysis-tcpdump-vlan/show-vlan-header-with-tcpdump.html

sogar von außerhalb eines VLANs manipulierte Rahmen mit VLAN-Tag mit falschen Werten geschickt werden.

N-5): Es können neben der VLAN-ID keine weiteren Informationen empfangen werden.

N-6): Eine zuverlässige Identifikation sämtlicher VLANs ist nicht möglich. Selbst wenn sämtlichen VLANs tagged sind, können sie nur dann identifiziert werden, wenn auf sämtlichen zu identifizierenden VLANs zum Zeitpunkt der Messung Verkehr empfangen wird.

N-8): Die Informationen sind sofort verfügbar, sobald aus einem VLAN Verkehr empfangen wird. Es ist allerdings keine Aussage darüber möglich, wie schnell Rahmen aus einem Netz empfangen werden.

E-1) und E-2): Eine Erweiterung des Protokolls ist nicht vorgesehen und führt zu Inkompatibilität, die 2 Felder mit der Gesamtlänge von 16 Bit für die Nutzlast bereits vollkommen für die TCI verwendet werden.

3.2 Patent „Automatic VLAN ID discovery for ethernet ports“

Dieses Softwarepatent aus den USA [JN09] beschreibt eine Methode zur Erkennung der an einem Anschluss verfügbaren VLANs, ohne zusätzlich zum Standard 802.1Q weitere Protokolle in einem Netz installieren zu müssen. Es basiert auf dem zuvor beschriebenen Verfahren des direkten Auslesens der Tags und soll eine teilweise Lösung von Anforderung N-6) darstellen. VLANs sollen also auch dann erkannt werden, wenn passiv kein Netzverkehr auf einem Anschluss empfangen wird.

Durch die auf 12 Bit limitierte Länge der VLAN-ID sind maximal 4094 verschiedene eindeutige VLAN-IDs möglich, wobei 2 weitere IDs reserviert und daher nicht verfügbar sind. Dadurch ist eine Aufzählung aller möglichen Kombinationen und der jeweilige Versand von Rahmen mit einer Anfrage (z.B. ein Ping-Paket) an diese VLANs in vertretbarer Zeit möglich. [IEE14a]

Wenn auf die Anfrage keine Antwort erfolgt, wird davon ausgegangen, dass der Switch den Rahmen aufgrund der ungültigen VLAN-ID verworfen hat. Wird aus dem adressierten Netz eine Antwort empfangen, so lässt sich daraus schließen, dass eine gültige VLAN-ID angesprochen wurde.

Bewertung der Anwendbarkeit

Von den zuvor herausgearbeiteten Anforderungen werden die folgenden erfüllt:

B-1): Sobald aus einem neuen VLAN ein Gerät auf Anfragen antwortet, ist die VLAN-ID unter den zuvor genannten Bedingungen sofort aus den Tags auslesbar. Weitere Konfigurationen von Seite des Netzes sind nicht notwendig.

B-3): Abgesehen von der Konfiguration der tagged VLANs ist keine weitere Konfiguration notwendig.

E-4): Das beschriebene Verfahren kann ohne Abhängigkeiten von Drittsoftware implementiert werden.

Folgende Anforderungen werden nicht erfüllt oder sind nicht zutreffend:

N-1): Um die verfügbaren VLANs herauszufinden, muss zuerst ein aufwändiger Suchlauf durchgeführt und auf sämtliche Antworten gewartet werden.

N-2): Das Verfahren ist in Standardprogrammen nicht implementiert und benötigt daher ein gesondertes Programm.

N-3): Nicht zutreffend, da Anforderung N-2) nicht erfüllt ist.

N-4) und N-7): Nicht erfüllt, da keine Sicherheitsmaßnahmen oder Authentifizierungsmöglichkeiten vorgesehen sind.

N-5): Es können neben der VLAN-ID keine weiteren Informationen empfangen werden.

N-6): Eine zuverlässige Identifikation sämtlicher VLANs ist nicht möglich. Aus einer ausbleibenden Antwort wird geschlossen, dass die entsprechende VLAN-ID nicht existiert. Ein anderer möglicher Grund hierfür könnte aber sein, dass in dem entsprechenden virtuellen Netz zu dem Zeitpunkt des Tests keine Geräte eingeschaltet waren oder der Rahmen aufgrund von Überlastung verworfen wurde.

N-8): Die Geschwindigkeit des Suchlaufs hängt von der Kapazität des Netzes ab. Es ist daher keine Aussage darüber möglich, wie schnell Antworten aus einem Netz empfangen werden.

B-2): Das Verfahren verursacht durch die aufwändigen und umfassenden Suchläufe bei dauerhaftem Einsatz eine hohe Netzlast und Beanspruchung der Ressourcen von dem sendenden Gerät und den Empfängern.

E-1) und E-2): Eine Erweiterung des Protokolls ist nicht vorgesehen und führt zu Inkompatibilität, die 2 Felder mit der Gesamtlänge von 16 Bit für die Nutzlast bereits vollkommen für die TCI verwendet werden.

E-3): Das beschriebene Verfahren beruht auf dem Standard IEEE 802.1Q, Implementierungen des Verfahrens selbst konnten aber nicht gefunden werden.

3.3 IEEE 802.1AB: Link Layer Discovery Protocol (LLDP)

Der IEEE-Standard „802.1AB: Station and Media Access Control Connectivity Discovery“ beschreibt das Protokoll Link Layer Discovery Protocol (LLDP) [IEE09]. Hierbei versenden sogenannte LLDP-Agents regelmäßig Ethernet-Rahmen an eine Multicast-Adresse, welche von den LLDP-Agents benachbarter Geräte empfangen werden. Die Rahmen enthalten z. B. Informationen über den Typ (z.B. IP-Telefon) und Fähigkeiten (Switching / Routing) des sendenden Gerätes. Die Empfänger können die Informationen analysieren und daraus Erkenntnisse über das Netz gewinnen, wodurch eine Selbstbeschreibung des lokalen Netzes ermöglicht wird.

LLDP-Rahmen werden über Ethernet mit dem EtherType 88-CC an eine von 3 möglichen Multicast-Adressen versendet und beinhalten die LLDP Data Unit (LLDPDU), also Protokolldateneinheit (PDU), mit der durch Ethernet gegebenen maximalen Größe von 1500 Byte, wie in Abbildung 3.2 dargestellt.

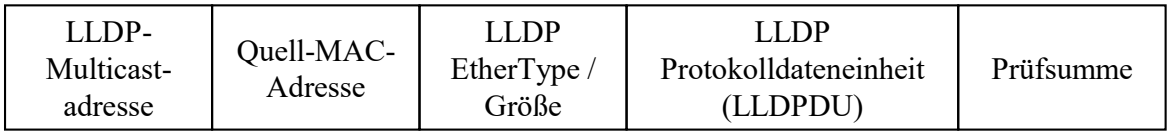

Abbildung 3.2: Aufbau eines LLDP-Rahmens [IEE09]

Die in Abbildung 3.3 dargestellte LLDPDU selbst enthält in der Form von Typ-Länge-Werten (TLVs) Informationen über die Eigenschaften und Konfiguration des Senders und wird mit zwei Null-Bytes abgeschlossen. Unter anderem werden Felder für den Namen und eine Beschreibung des Gerätes, die Wartungsadresse oder auch die VLAN-ID des sendenden Anschlusses eingefügt. Aus der Größe von 9 Bit für das Länge-Feld ergibt sich eine maximale Größe von maximal 513 Byte für jede TLV, wobei 2 Byte für den Header benötigt werden. Für die Nutzlast, den sogenannten "TLV information string", verbleiben somit 511 Byte.

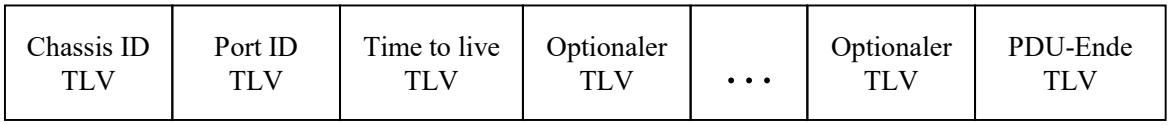

Abbildung 3.3: Inhalte einer LLDPDU [IEE09]

Zusätzlich zu den im Standard definierten TLVs sind frei definierbare, organisationsspezifische TLVs vorgesehen. Diese sind wie in Abbildung 3.4 aufgebaut: das Typ-Feld enthält den Wert 127, außerdem wird eine 4 Byte lange Zeichenfolge zur Identifikation der den Typ-Länge-Wert definierenden Organisation und des Subtyps mitgeschickt, womit 507 Byte für die Nutzlast bleiben. Hiermit könnte zum Beispiel ein Hersteller wie Cisco Informationen in die LLDP-Rahmen integrieren, die den Austausch von herstellerspezifischen Informationen zwischen benachbarten Switches ermöglichen sollen. Eine Weiterleitung dieser organisationsspezifischen TLVs ist im Standard allerdings nicht vorgesehen. Auf dieses Problem wird in Abschnitt 4.5 genauer eingegangen.

Abbildung 3.4: Aufbau und Inhalte einer organisationsspezifischen LLDP TLV[IEE09]

In dem Standard ist bereits eine TLV für die ID und den Namen des VLANs vorgesehen [IEE09, E.4]. Sofern die Switches für den Versand dieser Informationen konfiguriert wurden, können diese aus dem LLDP-Rahmen ausgelesen werden.

Bewertung der Anwendbarkeit

Von den zuvor herausgearbeiteten Anforderungen werden die folgenden erfüllt:

N-1): LLDP-Rahmen werden regelmäßig von LLDP-Agents versendet und enthalten sämtliche konfigurierten TLVs. Hiermit ist eine passive Identifikation der virtuellen Netze ohne großen Aufwand möglich.

N-2): Programme wie TCPdump können die LLDP-Rahmen anzeigen, zum Beispiel mit dem folgendem Befehl: [2]

```
tcpdump -nn -v -i eth0 -s 1500 -c 1 '(ether[12:2]=0x88cc or
    ether[20:2]=0x2000)'
```

N-5): Im LLDP-Standard werden bereits eine Vielzahl von TLVs definiert, zusätzlich sind benutzerdefinierte TLVs vorgesehen. Auf verschiedenen Plattformen verfügbare Implementierungen - wie zum Beispiel die Software LLDPD [Ber] - sehen das Hinzufügen von solchen benutzerdefinierten TLVs vor.

N-8): Der regelmäßige Versand von LLDPDUs ist im Protokoll vorgesehen, daher ist eine Identifikation innerhalb der konfigurierten Sendefrequenz möglich.

B-2): Das Protokoll sieht den regelmäßigen Versand von PDUs vor. Daraus ergibt sich bei dauerhaftem Betrieb eine geringfügige Belastung der Netze und Ressourcen.

E-1): In dem Standard sind zur Erweiterung benutzerdefinierte TLVs vorgesehen. [IEE09, 8.6] Daher ist die Kompatibilität zu alten Versionen bei der Integration neuer Funktionen nicht beeinträchtigt. LLDP-Agents können TLVs mit unbekanntem Typ einfach verwerfen.

E-2): In dem Standard sind zur Erweiterung benutzerdefinierte TLVs vorgesehen. [IEE09, 8.6] Daher ist das Hinzufügen neuer Funktionen sehr einfach und benötigt keine Änderungen an anderen Teilen des Protokolls.

E-3): Das Protokoll stellt einen etablierten Standard für die Selbstbeschreibung von Netzen dar und ist aufgrund der Unterstützung durch Produkte großer Hersteller wie Cisco und Dell weit verbreitet. Im Internet sind umfangreiche Informationen für Entwickler und Nutzer verfügbar.

E-4): Das Protokoll kann ohne Abhängigkeiten von Drittsoftware implementiert werden.

Folgende Anforderungen werden nicht erfüllt oder sind nicht zutreffend:

N-3): Die Informationen sind in den LLDPDUs enthalten, manche - so wie die VLAN-ID - allerdings als Zahl in Binärformat und nicht als Text. Da auch keine menschenlesbaren Bezeichner enthalten sind, ist die Identifikation durch den Subtyp bei bloßer Betrachtung der Ausgabe von Programmen wie TCPdump schwer erkennbar.

[2] http://unix.stackexchange.com/questions/127245/in-which-vlan-am-i-in

N-4) und N-7): Nicht erfüllt, da keine Sicherheitsmaßnahmen oder Authentifizierungsmöglichkeiten vorgesehen sind. Als grundlegende Sicherheitsmaßnahme wurde der Versand von einer Art Sicherheitsschlüssel in LLDP vorgeschlagen [LL08], wobei diese Methode einen Angriff durch Wiedereinspielung nicht verhindern kann.

N-6): Eine zuverlässige Identifikation ist nicht möglich. Sofern LLDP-Rahmen von einem LLDP-Agent zwischen Server und Client empfangen werden, werden diese dem Standard entsprechend nicht weitergeleitet.

B-1): Änderungen der Netztopologie müssen sämtlichen LLDP-Agents in diesem VLAN bekannt gemacht werden, bevor diese die korrekten Informationen senden können. Dieses Problem besteht, da eine Weiterleitung von TLVs nicht vorgesehen ist.

B-3): Die Konfiguration muss auf sämtlichen LLDP-Agents in dem VLAN durchgeführt werden, bevor diese die korrekten Informationen senden. Dieses Problem besteht, da eine Weiterleitung von TLVs nicht vorgesehen ist.

3.4 Cisco Discovery Protocol (CDP) und andere proprietäre „Discovery“ Protokolle der Sicherungsschicht

Vor der Standardisierung von LLDP wurden von Netzausrüstern verschiedene Protokolle für die Selbstbeschreibung von Netzen entwickelt, mit welchen zum Beispiel Switches untereinander kommunizieren und Informationen über ihre Konfiguration austauschen können. Zum Teil konnten darüber auch VLAN-Informationen transportiert werden.

Ein Beispiel hierfür ist das Cisco Discovery Protocol (CDP), das vor der Standardisierung von LLDP zum Teil auch von anderen Herstellern übernommen wurde. Das Protokoll wird zum Teil noch unterstützt, aber nicht mehr aktiv weiterentwickelt und durch LLDP ersetzt. Zum Beispiel können bei manchen Produkten CDP-Rahmen zwar noch empfangen, aber nicht mehr versendet werden [HP06].

Bewertung der Anwendbarkeit

Da LLDP der Nachfolger von CDP und ähnlichen proprietären Protokollen ist, entspricht das Ergebnis der Anforderungsanalyse weitestgehend dem letzten Kapitel. Zudem erfüllen sie nicht die Anforderung E-3), da diese Protokolle nicht mehr aktiv unterstützt werden und daher die Verbreitung in Zukunft noch weiter abnehmen wird.

3.5 Link Layer Topology Discovery (LLTD)

Das Link Layer Topology Discovery (LLTD) Protokoll [Mic10] wurde von Microsoft zur Sammlung von Informationen über die Netztopologie und Dienstgüte entwickelt. Hierbei ist die Verwendung in Schicht-2-Netzen wie Ethernet nach IEEE 802.3 als auch in kabellosen Netzen wie WLAN nach IEEE 802.11 vorgesehen.

Hierbei werden von einem sogenannten „Mapper“-Programm Anfragen zur Identifikation von angeschlossenen Geräten versendet, welche dann zum Beispiel von Druckern oder anderen Rechnern beantwortet werden, in diesem Kontext sogenannte „Responder“. In Windows-Systemen wird das Protokoll verwendet, um eine sogenannte „Network Map“ zu generieren,

welche die gewonnenen Informationen graphisch darstellt. Für Linux-Systeme wird von Microsoft eine lizenzkostenfreie Implementierung angeboten.

Das Protokoll wurde erstmals unter Windows Vista mitgeliefert und in andere Microsoft-Produkte wie unter anderem die X-BOX integriert. [MSd14] Die aktuelle Spezifikation stammt aus dem Jahr 2010. Informationen über aktuelle Entwicklungen und Adoption durch andere Hersteller sind aktuell nicht zu finden.

Informationen über die transportierten VLANs können zum Beispiel in dem Freitext-Feld „Friendly Name“ übertragen werden, während in den Feldern für Gerätenamen und -ID ein Bezeichner für VLANs definiert werden kann. Hiermit wäre bei der Verwendung von Windows ohne Zusatzprogramme und sogar ohne Verwendung der Kommandozeile die Identifikation der transportierten virtuellen Netze möglich. Für Linux-Systeme wäre eine Implementierung eines solchen Empfängers möglich.

Bewertung der Anwendbarkeit

Von den zuvor herausgearbeiteten Anforderungen werden die folgenden erfüllt:

N-8): Die Antwort des Responders erfolgt umgehend auf den Empfang der Anfrage des Mappers. Daher ist eine sehr schnelle Identifikation der VLANs möglich.

B-2): Das Protokoll sieht den Versand von Informationen nur auf Anfrage hin vor. Daraus ergibt sich im dauerhaften Betrieb eine extrem geringe Belastung der Netze und Ressourcen.

Folgende Anforderungen werden teilweise erfüllt:

N-1): Der Client, in diesem Fall Mapper genannt, muss zur Identifikation der Netztopologie eine Anfrage versenden. Hiermit ist die Identifikation der virtuellen Netze mit einem gewissen Aufwand verbunden.

N-5): Für eine Vielzahl der in Abschnitt 2.2.2 definierten Informationen sind bereits Felder vorgesehen, zum Beispiel die „Support Information“ zum Transport einer E-Mail Adresse. Selbst definierte Datentypen sind nicht vorgesehen, allerdings können die meisten Informationen in den Freitextfeldern übertragen werden.

B-1): Änderungen der Netztopologie müssen nur dem für den Versand der VLAN-Informationen konfigurierten Responder bekannt gemacht werden, bevor dieser auf Anfragen mit der richtigen Information antworten kann.

B-3): Die Konfiguration muss nur dem für den Versand der VLAN-Informationen konfigurierten Responder bekannt gemacht werden, bevor dieser auf Anfragen mit der richtigen Information antworten kann.

E-1): Zum Teil müssten zur Übertragung der gewünschten Informationen Freitextfelder verwendet werden, in denen auch andere Informationen übertragen werden. Da sich beim Hinzufügen neuer Informationen der Inhalt dieser Felder ändert und keine standardisierte Syntax hierfür vorgesehen ist, könnte dies zu Inkompatibilität führen.

E-2): Zusätzliche Informationen können in den Freitextfeldern hinzugefügt werden, es können allerdings keine Felder mit beliebigen Inhalten definiert werden.

E-3): Das proprietäre Protokoll ist als Bestandteil von Windows weit verbreitet. Allerdings konnten keine umfassenden Daten über die Implementierung in den Produkten anderer Hersteller gefunden werden.

Folgende Anforderungen werden nicht erfüllt oder sind nicht zutreffend:

N-2): Das Stellen der Anfrage ist in Standardprogrammen nicht implementiert und benötigt daher ein gesondertes Programm.

N-3): Nicht zutreffend, da Anforderung N-2) nicht erfüllt ist.

N-4) und N-7): Nicht erfüllt, da keine Sicherheitsmaßnahmen oder Authentifizierungsmöglichkeiten vorgesehen sind.

N-6): Eine zuverlässige Identifikation sämtlicher VLANs ist mit diesem Protokoll möglich, sofern in allen virtuellen Netzen Responder aktiv sind. Allerdings müssen hierfür zuerst in sämtliche dieser Netze Anfragen versendet werden. Hiermit ergeben sich ähnliche Probleme wie in Abschnitt 3.2 beschrieben.

E-4): Auf Windowssystemen besteht eine starke Abhängigkeit von der Implementierung durch Microsoft. Für Linux-Systeme ist ebenfalls eine Implementierung von Microsoft verfügbar.

Das Protokoll kann aber auch ohne Abhängigkeiten von Drittsoftware implementiert werden, da Microsoft die Spezifikationen des Protokolls veröffentlicht hat.

3.6 Selbstbeschreibung des VLANs durch Stellen einer Anfrage

Das Patent [JPTH08] beschreibt ein Protokoll zwischen zwei Rechnern in einem Netz, womit die VLAN-ID herausgefunden werden kann, ohne die Switches um neue Funktionalitäten erweitern zu müssen. Hierbei sendet ein in einem Netz neu angeschlossener Rechner einen „discovery request“, in dem er nach der ID des VLANs fragt. Ein zweiter Rechner sendet hierauf eine Antwort mit der ID des VLANs an den ersten Rechner.

Bewertung der Anwendbarkeit

Von den zuvor herausgearbeiteten Anforderungen werden die folgenden erfüllt:

N-8): Die Antwort des Responders erfolgt umgehend auf den Empfang der Anfrage des Mappers. Daher ist eine sehr schnelle Identifikation der VLANs möglich.

B-2): Das Protokoll sieht den Versand von Informationen nur auf Anfrage hin vor. Daraus ergibt sich im dauerhaften Betrieb eine extrem geringe Belastung der Netze und Ressourcen.

E-4): Das Protokoll kann ohne Abhängigkeiten von Drittsoftware implementiert werden.

Folgende Anforderungen werden teilweise erfüllt:

N-1): Der Client muss zur Identifikation der Netztopologie eine Anfrage versenden. Hiermit ist die Identifikation der virtuellen Netze mit einem gewissen Aufwand verbunden.

B-1): Änderungen der Netztopologie müssen nur dem für den Versand der VLAN-Informationen konfigurierten Server bekannt gemacht werden, bevor dieser mit der richtigen Information auf Anfragen antworten kann.

B-3): Die Konfiguration muss nur dem für den Versand der VLAN-Informationen konfigurierten Server bekannt gemacht werden, bevor dieser mit der richtigen Information auf Anfragen antworten kann.

E-1 und E-2): Die Erweiterbarkeit hängt von der Ausgestaltung des Protokolls ab, welche in dem Patent nicht näher spezifiziert wird.

Folgende Anforderungen werden nicht erfüllt oder sind nicht zutreffend:

N-2): Das Stellen der Anfrage ist in Standardprogrammen nicht implementiert und benötigt daher ein gesondertes Programm.

N-3): Nicht zutreffend, da Anforderung N-2) nicht erfüllt ist.

N-4) und N-7): Nicht erfüllt, da keine Sicherheitsmaßnahmen oder Authentifizierungsmöglichkeiten vorgesehen sind.

N-5): Das Patent spezifiziert die Inhalte der Anfragen und Antworten nicht genauer, daher kann über die übertragbaren Informationen keine Aussage getroffen werden.

N-6): Eine zuverlässige Identifikation sämtlicher VLANs ist mit diesem Protokoll möglich, sofern in allen virtuellen Netzen ein Server aktiv ist und somit auf die Anfrage antworten kann. Allerdings müssen hierfür zuerst in sämtliche dieser Netze Anfragen versendet werden. Hiermit ergeben sich ähnliche Probleme wie in Abschnitt 3.2 beschrieben.

E-3): Zu dem tatsächlichen Einsatz des beschriebenen Protokolls konnten keine Informationen gefunden werden.

3.7 Herstellerspezifische Erweiterungen für DHCP

DHCP ist ein Protokoll der Anwendungsschicht, das die Zuweisung von Konfigurationen an Zielrechner durch einen Server automatisiert. Der klassische Anwendungsfall ist hierbei die Zuweisung von IP-Adressen, allerdings gibt es auch noch eine Vielzahl von anderen Optionen. Der RFC2132 definiert Option 43 als „Vendor Specific Information“ [DA97, 8.4.]. Diese wird zum Beispiel von Microsoft im VOIP-Server „Lync Server“ verwendet, um VLAN-IDs zu versenden [3].

Die Benutzung eines bereits vorhandenen DHCP-Dienstes stellt einen geringen Aufwand dar. Die Installation des Dienstes in sämtlichen VLANs, wo dieser nicht zur Verfügung steht, bedeutet allerdings einen bedeutenden Zusatzaufwand aufgrund der nicht benötigten DHCP-Funktionen.

[3] https://technet.microsoft.com/de-de/library/gg398088(v=ocs.14).aspx

Bewertung der Anwendbarkeit

Von den zuvor herausgearbeiteten Anforderungen werden die folgenden erfüllt:

N-1): DHCP-Dienste senden regelmäßig Pakete. Hiermit ist eine passive Identifikation der virtuellen Netze ohne großen Aufwand möglich.

N-2): Programme wie TCPdump und Wireshark können den DHCP-Verkehr mitschneiden und z. B. über die Portnummer leicht herausfiltern.

N-6): Eine zuverlässige Identifikation sämtlicher VLANs ist mit diesem Protokoll möglich, sofern in allen virtuellen Netzen ein DHCP-Server aktiv ist.

N-8): Der regelmäßige Versand von DHCP-PDUs ist im Protokoll vorgesehen, daher ist eine Identifikation innerhalb der konfigurierten Sendefrequenz möglich.

B-2): Das Protokoll sieht den regelmäßigen Versand von PDUs vor. Daraus ergibt sich im dauerhaften Betrieb eine geringe Belastung der Netze und Ressourcen.

E-3): Bei DHCP handelt es sich um einen weit verbreiteten und etablierten Standard.

E-4): Sofern die frei konfigurierbaren Optionen für die Beschreibung der VLANs über einen bereits vorhandenen DHCP-Server versendet werden sollen, besteht eine Abhängigkeit von der hierfür verwendeten Software. Das Protokoll kann aber auch ohne Abhängigkeiten von Drittsoftware implementiert werden.

Folgende Anforderungen werden teilweise erfüllt:

N-3): Die Lesbarkeit ist von der Implementierung der Erweiterung abhängig, allerdings konnte keine Implementierung einer menschenlesbaren Variante gefunden werden. Grundsätzlich ist der Transport als menschenlesbarer Text in den pro Option verfügbaren 312 Byte aber möglich. [Ste06]

N-5): DHCP sieht eine Reihe von zusätzlichen Informationen vor, die in dem „options“ Feld transportiert werden. Hierin sind die in Abschnitt 2.2.2 geforderten Informationen zwar nicht enthalten, allerdings können selbst definierte DHCP-Optionen mitgesendet werden.

B-1): Änderungen der Netztopologie müssen nur dem für den Versand der VLAN-Informationen konfigurierten Server bekannt gemacht werden, bevor dieser mit der richtigen Information auf Anfragen antworten kann.

B-3): Die Konfiguration muss nur dem für den Versand der VLAN-Informationen konfigurierten DHCP-Server bekannt gemacht werden.

E-1): Zusätzliche Funktionen können in dem Feld „options“ hinzugefügt werden. Da zur eindeutigen Identifikation des Subtyps nur 1 Byte zur Verfügung steht, sind Überschneidungen mit anderen Protokollen und eine so verursachte Inkompatibilität möglich.

E-2): Zusätzliche Funktionen können in dem Feld „options“ hinzugefügt werden. Hierfür stehen allerdings lediglich 312 Byte zur Verfügung. [Ste06]

Folgende Anforderungen werden nicht erfüllt oder sind nicht zutreffend:

N-4) und N-7): Nicht erfüllt, da keine Sicherheitsmaßnahmen oder Authentifizierungsmöglichkeiten vorgesehen sind.

3.8 Zusammenfassung: Vergleich von Anforderungen und Stand der Technik

Die Ergebnisse der Evaluierung des Standes der Technik sind in Tabelle 3.1 zusammenfassend und in Anlehnung an Tabelle 2.1 dargestellt. Hierbei wird für jedes Verfahren aufgeführt, ob es die jeweiligen Anforderungen erfüllt.

		Kürzel	VLAN-Tags	VLAN-Scanner	LLDP	LLTD	ID anfragen	DHCP
Nutzerseitig	Funktional	Notwendig						
		N-1)	Nein	Nein	Ja	Teilweise	Teilweise	Ja
		N-2)	Ja	Nein	Ja	Nein	Nein	Ja
		N-3)	Nein	Nein	Teilweise	Teilweise	Nein	Teilweise
		N-4)	Nein	Nein	Nein	Nein	Nein	Nein
		Optional						
		N-5)	Nein	Nein	Ja	Teilweise	Nein	Teilweise
	Nicht-funktional	Notwendig						
		N-6)	Nein	Nein	Nein	Nein	Nein	Ja
		N-7)	Nein	Nein	Nein	Nein	Nein	Nein
		Optional						
		N-8)	Nein	Nein	Ja	Ja	Ja	Ja
Betreiberseitig	Funktional	Notwendig						
		B-1)	Ja	Ja	Nein	Teilweise	Teilweise	Teilweise
	Nicht-funktional	Notwendig						
		B-2)	Ja	Nein	Ja	Ja	Ja	Ja
		Optional						
		B-3)	Ja	Ja	Nein	Teilweise	Teilweise	Teilweise
Entwickler-seitig	Nicht-funktional	Notwendig						
		E-1)	Nein	Nein	Ja	Teilweise	Teilweise	Teilweise
		Optional						
		E-2)	Nein	Nein	Ja	Teilweise	Teilweise	Teilweise
		E-3)	Ja	Nein	Ja	Teilweise	Nein	Ja
		E-4)	Ja	Ja	Ja	Nein	Ja	Ja

Tabelle 3.1: Überblick über die Anforderungen und den Stand der Technik

3.9 Auswahl der Entwurfsgrundlage für das entwickelte Protokoll

Die durchgeführte Analyse des Standes der Technik in diesem Kapitel zeigt, dass das in Abschnitt 3.3 beschriebene Protokoll LLDP die größte Schnittmenge mit den Anforderungen aufweist. Daher wurde dieses als Entwurfsgrundlage für das entwickelte LAN-Beacon Protokoll gewählt. Für dieses werden in dem folgenden Kapitel 4 Erweiterungen und Änderungen von LLDP definiert, um sämtliche Anforderungen zu erfüllen.

Das größte Problem im produktiven Einsatz ist die Eigenschaft, dass LLDP-Rahmen von einem LLDP-Agent bei Empfang nicht weitergeleitet werden und eine zuverlässige Identifikation (Anforderung N-6)) der VLANs somit nicht möglich ist. Hierfür müssten die Informationen auf allen aktiven LLDP-Agents eingepflegt werden, was zu einem hohen Installations- und Wartungsaufwand führt und somit Anforderungen B-1) und B-3) widerspricht. Zudem ist die Menschenlesbarkeit nicht gegeben, welche nach Anforderung N-3) verlangt wird. Eine Möglichkeiten zur Authentifizierung, wie sie in Anforderung N-4) und N-7) gefordert wurde, ist ebenfalls nicht vorhanden.

Der Großteil der von LLDP nicht erfüllten Anforderungen konnte durch die anderen Protokolle allerdings ebenfalls nicht erfüllt werden. Ein Versand in menschenlesbarer Form ist bei DHCP in Freitextfeldern möglich, allerdings in dieser Form noch nicht implementiert. Die zuverlässige Identifikation ist in den Protokollen LLTD, DHCP und in dem Patent über die "Identifikation durch Anfrage" prinzipiell möglich. Für Installation und Betrieb ist bei den Verfahren "Auslesen von VLAN-Tags" und dem darauf aufbauenden "VLAN-Scanner" kein weiterer Aufwand notwendig. In den Protokollen LLTD, DHCP und nach dem Patent über die "Identifikation durch Anfrage" ist der Aufwand jeweils geringer als bei LLDP, da nur eine Server-Instanz gepflegt werden muss. Eine Möglichkeit zur Authentifizierung wurde in keinem der analysierten Protokolle angeboten.

Insbesondere ist die leichte Erweiterbarkeit von LLDP ein wichtiges Kriterium, da dies von keinem anderen Protokoll in einer ähnlich leicht umsetzbaren Art und Weise angeboten wird. Die zu erfüllenden Anforderungen können somit zum Großteil über die Integration von zusätzlichen Informationen in neuen TLVs abgebildet werden. Ein Teil der Implementierung ist somit kompatibel zu bestehenden LLDP-Agents, was die Integration in das Protokoll und bestehende Infrastrukturen erleichtert. Zudem ist durch die Standardisierung und weite Verbreitung eine große Menge an Dokumentation und Implementierungen vorhanden, was bei der weiteren Entwicklung hilfreich ist.

4 Konzeption und Entwurf des Protokolls

Das Protokoll wird auf Basis der in Kapitel 2 gewonnenen Anforderungen und in Anlehnung an den in Kapitel 3 beschriebenen Stand der Technik entwickelt. Hierbei dient das Link Layer Discovery Protocol als Grundlage für den Protokollentwurf, welches einen Großteil, aber nicht die Gesamtheit der Anforderungen erfüllt, wie in Abschnitt 3.3 gezeigt wurde.

Das Protokoll wird dahingehend erweitert, dass die Menschenlesbarkeit auch bei der Verwendung von Standardprogrammen gegeben ist. Die mangelnde Möglichkeit zur Authentifizierung wird durch Signaturen über die LAN-Beacon PDU gelöst, als Abwehrmaßnahmen gegen einen Angriff durch Wiedereinspielung sind Zeitstempel und ein Challenge-Response-Verfahren vorgesehen.

Ein Großteil der Anforderungen konnte durch standardkompatible Erweiterungen von LLDP erfüllt und die Syntax beibehalten werden. An anderen Stellen musste vom Standard allerdings abgewichen und dieser modifiziert werden. Die Gründe für die Modifikationen und deren Auswirkungen werden in diesem Abschnitt beschrieben. Der theoretische Entwurf des Protokolls dient für die Implementierung des Prototypen in Kapitel 5.

4.1 Datenmodell und neu definierte TLVs

Für eine erfolgreiche Kommunikation und einfache Nutzbarkeit des LAN-Beacon Protokoll müssen verschiedene Informationen übertragen werden. In Anforderung E-1) wurde die Notwendigkeit der Kompatibilität bei Erweiterung des Protokolls genannt, wobei Erweiterungen laut Anforderung E-2) möglichst einfach zu implementieren sein sollten. Das in Abschnitt 3.3 beschriebene Prinzip der Übertragung von Informationen in TLVs erfüllt diese beiden Anforderungen vollständig. Daher orientiert sich die Syntax der LAN-Beacon PDUs an der von LLDP.

Um sämtliche in Anforderung N-5) verlangten und in Abschnitt 2.2.2 näher definierten Informationen abbilden zu können, wurden die in Tabelle 4.1 aufgeführten TLVs definiert. Die Typ-ID ist frei wählbar, es muss lediglich auf eine eindeutige Zuordnung innerhalb des in Abschnitt 3.3 beschriebenen organisationsspezifischen Identifikators geachtet werden. Für diesen wurden die Buchstaben 'L', 'M' und 'U' gewählt, wobei dem Standard IEEE 802 entsprechend das zweite Bit des ersten Buchstabens auf 1 gesetzt wurde. [IEE14b, 9.3].

Beschreibung der TLV	**Subtyp-ID**
VLAN-ID (Binärzahl)	200
VLAN-Name (Menschenlesbarer Freitext)	201
Benutzerdefinierter Text (Menschenlesbarer Freitext)	202
IPv4-Netze (Binärzahlen)	203
IPv6-Netze (Maschinenlesbare Repräsentation)	204
Email-Adresse der Kontaktperson (Validierter Text)	205
Informationen über DHCP (Menschenlesbarer Freitext)	206
Informationen über den Router (Menschenlesbarer Freitext)	207
Authentifizierungsinformationen (Binärzahlen)	216
Textrepräsentation sämtlicher TLV-Inhalte (Menschenlesbarer Freitext)	217

Tabelle 4.1: Zur Übermittlung der Informationen definierte TLVs

Den TLVs für die Authentifizierungsinformationen (216) und die "Textrepräsentation sämtlicher TLV-Inhalte" (217) wurden Werte mit einigem Abstand zu den restlichen IDs zugeordnet, da diesen TLVs jeweils Sonderfunktionen zukommen. Die Inhalte der Authentifizierungs-TLV werden in Abschnitt 4.3 genauer beschrieben, während die Textrepräsentation sämtlicher TLV-Inhalte eine menschenlesbare Darstellung aller übergebenen Informationen inklusive der ansonsten binär übertragenen Informationen wie VLAN-ID und IP-Netze anbietet. Hiermit wird die Anforderung N-3) erfüllt.

Die Inhalte der Authentifizierungsanfrage werden in Abschnitt 4.3.2 beschrieben.

4.2 Zustandsmaschine des Protokolls

Im Betrieb des Protokolls nehmen Client und Server in Abhängigkeit von ihrer Konfiguration und der Interaktion untereinander verschiedene Zustände an. Sobald der Client zum Beispiel sämtliche aktuell im Puffer enthaltenen LAN-Beacon-Rahmen ausgewertet hat, geht er in einen Anzeigemodus über. Die verschiedenen Zustände und die Übergänge dazwischen können anhand von UML Zustandsdiagrammen [OMG09] dargestellt werden.

Die Abbildung 4.1 zeigt das Zustandsdiagramm für den LAN-Beacon Server an. Hierbei gibt es nur zwei Zustände: in seinem Grundzustand versendet der Server alle 5 Sekunden eine nicht-authentifizierte LAN-Beacon-PDU. Wird in diesem Zeitraum jedoch eine Authentifizierungsanfrage empfangen, geht der Server in den Modus zur Authentifizierung des Rahmens über. Sobald der authentifizierte Rahmen erstellt wurde, sendet das Programm ihn und kehrt in seinen Grundzustand zurück. Zu jedem Zeitpunkt kann das Programm auch beendet werden, indem ihm das entsprechende Signal SIGINT zum Beispiel durch die Tastenkombination Strg+C gesendet wird.

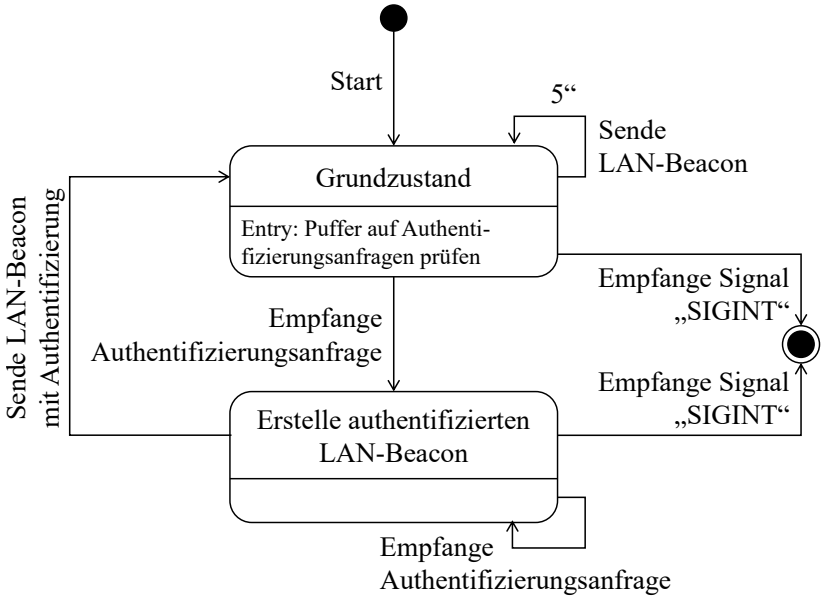

Abbildung 4.1: UML-Zustandsdiagramm des Protokollablaufs

In Abbildung 4.2 sind die verschiedenen Zustände des Client dargestellt. Im Grundzustand werden LAN-Beacon-Rahmen aus dem Puffer gelesen. Sofern diese Authentifizierungsinformationen enthalten, wird der Rahmen mit dem öffentlichen Schlüssel des Servers verifiziert. Schlägt die Verifizierung fehl, wird eine neue Authentifizierungsanfrage an den Server gesendet, ansonsten kehrt der Client in den Grundzustand zurück. Wenn keine Authentifizierungsinformationen enthalten sind, wird in Abhängigkeit vom gewählten Modus eine Authentifizierungsanfrage gesendet oder zurück in den Grundzustand übergegangen.

Sobald im Puffer keine weiteren Rahmen enthalten sind, geht der Client in den Anzeigemodus über. Hierbei werden die in den vergangenen Schritten empfangenen und ausgewerteten Informationen nacheinander ausgegeben, bis alle aktuellen Rahmen abgearbeitet worden sind. Wird die maximale Anzeigehäufigkeit eines Rahmens erreicht, wird dieser aus der Sammlung der aktuellen Rahmen gelöscht und der Speicher freigegeben. Das Programm kann in diesem Modus ebenfalls zu jedem Zeitpunkt beendet werden, indem ihm das entsprechende Signal SIGINT gesendet wird.

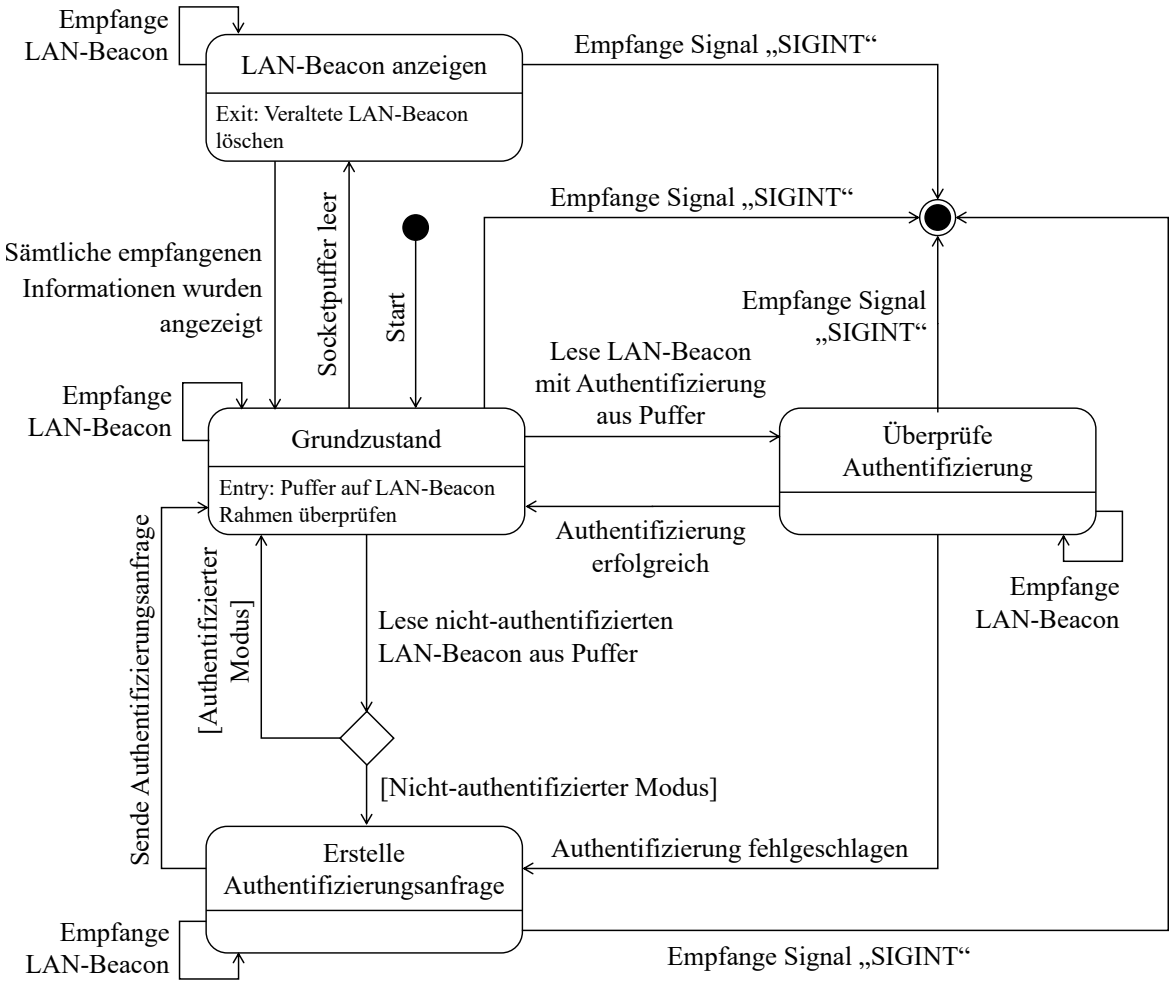

Abbildung 4.2: UML-Zustandsdiagramm des Protokollablaufs

4.3 Beschreibung des Challenge-Response Verfahrens

In Anforderung N-4) und Abschnitt 2.2.1 wurde die Notwendigkeit zur Authentifizierung der durch das Protokoll empfangenen Informationen beschrieben. Hierbei wurden auch die Gründe für die Anforderung N-7) dargestellt, nach der das Verfahren Abwehrmaßnahmen gegen einen Angriff durch Wiedereinspielung und die Möglichkeit einer aktiven Authentifizierung vorsehen soll. Sämtliche in Kapitel 3 beschriebenen Protokolle bieten allerdings nicht einmal grundlegende Sicherheitsmaßnahmen an, einschließlich des hier als Grundlage für die Entwicklung dienenden LLDP.

4.3.1 Gewählte Sicherheitsmaßnahmen

Als erste Anforderung für das Verfahren wurde genannt, dass der zur Authentifizierung versandte Identitätsnachweis nur für den betreffenden Rahmen gültig sein darf. Dies wäre bei einer Authentifizierung durch einen Sicherheitsschlüssel [LL08] zum Beispiel nicht gegeben. Daher fiel die Wahl auf den Versand einer Signatur über die versandten Informationen als

Authentizitätsnachweis.

Hierfür wurde der Algorithmus SHA-256 gewählt, da dieser - wie in Anforderung E-3) gefordert - standardisiert und weit verbreitet ist. [Eas11] Bei der Wahl der Länge des Hashes muss in diesem Anwendungsfall darauf geachtet werden, dass einzelne TLVs eine Größenbeschränkung von 507 Byte Nutzlast haben und ein LAN-Beacon Rahmen maximal eine Länge von 1500 Byte haben kann.

Der Hash des Version SHA-512 kann daher nicht innerhalb einer TLV transportiert werden, während der Hash von SHA-384 bereits ungefähr zusätzliche 10% der maximalen Nutzlast eines Ethernet-Rahmens einnehmen würde. Der weniger verbreitete Algorithmus SHA-224 wurde für hier nicht zutreffende Spezialfälle entwickelt und bietet lediglich eine um vernachlässigbare 32 Byte verringerte Größe der Signatur an. [Hou04]

Um die in Anforderung N-7) beschriebenen Angriffe durch Wiedereinspielung zu verhindern, wurden als Gegenmaßnahmen die Verwendung von Zeitstempeln und eines Challenge-Response-Verfahrens gewählt.

Durch den Zeitstempel kann die Zeitspanne der Gültigkeit eines Rahmens eingeschränkt werden. Die genaue Zeitspanne der Gültigkeit kann vom Client selbst gewählt werden, wobei hier ein Kompromiss zwischen Sicherheit und Fehlertoleranz gemacht werden muss. Aufgrund von unterschiedlich synchronisierten Systemuhren und Latenzen kann kein exakter Zeitpunkt verwendet werden. Der Zeitstempel wird als vorzeichenlose 32-Bit Zahl übertragen, deren Inhalt dem Format der Unix Systemzeit entspricht.

Der Zeitstempel soll die Wiederverwendung von Rahmen verhindern. Da sich aus dem 4 Byte langen Feld $2^{4*8} = 4294967296$ Möglichkeiten für die Challenge ergeben, wäre sogar eine Speicherung der Signaturen für sämtliche Challenges für eine LAN-Beacon PDU in ungefähr 1,12 Terabyte Speicher möglich. Ohne weitere Sicherheitsmaßnahmen könnte ein Angreifer daher beliebig viele Challenges anfragen und auf Vorrat speichern.

Das Challenge-Response-Verfahren stellt eine Möglichkeit zur aktiven Authentifizierung dar und dient hierbei als zusätzliche Sicherheitsmaßnahme. Die zu sendende 32-Bit Zahl ist in der Authentifizierungsanfrage des Client enthalten.

Dieses Vorgehen soll Schwachstellen verringern, die sich bei falscher Systemzeit oder durch die Weiterleitung von Rahmen aus einem freundlichen Netz ergeben, bietet dabei allerdings ebenfalls keinen vollkommenen Schutz: zum Beispiel könnte ein Angreifer innerhalb eines kurzen Zeitrahmens die Authentifizierungsanfrage eines Clients in einem anderen Netzsegment an einen Server schicken, von diesem die Antwort dann wieder an das ursprüngliche Netzsegment weiterleiten und somit dem Client den Anschluss an ein sicheres Netz vorspielen.

4.3.2 Ablauf des Authentifizierungsverfahrens

Der Ablauf der Kommunikation in dem Challenge-Response-Verfahren ist in Abbildung 4.3 dargestellt. Nachfolgend werden die einzelnen Schritte beschrieben:

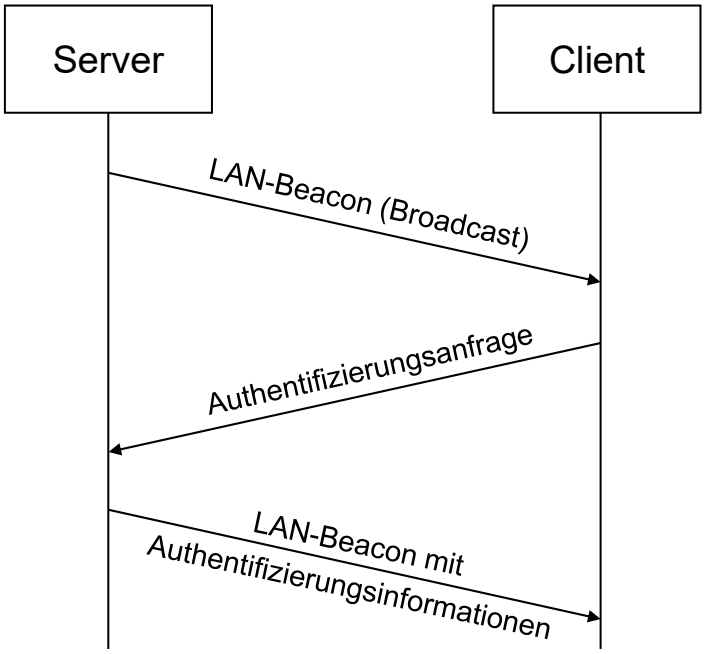

Abbildung 4.3: Ablauf des Challenge-Response-Verfahrens

Sofern der Client eine Authentifizierung wünscht, sendet er dem Server eine Anfrage zur Zusendung eines authentifizierten LAN-Beacon. Hierbei wird ein Rahmen - wie in Abbildung 4.4 dargestellt - mit dem EtherType 0x88-B6 per Unicast an die Adresse gesendet, von welcher der zu authentifizierende LAN-Beacon empfangen wurde. Der Rahmen enthält neben den in Ethernet vorgesehenen Adressen und der Prüfsumme lediglich ein 32 Bit langes Feld mit der Challenge.

Ziel-MAC-Adresse (Server)	Quell-MAC-Adresse (Client)	Challenge (4 Byte, positiv)	Prüfsumme

Abbildung 4.4: Format der Authentifizierungsanfrage inklusive Challenge, die an den Server geschickt wird

Der Server antwortet bei Erhalt der Authentifizierungsanfrage mit einem authentifizierten LAN-Beacon per Unicast, der - wie in Abbildung 4.5 ersichtlich - zusätzlich zu den üblichen Inhalten die Authentifizierungsinformationen enthält.

Hierfür wird eine neue definierte TLV mit den Inhalten Challenge, Zeitstempel und Signatur verwendet. Die Signatur wird mit einem privaten Schlüssel des Servers über den gesamten Inhalt des Rahmens ab dem EtherType bis zum Zeitstempel erstellt. Der Rest des Headers, also die Ziel- und die Quell-MAC-Adressen, werden nicht signiert. Dies soll eventuellen Problemen vorbeugen, die durch das Einfügen von VLAN-Tags entstehen könnten. Zusätzlich

bieten diese nur einen geringen Grad an zusätzlicher Sicherheit, da ein potentieller Angreifer seine MAC-Adresse leicht manipulieren und an die des ursprünglichen Senders angleichen kann.

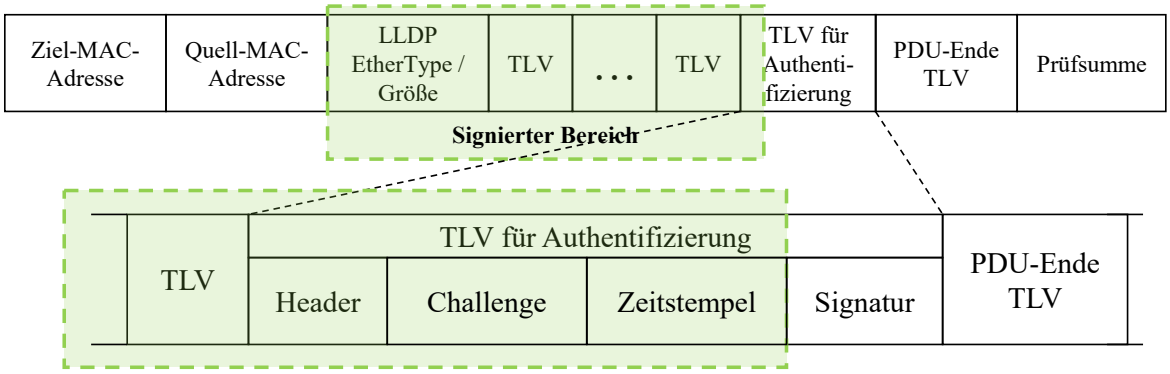

Abbildung 4.5: Inhalte einer TLV für Authentifizierungsinformationen und signierter Bereich

Der Client kann den vom Server erhaltenen LAN-Beacon verifizieren, indem er die Challenge und den Zeitstempel auf Korrektheit prüft und anschließend die Signatur mit dem öffentlichen Schlüssel des Servers verifiziert. Wird nach einer bestimmten Zeitspanne kein gültig authentifizierter LAN-Beacon Rahmens empfangen, so sendet der Client erneut eine Anfrage mit einer neuen Challenge an den Server.

Im Rahmen dieser Bachelorarbeit wird die Verfügbarkeit von Schlüsselpaaren daher vorausgesetzt; der zu dem jeweiligen privaten Schlüssel auf Serverseite gehörige öffentliche Schlüssel muss bei den authentifizierenden Clients also bereits vorhanden sein. Ein Versand von Zertifikaten, Schlüsseln oder weiteren Informationen bezüglich des öffentlichen Schlüssels sind im aktuellen Umfang des Protokolls daher nicht vorgesehen.

Für eine zuverlässige und skalierbare Möglichkeit zur Verifizierung sollte demzufolge eine Public-Key-Infrastruktur vorhanden sein, worauf in Abschnitt 5.7.3 genauer eingegangen wird.

4.4 Definition wohlgeformter Protokolldateneinheiten

Die genaue Syntax valider Rahmen ist durch die folgende formale Grammatik im eBNF-Format gegeben. [Int96] Diese definiert die Syntax eines LAN-Beacon-Rahmens, wobei die entworfenen Subtypen und Inhalte in Abschnitt 4.1 beschrieben werden.

Die in der Aufzählung 4.1 definierten LAN-Beacon-Rahmen enthalten abgesehen von der PDU die in Ethernet-Rahmen verlangten Informationen. Die PDU enthält die in Abbildung 3.4 dargestellten TLVs, die jeweils durch ihre Typ- und Länge-Informationen und den organisationsspezifischen Identifikator und Subtyp beschrieben werden. Das Feld ”Inhalt” für die Nutzlast jeder TLV, also der ”TLV information string” (siehe Abschnitt 3.3), kann bis zu 507 Byte lang sein und wird dem LLDP-Standard entsprechend [IEE09, 8.4.3 c)3)] in UTF-8 kodiert. Hierin sind auch nicht druckbare ASCII-Zeichen enthalten.

```
Rahmen = Ethernet-Header, LAN-Beacon PDU, Prüfsumme
  Ethernet-Header = Ziel-Adresse, Quell-Adresse, EtherType
    Ziel-Adresse = MAC-Adresse
    Quell-Adresse = MAC-Adresse
    MAC-Adresse = 48 * Binärziffer

    EtherType = '0x88B6'

  LAN-Beacon PDU = TLVs, PDU-Ende-TLV
    TLVs = TLV | TLV, TLVs
      TLV = TLV-Typ, TLV-Länge, Organisations-Kennung,
            Organisationsspezifischer Subtyp, Inhalt
        TLV-Typ = 7 * Binärziffer
        TLV-Länge = 9 * Binärziffer
        Organisations-Kennung = 24 * Binärziffer
        Organisationsspezifischer Subtyp = 8 * Binärziffer
        Inhalt = 507 * [ Byte ]

    PDU-Ende-TLV = '0x0000'

  Prüfsumme = 32 * Binärziffer

  Byte = 8 * Binärziffer
  Binärziffer = '0' | '1'
```

Listing 4.1: Valide Syntax von LAN-Beacon Rahmen

Wie man in Aufzählung 4.2 sehen kann, ist die in Abschnitt 4.3.2 beschriebene Authentifizierungsanfrage wesentlich einfacher aufgebaut, da sie neben dem Ethernet-Header und der Prüfsumme lediglich 4 Byte für die Challenge selbst enthält.

```
Rahmen = Ethernet-Header, Challenge, Prüfsumme
  Ethernet-Header = Ziel-Adresse, Quell-Adresse, EtherType
    Ziel-Adresse = MAC-Adresse
    Quell-Adresse = MAC-Adresse
      MAC-Adresse = 48 * Binärziffer

    EtherType = '0x88B5'

  Challenge = 32 * Binärziffer
  Prüfsumme = 32 * Binärziffer
  Binärziffer = '0' | '1'
```

Listing 4.2: Valide Syntax von LAN-Beacon Authentifizierungsanfragen

4.5 Anpassung von Multicast-Adresse und EtherType

Im Rahmen der Evaluierung eines Zwischenstandes des Prototypen wurde festgestellt, dass manche Switches LLDP-Rahmen nicht weiterleiten. Dieses Verhalten konnte nur bei Switches mit erweitertem Funktionsumfang festgestellt werden. Das Problem waren solche Geräte, auf denen ein LLDP-Agent installiert und aktiviert war. Laut Webseite des Herstellers des getesteten Gerätes ist dieses Verhalten so vorgesehen. [HP15] Wie in Abschnitt 3.3 beschrieben entspricht dieses Verhalten dem LLDP-Standard [IEE09, 8.6 d)]. Zudem ist die Weiterleitung von Informationen aus organisationsspezifischen TLVs nicht vorgesehen.

Daher war die Anforderung N-6) nicht erfüllt, da die versandten Rahmen nicht zuverlässig empfangen werden konnten. Die Installation von einem LAN-Beacon Server auf sämtlichen Switches wäre nicht möglich gewesen und hätte den Anforderungen B-1) und B-3) bezüglich unaufwändigem Betrieb und einfacher Installation widersprochen. Daher musste ein Weg gefunden werden, um die Weiterleitung der Rahmen durch sämtliche Switches sicherzustellen.

In dem betrachteten Fall wurde von dem Switch die Entscheidung über die Weiterleitung der Rahmen lediglich auf Basis der gewählten Multicastadresse getroffen. Da dieses Verhalten im Standard aber nicht genauer spezifiziert ist, könnten andere Produkte die Entscheidung über die Weiterleitung allerdings auch auf Basis des EtherTypes treffen.

Daher wurde sowohl die Zieladresse von der Multicastadresse 01:80:C2:00:00:0E auf die Broadcastadresse FF:FF:FF:FF:FF:FF geändert, als auch anstelle des EtherType 0x88-CC der lokale, experimentelle Wert 0x88-B5 verwendet [Int17]. Hiermit werden die Rahmen innerhalb der gesamten Broadcast-Domäne weitergeleitet und sind somit an sämtlichen Anschlüssen innerhalb eines VLAN verfügbar.

Diese Modifikationen führen allerdings dazu, dass die beschriebene Lösung in dieser Form nicht mehr dem Standard entspricht und eine Kommunikation mit anderen LLDP-Agents nicht mehr gegeben ist. Obwohl die Syntax von LLDP beibehalten wird, handelt es sich daher mangels Kompatibilität faktisch um ein neues Protokoll.

4.6 Fehlerfälle und -behandlung

Im Rahmen der Kommunikation mit dem LAN-Beacon Protokoll kann es zu verschiedenen Fehlern kommen. Da die Übertragung über Ethernet nur eine einfache Prüfsumme und keine weiteren Möglichkeiten zur Fehlerbehandlung vorsieht, ist im Gegensatz zu verbindungsorientierten Protokollen wie TCP der Verlust von PDUs ohne Fehlermeldung möglich. Bleibt zum Beispiel die Antwort eines Kommunikationspartners aus, muss daher im Protokoll das zu wählende Vorgehen vorgegeben sein.

Zur Erkennung der Fehler kann neben der Ethernet-Prüfsumme die in Abschnitt 4.4 definierte Grammatik verwendet werden. Ebenfalls als ungültig gelten solche Rahmen, die ungültige Authentifizierungsinformationen enthalten, wie sie in Abschnitt 4.3.2 beschrieben wurden. Dies ist der Fall, wenn der Zeitstempel eine zu alte oder zu neue Uhrzeit enthält, die Challenge nicht dem angefragten Wert entspricht oder die Überprüfung der Signatur mit dem öffentlichen Schlüssel des Servers fehlschlägt.

Da fehlerhafte Rahmen verworfen werden, entspricht die Behandlung dem kompletten Verlust des betreffenden Rahmens. Dieses Vorgehen ist so ebenfalls im LLDP-Standard vorgesehen [IEE09, 6.6.1]. Diese Fehler können hierbei zu unterschiedlichen Zeitpunkten im Protokollablauf auftreten, wie dies in Abbildung 4.6 dargestellt ist:

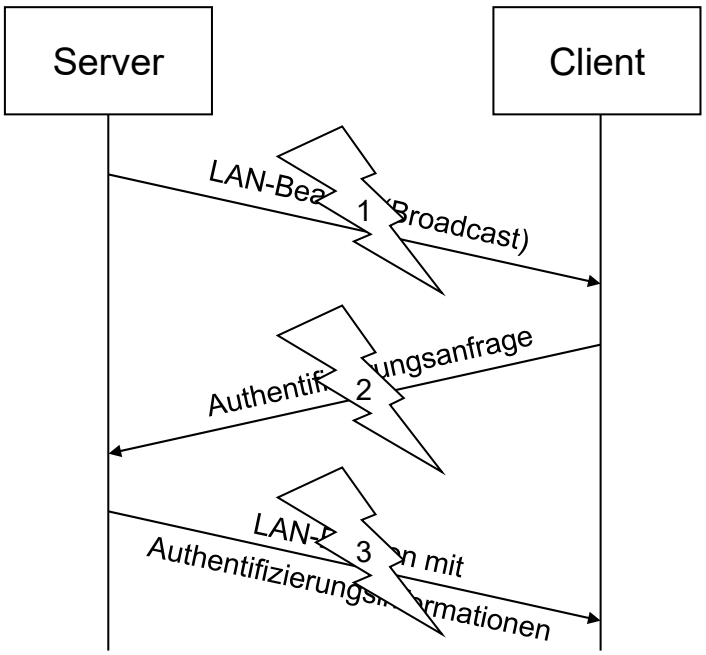

Abbildung 4.6: Mögliche Verbindungsprobleme

Falls ein Verlust in der mit 1 markierten Phase "LAN-Beacon (Broadcast)" passiert, muss auf einen erneuten Empfang zu einem späteren Zeitpunkt gewartet werden. Sofern zuvor noch keine Rahmen empfangen wurden, kommt keine Kommunikation zustande. Ein Problem stellt hierbei dar, dass der Client fälschlicherweise davon ausgeht, sich in einem anderen VLAN zu befinden als der Sender.

Wenn von Clientseite die Authentifizierung gewünscht ist, ergeben sich weitere mögliche Fehlerfälle. So kann in der mit 2 markierten Phase "Authentifizierungsanfrage" ein Fehler bei der Übertragung der Challenge passieren. Durch die 32 Bit lange Prüfsumme wird aufgrund der geringen Länge der Challenge der überwiegende Teil der Fehler erkannt. Wird die Authentifizierungsanfrage auf der Serverseite verworfen, so sendet dieser keinen authentifizierten LAN-Beacon Rahmen. Sollte dennoch eine andere Zahl empfangen und als gültig angenommen werden, so sendet der Server aufgrund der fehlerhaften Challenge einen von Clientseite insgesamt als ungültig befundenen Rahmen. In beiden Fällen kommt es hierbei zu demselben Effekt, wie er im folgenden Fehlerfall beschrieben ist.

Zuletzt kann es zu der fehlerhaften Übermittlung oder dem Verlust von einem LAN-Beacon mit Authentifizierungsinformationen kommen, was als Fehlerfall 3 in der Phase "LAN-Beacon mit Authentifizierungsinformationen" markiert wurde. Da die authentifizierten LAN-Beacon Rahmen allerdings pro Anfrage nur einmal gesendet werden, wartet der Client vergeblich auf eine erfolgreiche Authentifizierung von Server-Seite und fragt daher nach einer bestimmten Zeitspanne erneut die Authentifizierung beim Server an. Solange noch keine gültig authentifizierten PDUs empfangen wurden, kann der Client die empfangenen Daten nicht verifizieren und muss sie daher im authentifizierten Modus als nicht vertrauenswürdig behandeln.

5 Implementierung und Evaluierung

Das in Kapitel 4 beschriebene LAN-Beacon Protokoll wurde im Rahmen der Bachelorarbeit in C implementiert. In diesem Kapitel soll ein Überblick über Implementierungsentscheidungen und Erkenntnisse gegeben werden, die für zukünftige Implementierungen nützlich sein könnten. Anschließend wird die Lösung auf einem Einplatinencomputer eingerichtet und anhand eines realistischen Anwendungsfalls und der in Kapitel 2 beschriebenen Anforderungen evaluiert.

5.1 Verwendung fremder Programme und Bibliotheken

Wie in Anforderung E-4) verlangt und in Abschnitt 2.1.3 genauer beschrieben, soll das entwickelte Programm möglichst portabel sein und daher möglichst geringe externe Abhängigkeiten haben. Dies gilt sowohl für die Betriebssysteme, als auch für die Hardwareplattformen, auf denen der entwickelte Prototyp lauffähig sein soll. Von dieser Anforderung ist die Wahl verwendeten Bibliotheken als auch die Abhängigkeit von anderen Programmen im Betrieb beeinflusst.

Bei der Entwicklung wurde - soweit möglich - auf die Verwendung von wenigen und verbreiteten Standardbibliotheken geachtet. Die Bibliotheken sind daher großteils C-Standard-Bibliotheken oder entsprechen zumindest den Posix-Standards. Daher sollten die meisten Bibliotheken in einer kompatiblen Version auf Unix-System verfügbar sein, während für eine Portierung auf Windows eine überschaubare Menge an Änderungen notwendig sind.

Die wichtigste Ausnahmen stellt hierbei die Netzkommunikation dar, welche in anderen Unix-Derivaten wie BSD oder auch auf Windows-Systemen nicht identisch in der Verwendung sind und zum Teil Einschränkungen unterliegen. Dies gilt insbesondere für die Linux Raw-Sockets [SFR03], die für den Versand der Ethernet-Rahmen verwendet wurden.

Für die Implementierung der in Abschnitt 5.6 beschriebenen Lösung wird zudem der Bildschirm über eine herstellerspezifische Schnittstelle angesprochen und die Ausgabe daher lediglich von diesem unterstützt. Die Ausgabe ist allerdings in eine separate Funktion ausgelagert, an die die Daten in bereits formatierter Form übergeben werden. Die Ausgabe kann daher leicht auf andere Schnittstellen portiert werden, ohne tiefgreifende Kenntnisse des Programmes zu besitzen zu müssen.

Zusätzlich wurde die Anbindung an eine LLDP-Agent-Software überprüft. Hierfür wurde in der Anfangsphase der Versand der selbst definierten LLDP-TLVs über das frei verfügbare Programm LLDPD [Ber] umgesetzt.

Daher wurde für die Versandfunktionalitäten die Einbindung einer LLDP-Bibliothek untersucht, um während des Betriebs lediglich den LAN-Beacon Prototyp selbst ausführen zu müssen. Sofern die Bibliotheken statisch gelinkt werden, hat das Programm innerhalb einer Rechnerarchitektur dann keine Abhängigkeiten von anderen installierten Paketen.

Diese beiden möglichen Vorgehensweisen wurden jedoch aus folgenden Gründen verworfen:

Verfügbarkeit des Agent-Programmes: Es kann nicht davon ausgegangen werden, dass das Programm auf dem zum Versand oder Empfang gewählten System verfügbar ist. Zudem muss für eine Verwendung des LAN-Beacon Prototypen dann auch das Agent laufen. Die Abhängigkeit von der Verfügbarkeit dieses oder eines anderen Programmes hätte die Einsatzmöglichkeiten also möglicherweise eingeschränkt.

Ressourcenknappheit: Die Verwendung eines separaten LLDP-Servers oder einer solchen Bibliothek hätte zusätzlichen Ressourcenaufwand bedeutet. Dieser rührt von dem erweiterten Funktionsumfang solcher Software her, welcher nicht benötigt wird.

Obwohl dieser Zusatzaufwand auf den meisten Systemen unerheblich ist, muss dies nicht auf sämtlichen Betriebsumgebungen gegeben sein: ein Beispiel dafür sind ein in späteren Portierungen unterstütztes, eingebettetes System oder ein Kleinrechner, auf dem mehrere Dienste laufen und infolgedessen jeder Dienst sparsam mit den Ressourcen umgehen muss.

Gestaltungsfreiheit für das Protokolls: Durch die Einbindung eines bestehenden Programms wäre die Gestaltungsfreiheit beim Entwurf des Protokolls eingeschränkt gewesen: zum einen wäre die Signierung der gesamten PDU anstatt lediglich eines Teiles der TLVs wesentlich schwerer umsetzbar gewesen.

Zusätzlich hätte man für eine Änderung des EtherTypes und der Ziel MAC-Adresse den Quellcode anpassen müssen. Bei einem Update der Software oder der Bibliotheken hätte man diese Änderung erneut durchführen müssen.

Änderungen der Schnittstelle und des Programmverhaltens: Bei der Verwendung eines externen Programmes ist es nicht ausgeschlossen, dass sich bei diesem in zukünftigen Versionen die Schnittstelle und das Programmverhalten ändern. In diesem Fall müsste dann auch der Code der LAN-Beacon Implementierung angepasst werden. Eine alleinstehende Implementierung ermöglicht daher einen geringeren zukünftigen Wartungsaufwand, da Ethernet ein etablierter Standard ist und Raw-Sockets sehr wahrscheinlich auch in Zukunft eine stabile Schnittstelle anbieten.

5.2 Struktur und Ablauf des Programmes

Das Programm wurde zur leichteren Wartung und Erweiterung (siehe Anforderung E-2) möglichst modular aufgebaut und der Quelltext in möglichst selbstständige und wiederverwendbare Komponenten aufgeteilt. Der Zusammenhang ist in Abbildung 5.1 dargestellt, wobei die Rechtecke Funktionen und die Pfeile deren Aufruf darstellen.

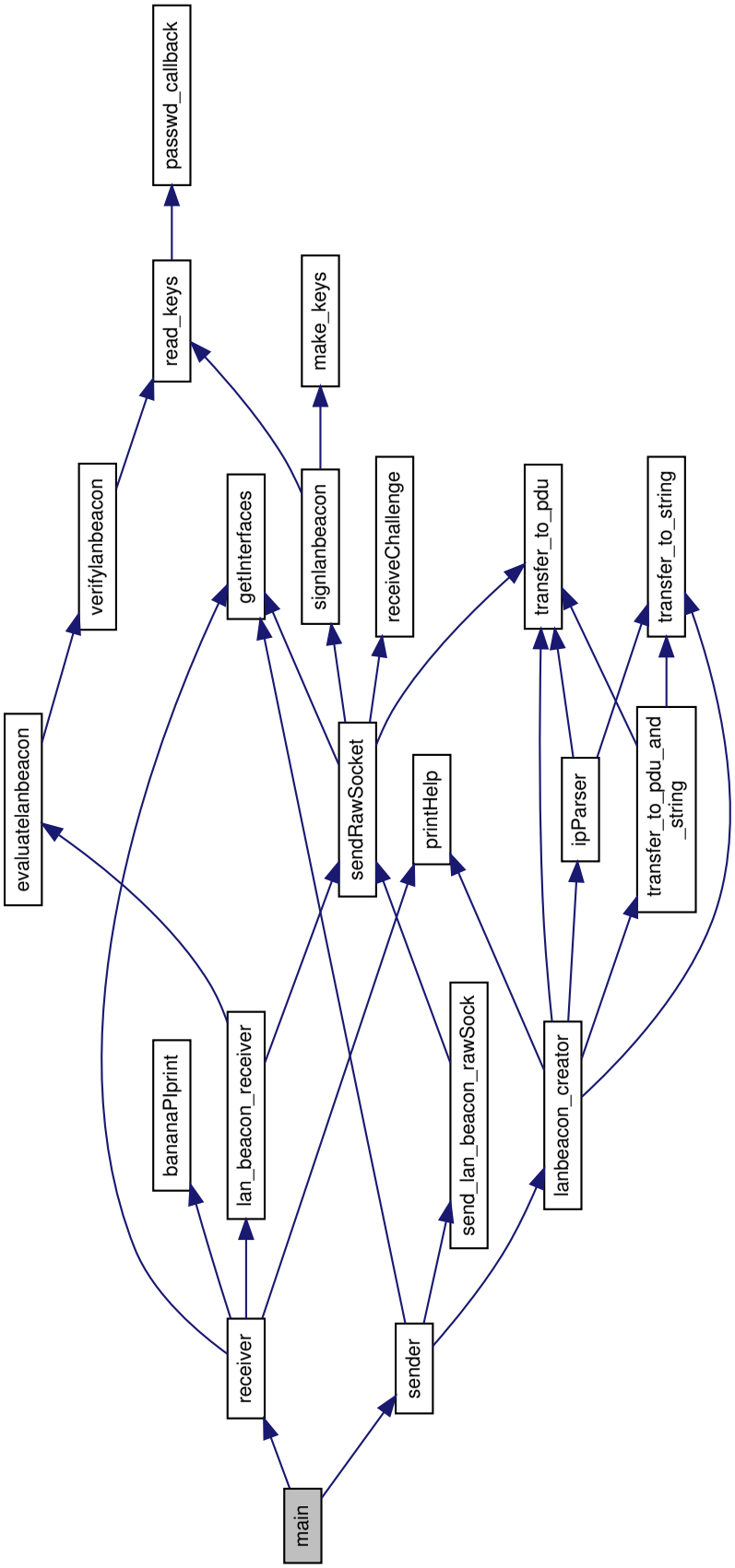

Abbildung 5.1: Architekturskizze der LAN-Beacon Implementierung

Eine Übersicht über die Funktionsweise des Programmes kann daher anhand einer Beschreibung der wichtigsten Funktionen und deren Umfang dargestellt werden. Im Folgenden sind diese daher abstrakt beschrieben, wobei sich hierbei an der Aufteilung in Empfängermodus (Funktion `receiver`) und Sendermodus (Funktion `sender`) durch die Funktion `main` orientiert wird.

5.2.1 Funktionsweise des Empfängermodus

Im Anfangszustand des Empfängers übernimmt die Funktion `lan_beacon_receiver` die Kommunikation. Hierbei wird am Anfang auf den Empfang von LAN-Beacon Rahmen gewartet, deren TLVs anschließend in der Funktion `evaluatelanbeacon` ausgewertet werden. Sofern Authentifizierungsinformationen enthalten sind, wird mit der Funktion `verfiylanbeacon` die Signatur verifiziert.

Wenn keine Authentifizierungsinformationen enthalten sind, der Client aber im authentifizierten Modus gestartet wurde, wird mit der Funktion `sendRawSocket` eine Authentifizierungsanfrage versendet. Hierbei erfolgt der Empfang von LAN-Beacon-Rahmen über eine Schleife, die so lange ausgeführt wird, bis sämtliche Rahmen aus dem Puffer ausgelesen wurden.

Anschließend werden die empfangenen Rahmen mit der Funktion `lan_beacon_print` ausgegeben. Sobald sämtliche Rahmen angezeigt worden sind, geht das Programm wieder in den Anfangszustand und damit zum Auslesen neuer Rahmen aus dem Puffer über.

5.2.2 Funktionsweise des Sendermodus

Beim Start des Programmes werden mit der Funktion `lanbeacon_creator` die Startparameter des Programmes ausgewertet und mit den Inhalten eine LAN-Beacon PDU erstellt. Hierbei werden verschiedene Hilfsfunktionen verwendet wie zum Beispiel `ipParser` zur Auswertung der IP-Subnetze oder die Funktionen `transfer_to_pdu`, `transfer_to_string` und `transfer_to_pdu_and_string` zur Übertragung der Inhalte in die PDU.

Hieran anschließend geht der Server zum Empfang des LAN-Beacons mit der Hilfsfunktion `send_lan_beacon_rawSock` über, die der Funktion `sendRawSocket` die nötigen Daten und Konfigurationen übergibt. Hiermit werden in konfigurierbaren Zeitabständen die PDUs versendet.

Wenn die Funktion `receiveChallenge` in der Zwischenzeit Authentifizierungsanfragen erhält, so werden die Authentifizierungsinformationen hinzugefügt und der LAN-Beacon mit der Funktion `signlanbeacon` signiert.

Nach dem Versand der authentifizierten TLV geht das Programm wieder zum periodischen Versand der nicht-authentifizierten LAN-Beacon an die Broadcast-Adresse über und ließt zwischenzeitlich empfangene Authentifizierungsanfragen aus oder wartet auf solche.

5.3 Datenmodell

Für den Betrieb des Clients und des Servers werden verschiedene Informationen und Daten benötigt, die auf den jeweiligen Geräten für ordnungsgemäßen Betrieb gehalten werden müssen. Dies gilt sowohl für die Sender- als auch für die Empfängerseite. Neben den beim Start übergebenen Konfigurationen ist ein Grund hierfür zum Beispiel auch das Authentifizierungsprotokoll, da dieses wegen der Challenge nicht zustandslos ist. Abgesehen davon

müssen auf der Senderseite die Informationen zum Versand und auf der Empfängerseite zum Anzeigen gehalten werden.

Die während der Kommunikation anfallenden Daten selbst müssen von Protokollseite her allerdings nicht über mehrere Kommunikationsabläufe gespeichert werden. Daher sind keine Konfigurationsdateien vorgesehen, lediglich im Fall der Authentifizierung müssen der private bzw. öffentliche Schlüssel gelesen werden. Die Informationen werden daher lediglich im Hauptspeicher vorgehalten. Abhängig von dem gewählten Modus werden hierfür unterschiedliche Datenstrukturen angelegt.

Um die beschriebenen Anforderungen erfüllen zu können, sind die folgenden Informationen notwendig. Hierbei wird auf die Beschreibung von selbsterklärenden Inhalten wie der Länge von Speicherfeldern und Hilfsinformationen (z.B. `lan_beacon_pdu_len`) verzichtet.

Auf Senderseite gehaltene Daten: (siehe Abbildung 5.2)

- Konfiguration des Servers: (`sender_information`)
  - Schnittstelle, die für den Versand der Informationen spezifiziert worden ist (`interface_to_send_on`)
  - Inhalte der TLVs, für die bei Start des Programmes Parameter übergeben worden sind (`lanBeacon_PDU`). Die möglichen Inhalte sind in Abschnitt 2.2.2 spezifiziert
  - Sekunden zwischen dem Versand von PDUs (`send_frequency`)
  - Zur Signierung benutzter Schlüssel und zugehörige Informationen: (`open_ssl_keys`)
    - Pfad zum privaten Schlüssel (`path_To_Signing_Key`)
    - Pfad zum öffentlichen Schlüssel, falls dieser erstellt werden soll (`path_To_Verifying_Key`)
    - Passwort für den privaten Schlüssel (`pcszPassphrase`)
    - Indikator, ob Schlüssel erstellt werden sollen (`generate_keys`)
  - Netzschnittstellen zum Empfang der Authentifizierungsanfragen:
    - EtherType der Authentifizierungsanfragen, anhand dessen diese aus dem Netzverkehr gefiltert werden (`etherType`)
- Zusätzlich vorübergehend beim Empfang einer Authentifizierungsanfrage:
  - MAC-Adresse des Clients als Zieladresse für die Antwort
  - In der Authentifizierungsanfrage enthaltene Challenge, um diese in die authentifizierte Antwort an den anfragenden Client einzufügen
  - Aktuelle Uhrzeit, um diese als Zeitstempel in der Antwort an den Client einzufügen

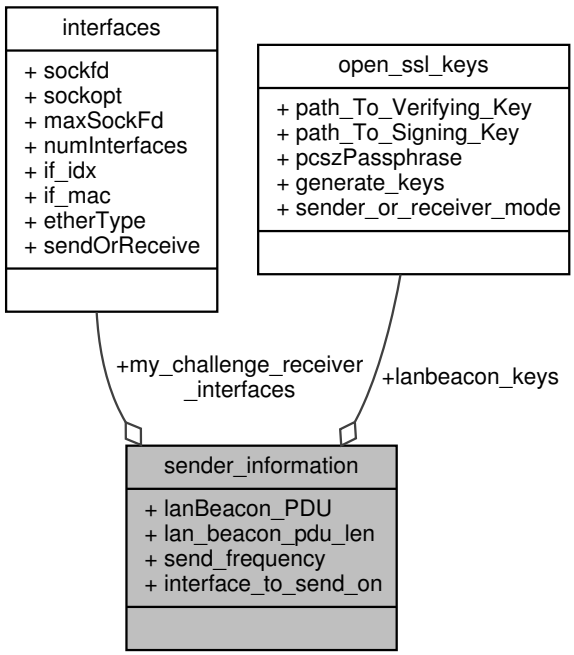

Abbildung 5.2: Für Datenhaltung verwendete Datenstruktur auf Serverseite

Auf Empfängerseite gehaltene Daten (siehe Abbildung 5.3):

- Konfiguration des Clients: (`receiver_information`)
  - Indikator für den beim Start spezifizierten Modus, ob Client eine Authentifizierung verlangen soll
    (`authenticated_mode`)
  - Anzahl der Sekunden, bevor die Anzeige zu den nächsten Informationen übergeht (`scroll_speed`)
  - Empfangene LAN-Beacon Rahmen:
    - Empfangene LLDPDU (`lan_beacon_ReceivedPayload`)
    - Inhalte der TLVs, die aus der LAN-Beacon PDU ausgelesen und für die Ausgabe formatiert worden sind (`parsedBeaconContents`). Die möglichen Inhalte sind in Abschnitt 2.2.2 spezifiziert
    - Die in der Authentifizierungsanfrage für diesen Rahmen gesendete Challenge `challenge`
    - MAC-Adresse des Servers, von dem die LAN-Beacon PDU empfangen wurde (`current_destination_mac`)
    - Indikator, ob die in der PDU enthaltenen Informationen bereits erfolgreich authentifiziert worden sind (`successfullyAuthenticated`)
    - Anzahl, wie oft die Inhalte der PDU noch angezeigt werden sollen. Dieser Countdown wird zurückgesetzt, wenn erneut ein gültig authentifizierter Rahmen empfangen wird (`times_left_to_display`)

- Öffentlicher Schlüssel zur Überprüfung der Authentizität des Servers, an den die Anfrage gesendet wurde (`open_ssl_keys`)
  - Pfad zum öffentlichen Schlüssel (`path_To_Verifying_Key`)
- Netzschnittstellen zum Empfang der LAN-Beacon PDUs:
  - EtherType der LAN-Beacon Rahmen, anhand dessen diese aus dem Netzverkehr gefiltert werden (`etherType`)

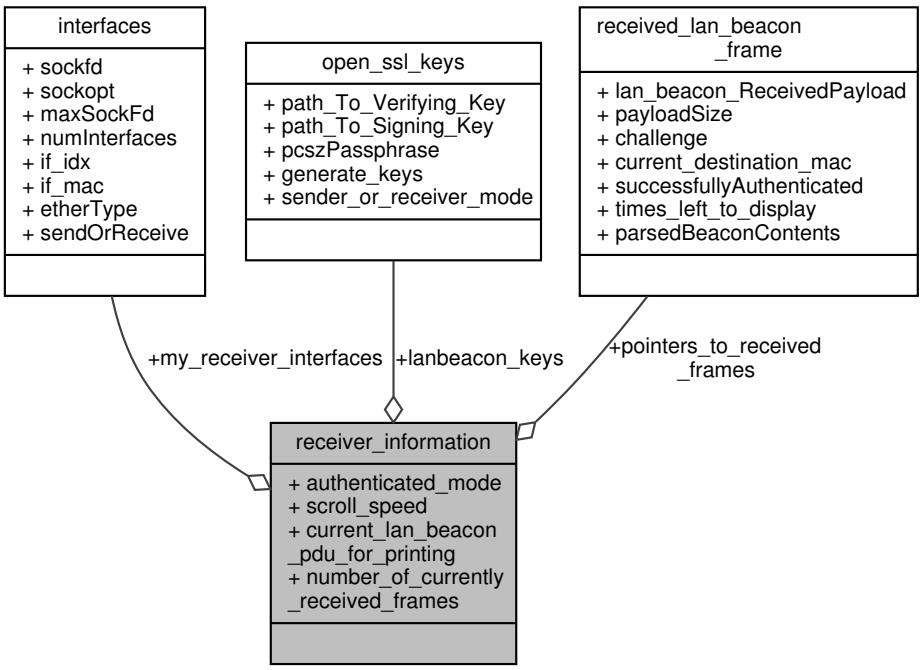

Abbildung 5.3: Für Datenhaltung verwendete Datenstruktur auf Clientseite

5.4 Betriebsumgebungen

Die Implementierung wurde für Linux umgesetzt und unter diesem Betriebssystem getestet. Grundsätzlich sollte eine Portierung auf andere Unix-Systeme leicht möglich sein, wobei einige der Einschränkungen in 5.1 beschrieben sind.

Für die Ausführung sind aufgrund der Verwendung von Raw-Sockets Administratoren-Rechte notwendig. Um die damit einhergehenden Sicherheitsrisiken zu minimieren, könnte die Software auf eine in Abschnitt 5.7.2 beschriebene, alternative Architektur umgestellt werden, bei der nur ein kleiner Teil der Software mit Administratoren-Rechten ausgeführt wird.

Für den Betrieb des Programmes ist abgesehen von der Verteilung der Schlüssel keine weitere Wartung notwendig. Sofern sich etwas an der Konfiguration des Netzes ändert, müssen diese Informationen in das Programm eingepflegt werden. Hierfür wird es beendet und mit den aktualisierten Parametern neu gestartet. Sofern der Start über Skripte automatisiert erfolgt, müssen diese angepasst werden.

Die LAN-Beacon-Implementierung bietet die zwei grundsätzlichen Betriebsmodi **Server** und **Client** an. Sie kann auf verschiedensten Umgebungen betrieben werden, die von dem Anwendungsfall abhängig sind. Im Rahmen von unterschiedlich aufwändigen Portierungen sind zudem auch weitere Betriebsumgebungen möglich, wie zum Beispiel Switches. Daher sind große Unterschiede bezüglich der verwendeten Hardware gegeben. Beispielsweise sind die folgenden Umgebungen denkbar:

Server im Sendermodus: Für den regelmäßigen Versand der LAN-Beacon Rahmen muss innerhalb jeden VLANs eine Instanz des LAN-Beacon Servers laufen, wobei diese durchaus auf dem gleichen physischen Gerät betrieben werden können. Hierfür bieten sich solche Rechner an, die bereits dauerhaft betrieben werden, wie zum Beispiel Dateiserver.

Switch im Sendermodus: Noch besser wäre der Betrieb des Servers auf einem Switch, da dieser auf jeden Fall aktiv sein muss, wenn ein Gerät über diesen Anschluss kommunizieren will. Zusätzlich sind auf diesen Geräten bereits häufig LLDP-Agents aktiv, in die die Funktionalität des LAN-Beacon Prototypen langfristig integriert werden könnte.

Switch oder Server im Empfängermodus: Zusätzlich könnte ein Switch auch im Empfängermodus aktiv sein, um ständig Informationen über den Netzzustand an sämtlichen Anschlüssen zu erhalten. Hieraus können dann Rückschlüsse auf Änderung der Netztopologie gezogen werden. Zusätzlich kann durch Authentifizierungsanfragen überprüft werden, ob ein fremdes und nicht zu der üblichen Ausstattung gehörendes Gerät im Netz LAN-Beacon Rahmen versendet.

Identifikation der transportierten VLANs: Um den ursprünglich in Abschnitt 1.1 definierten Anwendungsfall der Identifikation von an einem Anschluss transportierten VLANs zu erfüllen, kann ein Rechner im Empfängermodus betrieben werden. Dieser wertet dann die empfangenen Informationen aus und führt die Authentifizierung durch. Anhand der Informationen kann er dann die verfügbaren VLANs anzeigen.

Überprüfung zweier Anschlüsse auf gleiches VLAN oder lokales Netz: Selbst wenn in einem VLAN kein LAN-Beacon Server in Betrieb ist, kann überprüft werden, ob zwei Anschlüsse dasselbe VLAN transportieren. Hierfür kann an einem Anschluss eine Instanz im Sendermodus gestartet werden, während an dem anderen Anschluss eine Instanz im Empfängermodus aktiv ist. Sofern bei der zweiten Instanz Rahmen ankommen, sind beide Anschlüsse in demselben lokalen und - falls in solche eingeteilt - auch virtuellen Netz.

Weitere Anwendungsfälle sind möglich, sofern das Protokoll und der Prototyp um zusätzliche Funktionen erweitert werden. Die Voraussetzungen und Möglichkeiten werden hierbei in Abschnitt 6.2 näher beschrieben.

5.5 Verwendete Technologien für Entwicklung und Deployment

Um die Installation, den Betrieb und die Weiterentwicklung der Software möglichst weitgehend zu erleichtern, wurde auf verschiedene Programme zurückgegriffen. Sie gehen jeweils auf unterschiedliche Anforderungen aus Kapitel 2 ein und unterstützen bei deren Lösung.

Ein Beispiel hierfür ist die automatische Konfiguration von Zielsystemen und Softwareabhängigkeiten mit Autoconf, um die Installation wie in Anforderung B-3) verlangt möglichst weitgehend zu erleichtern. Hierbei werden anhand von durch den Entwicklern angelegten Konfigurationen Skripte generiert, die die Fähigkeiten des Zielsystems überprüfen und Installationsverzeichnisse für das Programm, die Dokumentation aber beispielsweise auch die Übersetzungsdateien festlegen. [FSF16a]

Um die Verwendung und die Weiterentwicklung des Programmes zu erleichtern, wurde sowohl die korrekte Verwendung als auch der Quelltext selbst dokumentiert (Anforderung E-2). Hierfür wurde neben der integrierten Hilfefunktion eine man-page erstellt.

Doxygen wurde verwendet, um Schnittstellen und Datenstrukturen direkt im Quelltext zu dokumentieren. Mit dem Programm wurden auch die in Abschnitt 5.2 beschriebenen Abbildungen erstellt. Zusätzlich kann die Dokumentation in verschiedenen Formaten exportiert werden, wie zum Beispiel HTML- oder PDF-Versionen. [vH16] Einige der mit Hilfe dieses Programmes erstellten Grafiken werden in Abschnitt 5.2 dargestellt und beschrieben.

Um die Internationalisierung des Programmes zu erleichtern, wurde GNU gettext verwendet. Hiermit können die Ausgaben des Programmes auch ohne Programmierkenntnisse leicht übersetzt werden. Im Rahmen der Bachelorarbeit wurde hiermit neben der Standardsprache Englisch Unterstützung für die deutsche Sprache integriert. [FSF16b]

5.6 Anwendungsbeispiel: integrierte Lösung auf Basis eines Kleinstrechners mit Bildschirm

Um die Anwendbarkeit von entwickeltem Protokoll und Prototyp in dem in Kapitel 1.1 beschriebenen Anwendungsfall zu zeigen, wurde im Rahmen der Bachelorarbeit der in Abbildung 5.4 dargestellte Kleinrechner mit Bildschirm konfiguriert. In Verbindung mit einer mobilen Stromversorgung - wie sie als Handyladestation verkauft werden - ist man sogar von der Verfügbarkeit eines Stromanschlusses unabhängig. Hiermit kann der Kleinrechner als mobile Einheit zur Identifikation von an einem Anschluss transportierten VLANs verwendet werden.

Abbildung 5.4: Kleinrechner mit Display während der Ausführung des LAN-Beacon Prototypen im Empfängermodus

Die besondere Herausforderung war hierbei die Formatierung von den empfangenen PDUs zur strukturierten Ausgabe, um auf dem begrenzten Platz und bei der niedrigen Auflösung die Information übersichtlich darstellen zu können. Für gute Lesbarkeit wurde eine Ausgabe von maximal 40 mal 15 Zeichen vorgesehen.

Um gesuchte Felder in der Ausgabe schnell finden zu können, werden für alle TLVs die Titel ausgeben und die Inhalte danach mit Einrückungen über mehrere Zeilen umgebrochen. Sofern mehr als die 15 verfügbaren Zeichen benötigt werden, werden die Inhalte über mehrere Seiten aufgeteilt und diese in einem konfigurierbaren Zeitabstand durchgeblättert und mehrfach auf dem Bildschirm ausgegeben.

Auf der Standardausgabe werden zusätzlich noch weitere Informationen ausgegeben, die bei der Ausführung des Programmes an einem Rechner mit vollwertigem Bildschirm oder über SSH sichtbar sind.

5.7 Evaluation

Das praktische Ergebnis der Bachelorarbeit, also der Prototyp und seine Anwendung auf Basis des Kleinrechners, wurde während der Entwicklung mehrfach evaluiert und unter anderem mit den in Abschnitt 2.1 genannten Experten besprochen. Die Ergebnisse dieser Gespräche wurden in die Anforderungen eingebracht, um sämtliche Anwendungsszenarios bestmöglich abzudecken.

Hierbei wurden vonseiten potentieller Anwender auch Probleme aufgezeigt, die beim alltäglichen Einsatz aufkommen, jedoch von Entwicklerseite nicht bedacht worden waren. Ein Beispiel hierfür ist, dass die aktuelle Lösung im Empfängermodus bei der Betrachtung keine Interaktionsmöglichkeiten anbietet. Es werden die Inhalte der empfangenen PDUs nacheinander ausgegeben, wobei lediglich die Anzeigedauer vor dem Wechsel zu den nächsten

Informationen eingestellt werden kann. Mangels Eingabemöglichkeiten sind weitere Interaktionsmöglichkeiten auf der beschriebenen Hardwarebasis nicht möglich.

In diesem Abschnitt wird beschrieben, inwiefern das entworfene Protokoll und der implementierte Prototyp die gestellten Anforderungen erfüllen. Ferner werden Möglichkeiten zur Verbesserung verschiedener Aspekte der Implementierung aufgezeigt, die im Rahmen dieser Bachelorarbeit nicht umgesetzt wurden.

5.7.1 Implementierter Funktionsumfang und Vergleich mit der Anforderungsanalyse

Wie man in Tabelle 5.1 sehen kann, werden sämtliche der in Kapitel 2 beschriebenen Anforderungen erfüllt, manche aufgrund von Designentscheidungen allerdings mit geringfügigen Einschränkungen.

Bei Anschluss eines Gerätes können die Rahmen mit nur geringem Aufwand (Anforderung N-1) über Standardprogramme wie TCPdump oder Wireshark empfangen werden (Anforderung N-2). Da sämtliche Informationen in der TLV 217 in Textform übertragen werden, ist die Ausgabe des Netzverkehrs menschenlesbar (Anforderung N-3). Die genannten zusätzlichen Informationen können in den in Abschnitt 2.2.2 beschriebenen TLVs transportiert werden (Anforderung N-5). Die Rahmen werden mit einem neuen EtherType an die Broadcast-Adresse FF:FF:FF:FF:FF:FF gesendet und damit zuverlässig an sämtliche Anschlüsse weitergeleitet (Anforderung N-6). PDUs werden regelmäßig versendet und sind bei entsprechend eingestellter Frequenz schnell verfügbar (Anforderung N-8).

Das in Abschnitt 4.3 beschriebene Verfahren ermöglicht eine Authentifizierung (Anforderung N-4) und beinhaltet Maßnahmen gegen Angriffe durch Wiedereinspielung (Anforderung N-7). Diese bieten ein grundlegendes Sicherheitsniveau, auch wenn die Maßnahmen gegen fortgeschrittene Angriffe nicht ausreichen, wie sie beispielhaft in Abschnitt 5.7.2 beschrieben werden.

Durch die zuverlässige Weiterleitung aufgrund des geänderten EtherTypes und der Broadcast-Adresse muss nur noch eine Server-Instanz pro VLAN installiert und gepflegt werden, was den Aufwand deutlich senkt (Anforderungen B-1 und B-3). Durch den regelmäßigen Versand von PDUs entsteht eine dauerhafte, aber vernachlässigbar geringe Belastung der Netze und der angeschlossenen Systeme (Anforderung B-2).

Dabei ist eine gute Erweiterbarkeit gegeben, da die Syntax der PDU auf Typ-Länge-Werten basiert und neue TLVs definiert werden können, ohne die Kompatibilität zu alten Implementierungen zu verlieren (Anforderungen E-1 und E-2). Das als Grundlage dienende Protokoll LLDP ist ein weit verbreiteter und ausführlich dokumentierter Industriestandard (Anforderung E-3). Die Implementierung wurde lediglich mit Abhängigkeiten zu den in Abschnitt 5.1 beschriebenen Standardbibliotheken umgesetzt, wobei die Anzeige auf dem Display die einzige Ausnahme ist (Anforderung E-4).

		Kürzel	Anforderung	LAN-Beacon
Nutzerseitig	Funktional	*Notwendig*		
		N-1)	Für die Identifikation der an einem Netzanschluss transportierten VLANs soll ein möglichst geringer Aufwand notwendig sein.	Erfüllt
		N-2)	Die empfangenen Informationen sollen auf Basis von Standardprogrammen wie TCPdump oder Wireshark ausgelesen werden können.	Erfüllt
		N-3)	Die Informationen sollen von den in Anforderung N-2) genannten Standardprogrammen in menschenlesbarer Form ausgegeben werden.	Erfüllt
		N-4)	Die Authentizität des Servers und der empfangenen Informationen sollen verifiziert werden können.	Erfüllt
		Optional		
		N-5)	Die in Abschnitt 2.2.2 aufgezählten weiteren Informationen sollen empfangen werden können.	Erfüllt
	Nicht-funktional	*Notwendig*		
		N-6)	Die Identifikation sämtlicher an einem Netzanschluss transportierten virtuellen LANs soll zuverlässig möglich sein.	Erfüllt
		N-7)	Es sollen Schutzmaßnahmen gegen Angriff durch Wiedereinspielung vorgesehen sein.	Erfüllt
		Optional		
		N-8)	Die Ergebnisse der Identifikation der virtuellen Netze sollen nach möglichst kurzer Zeit zur Verfügung stehen.	Erfüllt
Betreiberseitig	Funktional	*Notwendig*		
		B-1)	Bei der Änderung der Netztopologie soll für die Einpflegung der Informationen ein möglichst geringer Aufwand notwendig sein.	Teilweise erfüllt
	Nicht-funktional	*Notwendig*		
		B-2)	Der dauerhafte Betrieb des Protokolls soll mit einer geringen Auslastung des Netzes und der Ressourcen von Client und Server möglich sein.	Erfüllt
		Optional		
		B-3)	Für die Integration des Protokolls in ein virtuelles LAN soll ein möglichst geringer Aufwand notwendig sein.	Teilweise erfüllt
Entwicklerseitig	Nicht-funktional	*Notwendig*		
		E-1)	Die Integration neuer Funktionen in das Protokoll soll bei gleichzeitigem Erhalt der Kompatibilität möglich sein.	Erfüllt
		Optional		
		E-2)	Für die Integration von neuen Funktionen in das Protokoll soll möglichst wenig Aufwand notwendig sein.	Erfüllt
		E-3)	Das Protokoll soll auf möglichst weit verbreiteten Standards aufbauen.	Erfüllt
		E-4)	Es sollen möglichst wenige Abhängigkeiten von Drittprogrammen und -bibliotheken bestehen.	Erfüllt

Tabelle 5.1: Überblick über die durch Protokoll und Implementierung erfüllten Anforderungen

5.7.2 Mögliche Angriffsszenarien

Die in Abschnitt 4.3 beschriebenen Sicherheitsmaßnahmen dienen vor allem der Verhinderung eines Angriffes durch Wiedereinspielung. Hierbei wurde bereits aufgezeigt, dass das Verfahren keinen sehr starken Schutz bietet. Da die Authentifizierung automatisch erfolgt und die Authentifizierungsanfragen auf der Serverseite in den meisten Fällen wahrscheinlich nicht überwacht werden, muss also die Möglichkeit von Man-in-the-Middle Angriffen beachtet werden.

Zusätzlich kann sich aus der zukünftigen Entwicklung der Bedarf nach längeren Signaturen ergeben. Aufgrund der Tatsache, dass in einer TLV aber lediglich 507 Byte transportiert werden können, ist der hierfür verfügbare Platz begrenzt. Da 512 Byte lange Signaturen bereits nicht mehr möglich sind, sind also bei der Verwendung von SHA-Signaturen 384 Byte die größtmögliche Länge. Eine Aufteilung von zusammengehörigen Inhalten auf mehrere TLVs ist nach dem IEEE-802.1AB Standard nicht vorgesehen. Eine solche Lösung würde folglich eine noch weitere Abweichung von dem Standard bedeuten.

Zusätzlich ergeben sich wie in Abschnitt 5.4 beschrieben Sicherheitsprobleme aus der Ausführung des Prototypen mit Systemrechten. Um die damit einhergehenden Sicherheitsrisiken zu minimieren, könnte die Software in zwei Teile aufgeteilt werden: ein möglichst minimaler Teil läuft für die Kommunikation mit Systemrechten, während ein anderer Prozess mit eingeschränkten Rechten die restlichen Aufgaben übernimmt. Neben anderen Sicherheitsmaßnahmen wird dieses Vorgehen auch in anderen LLDP-Implementierungen verwendet [Ber17]. Im Rahmen der weiteren Entwicklung des Prototypen könnten solche oder andere Maßnahmen integriert werden.

5.7.3 Schlüsselmanagement und -verteilung

Wie in Abschnitt 4.3 beschrieben setzen der entwickelte Prototyp und das Protokoll aktuell die Verteilung von Schlüsseln voraus. Zusätzlich wird in dem Protokoll nicht angegeben, mit welchem Schlüssel ein Rahmen signiert wurde. Beim Einsatz in größeren Netzen stellt dies ein Problem dar, da unterschiedliche Server nicht den denselben privaten Schlüssel verwenden sollten.

Alternativ können Zertifikate in Anlehnung an eine X.509-Infrastruktur in TLVs mitgesendet werden. Dies würde die Größe der PDU allerdings wesentlich erhöhen, da hier zusätzlich zu den Schlüsseln noch weitere Zertifikats-Informationen enthalten sind. Zudem müssten Stammzertifikate (engl. root certificate) auf den Clients verfügbar sein, mit denen sie die Authentizität der Zertifikate sicherstellen können.

Ein weiteres Problem stellen die ständig wachsenden Schlüssellängen dar. Aufgrund der verwendeten Länge der RSA-Schlüssel von 2048 Bit ist ein Transport des öffentlichen Schlüssel mit einer beispielhaft experimentell bestimmten Länge von 451 Bytes (ohne zusätzliche Zertifikatsinformationen) aktuell gerade noch in einer TLV möglich. Bei weiter wachsenden Schlüsseln wird dies aber nicht mehr gegeben sein. Der Versand von Zertifikaten müsste dann also ebenfalls auf ein externes Protokoll ausgelagert werden.

Um den Versand der Schlüssel dennoch innerhalb einer TLV realisieren zu können, stellt die Verwendung von Elliptische-Kurven-Kryptografie vorübergehend eine Lösung dar: als Äquivalent für 2048 Bit lange RSA-Schlüssel genügen laut NIST bereits 233 Bit lange Elliptische-Kurven-Schlüssel. [BW06] Langfristig werden allerdings auch hier größere Schlüssellängen erforderlich sein, um ein gleichbleibendes Sicherheitsniveau sicherzustellen.

Eine weitere Möglichkeit ist der regelmäßige Bezug einer Auswahl von öffentlichen Schlüsseln, welche der Client bereithält und dann für die Verifizierung durchprobiert. Dies ist im aktuellen Funktionsumfang allerdings nicht enthalten, nur bedingt skalierbar und stellt einen unnötigen Zusatzaufwand dar.

Sämtliche hier beschriebenen Lösungen weisen Schwächen auf. Um den regelmäßigen und flächendeckenden Einsatz auch in großen Netzen zu ermöglichen, ist daher die Integration einer PKI-Lösung sinnvoll. Beispielsweise könnte eine Lösung dabei folgendermaßen aussehen:

Aufseiten des Protokolls könnten Möglichkeiten für die Auswahl und eventuell für die Verteilung von öffentlichen Schlüsseln integriert werden. Dafür könnte das LAN-Beacon Protokoll dahingehend erweitert werden, dass die Quelle für Zertifikate bestimmt und diese dann von Clients heruntergeladen werden können. Zum Beispiel könnte der Hash des öffentlichen Schlüssels mitgesendet werden, anhand dessen der Client aus einer Sammlung von Schlüsseln den richtigen auswählt. Sofern der nötige Schlüssel nicht verfügbar ist, könnte in einem separaten Protokoll der Versand der Zertifikate oder Schlüssel erfolgen, die zuvor bei der spezifizierten Quelle angefragt werden.

6 Zusammenfassung und Ausblick

Die vorliegende Arbeit hat eine Lösung für die authentifizierte Selbstbeschreibung von Netzen dargestellt. Diese erfüllt neben dem in Kapitel 1 beschrieben Anwendungsfall der Identifikation von auf einem Anschluss transportierten VLANs auch weitere Anwendungsfälle.

Diese Anforderungen und Rahmenbedingungen für eine erfolgreiche Lösung der Problemstellung wurden in Kapitel 2 untersucht. Dabei wurde neben den Anforderungen der Nutzer auch auf die Bedürfnisse von Betreibern und Entwicklern eingegangen.

Die Analyse bestehender Lösungen in Kapitel 3 hat gezeigt, dass bisher bestehende Lösungen nur Teilaspekte lösen. Der schwerwiegendste Mangel war hierbei, dass kein Protokoll einfach, zuverlässig und mit Standardprogrammen nutzbar war. Zudem wurde auch in keinem Fall eine Möglichkeit zur Authentifizierung angeboten .

Daher fiel die Entscheidung auf eine Eigenentwicklung, die durch Erweiterung und Modifikation des Protokolls LLDP die verbleibenden Anforderungen erfüllen sollte. Mit dem in Kapitel 4 systematisch entwickelten Protokoll ist nicht nur die Identifikation von auf einem Anschluss transportierten VLANs möglich, daneben wurden auch weitere TLVs definiert, um zusätzliche Typen von Informationen zu transportieren. Zusätzlich wurde eine Möglichkeit zur Authentifizierung auf Basis von Signaturen und mit Abwehrmaßnahmen gegen Angriffe durch Wiedereinspielung dargestellt.

Die Anwendbarkeit des Protokolls wurde anhand des in Kapitel 5 beschriebenen Prototypen auf einem Kleinrechner evaluiert. Hierbei hat sich gezeigt, dass die Einsatzmöglichkeit in dem definierten Anwendungsfall gegeben ist und eine leicht und intuitiv zu bedienende Lösung entwickelt wurde. Bei der Entwicklung wurde auf einen modularen Aufbau, Portabilität auf andere Betriebsumgebungen und einfache Internationalisierung geachtet. Die Installation auf neuen Geräten wurde weitmöglichst vereinfacht und die Anwendung dokumentiert.

6.1 Zentrale Neuerungen und weitere Forschung

Die Ergebnisse der Bachelorarbeit sind das entwickelte Protokoll und der implementierte Prototyp. Beide wurden selbst entwickelt, da keine existierende Lösung für die spezifischen Anwendungsfälle gefunden werden konnte. Hierbei bleiben allerdings Fragen offen, zum Beispiel nach dem Grad der Sicherheit in dem aktuellen Protokollentwurf und nach möglichen Angriffen. Zudem könnten weitere Funktionen oder verbesserte Bedienungsmöglichkeiten in den Prototyp integriert werden. Diese Fragestellungen stellen Anknüpfungspunkte für weitere Forschung dar.

Die zentrale Neuerung ist die einfache Identifikation von virtuellen Netzen und die darauf aufbauende Entwicklung der kompakten Lösung zur Identifikation der Netze auf Basis eines Kleinrechners. Dieser konnte durch die Verwendung von Standardhardware für weniger als 80 Euro zusammengebaut werden. Somit wird die schnelle Analyse der Netztopologie ermöglicht, ohne hierfür spezielle Software oder einen kompletten Rechner transportieren zu

müssen. Die zusätzliche Auswertung des LLDP-Standard entsprechenden Verkehrs ist mit geringfügigen Erweiterungen möglich. Hiermit könnte der entwickelte Prototyp noch weitere Informationen als die in Abschnitt 2.2.2 beschriebenen darstellen. Dies würde die Analyse der Netztopologie noch weiter erleichtern.

Da die verwendete Hardware noch weit von ihren Kapazitätsgrenzen entfernt ist, wäre eine Portierung auf noch günstigere, sparsamere und kompaktere Plattformen möglich. Beim Empfang von vielen verschiedenen oder umfangreichen LAN-Beacon- und eventuell LLDP-Rahmen ist eine übersichtliche Darstellung aufgrund der fehlenden Interaktionsmöglichkeiten schwierig. Da Bildschirme mit Touchscreen-Funktion aber mittlerweile für knapp über 20 Euro verfügbar sind, könnte ein solcher eingebaut und das Programm um die Eingabemöglichkeit erweitert werden. Hiermit könnte durch die Informationen geblättert oder zusätzlich integrierte Analysefunktionen bedient werden.

Die weitreichendste Änderung des Protokolls stellt die in Kapitel 4.5 beschriebene Anpassung von Zieladresse und EtherType dar. Diese war nötig, um den Versand der Rahmen innerhalb einer gesamten Broadcast-Domäne zu ermöglichen, der dem Standard entsprechend nicht vorgesehen war. Daher stellt sich die Frage, ob andere Lösungen mit weniger einschneidenden Änderungen gefunden werden können, bei welchen die Kompatibilität mit LLDP erhalten bleibt. Diese könnten dann dort oder in andere Protokolle integriert werden, um zumindest einen Teil der in Kapitel 1 und 2 beschriebenen Anwendungsfälle und Anforderungen mit Standardprotokollen abzudecken.

Ein Beispiel hierfür ist die Möglichkeit zur Authentifizierung inklusive Maßnahmen gegen einen Angriff durch Wiedereinspielung. Die beschriebene Lösung könnte sehr einfach in LLDP integriert werden, während die Kompatibilität zu älteren Implementierungen erhalten bleibt. Dies hätte vor allem in physisch öffentlich zugänglichen Netzen wie an einer Universität den Vorteil, dass der Herkunft der empfangenen Informationen zumindest mit einem grundlegenden Sicherheitsniveau vertraut werden kann.

Um dieses Sicherheitsniveau zu erhöhen, ist es notwendig und sinnvoll, zusätzlich zu dem Abschnitt 5.7.2 mögliche Angriffsszenarien und Gegenmaßnahmen herauszuarbeiten. Da auch die zugrundeliegenden Verfahren für die Signierung möglicherweise bisher unbekannte Sicherheitslücken aufweisen, ist zudem eine regelmäßige Überprüfung des Sicherheitsniveaus notwendig.

6.2 Durch zukünftige Erweiterungen mögliche Anwendungsfälle

Auf Basis der beschriebenen Entwicklung sind weitere Funktionen und Anwendungsfälle denkbar. Zum Teil sind diese anhand des aktuellen Entwicklungsstandes umsetzbar, andere bieten sich als Anknüpfungspunkte für zukünftige Erweiterungen an. Für beide der hier beschriebenen Anwendungsfälle ist die Authentifizierung der empfangenen Daten sinnvoll.

Das Protokoll könnte in Verbindung mit Software-defined Networking (SDN) nützlich sein, bei welchem die Konfiguration des Verhaltens der Weiterleitungsebene weitgehend automatisiert und zentral erfolgt. Hierbei liegt der Fokus auf der Möglichkeit zur schnellen Anpassung der Konfiguration von Netzen. Da Protokolle wie OpenFlow [DPM+10] ebenfalls nicht auf Menschenlesbarkeit des Netzverkehrs ausgelegt sind, könnten Informationen über aktuelle Topologie und Weiterleitung des Verkehrs über das LAN-Beacon Protokoll bereitgestellt werden. Somit könnten dann an sämtlichen angeschlossenen Rechnern leicht zugängliche und aktuelle Informationen empfangen werden. Der Versand selbst könnte auf einem Rechner

oder durch die Software der Kontrollschicht übernommen werden.

Zudem könnte auch ein Teil der Konfiguration von angeschlossenen Geräten durch das Protokoll erfolgen. So könnte zum Beispiel die aktuelle Kontaktadresse der Systemadministratorin über das Protokoll versendet und für die automatische Fehlermeldung durch Rechner und Netzhardware verwendet werden. Ändert sich die Konfiguration zu einem späteren Zeitpunkt, könnten diese Änderungen den entsprechenden Geräten durch das Protokoll bekannt gemacht werden. Um die Art der sinnvoll zu transportierenden Informationen zu bestimmen sind weitere Untersuchungen notwendig, wobei die in Abschnitt 3.3 beschriebene Größenbeschränkung auf insgesamt 1500 Byte abzüglich Signatur ein limitierender Faktor ist.

Abbildungsverzeichnis

Literaturverzeichnis

[Ber] Bernat, Vincent: *lldpd » implementation of IEEE 802.1ab (LLDP).* https://vincentbernat.github.io/lldpd/index.html. (Accessed on 07/09/2017).

[Ber17] Bernat, Vincent: *lldpd: implementation of IEEE 802.1ab (LLDP).* https://github.com/vincentbernat/lldpd/blob/master/README.md, 05 2017. (Accessed on 07/09/2017).

[BW06] Blake-Wilson, et al.: *RFC 4492 - Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS).* https://tools.ietf.org/html/rfc4492, 5 2006. (Accessed on 06/28/2017).

[CS06] Cisco Systems, Inc.: *Inter-Switch Link and IEEE 802.1Q Frame Format.* http://www.cisco.com/c/en/us/support/docs/lan-switching/8021q/17056-741-4.html, 08 2006. (Accessed on 07/24/2017).

[DA97] Droms, Ralph und Steve Alexander: *RFC 2132 - DHCP Options and BOOTP Vendor Extensions.* RFC 2132, März 1997.

[DPM+10] Das, Saurav, Guru Parulkar, Nick McKeown, Preeti Singh, Daniel Getachew und Lyndon Ong: *Packet and circuit network convergence with OpenFlow.* In: *Optical Fiber Communication Conference*, Seite OTuG1. Optical Society of America, 2010.

[Eas11] Eastlake & Hansen: *RFC 6234 - US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF).* https://tools.ietf.org/html/rfc6234, 05 2011. (Accessed on 07/16/2017).

[FSF16a] Free Software Foundation, Inc.: *Autoconf.* http://www.gnu.org/software/autoconf/autoconf.html, 05 2016. (Accessed on 07/10/2017).

[FSF16b] Free Software Foundation, Inc.: *gettext.* https://www.gnu.org/software/gettext/, 07 2016. (Accessed on 07/10/2017).

[HD04] HICKEY, ANN M. und ALAN M. DAVIS: *A Unified Model of Requirements Elicitation.* Journal of Management Information Systems, 20(4):65–84, 2004.

[Hou04] Housley: *RFC 3874 - A 224-bit One-way Hash Function: SHA-224.* https://tools.ietf.org/html/rfc3874, 09 2004. (Accessed on 07/16/2017).

[HP06] Hewlett-Packard Company: *Alternative Features for Link Aggregation and Device Discovery (End of Support for FEC and CDP).* http://whp-hou9.cold.extweb.hp.com/pub/networking/software/LLDP-and-LACP-statement.pdf, 1 2006. (Accessed on 06/25/2017).

[HP15] Hewlett-Packard Development Company, L.P.: *LLDP*. http://h22208.www2.hpe.com/eginfolib/networking/docs/switches/WB/15-18/5998-8162_wb_2920_mcg/content/ch06s03.html, 2015. (Accessed on 07/08/2017).

[IEE09] IEEE Computer Society: *Station and Media Access Control Connectivity Discovery*. http://standards.ieee.org/getieee802/download/802.1AB-2009.pdf, 09 2009. (Accessed on 11/02/2016).

[IEE14a] IEEE Computer Society: *IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks*. https://standards.ieee.org/findstds/standard/802.1Q-2014.html, 11 2014. (Accessed on 06/28/2017).

[IEE14b] IEEE Computer Society: *IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture*. https://standards.ieee.org/findstds/standard/802-2014.html, 06 2014. (Accessed on 07/25/2017).

[Int96] International Organization for Standardization: *ISO/IEC 14977:1996 Information Technology - Syntactic Metalanguage - Extended BNF*, 1996.

[Int17] Internet Assigned Numbers Authority: *IEEE 802 Numbers*. https://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.xhtml, 05 2017. (Accessed on 07/07/2017).

[JN09] Jain, V. und J.R. Nogueras: *Automatic VLAN ID discovery for ethernet ports*, Februar 24 2009. US Patent 7,496,052.

[JPTH08] Johnson, T., M. Pothier, R. Turnbull und K. Huang: *Method and apparatus for VLAN ID discovery*, Juni 10 2008. US Patent 7,385,973.

[LL08] Lee, S.C. und P.C. Liu: *Method for preventing unauthorized connection in network system*, Juni 12 2008. US Patent App. 11/798,313.

[Mic10] Microsoft: *Link Layer Topology Discovery Protocol Specification*. https://msdn.microsoft.com/en-us/windows/hardware/gg463061.aspx, 09 2010. (Accessed on 11/03/2016).

[MSd14] *Link Layer Topology Discovery (Compact 2013)*. https://technet.microsoft.com/de-de/library/gg156955.aspx, 3 2014. (Accessed on 06/25/2017).

[OMG09] Object Management Group, Inc. (OMG): *Unified Modeling Language Specification (version 2.2)*. http://www.omg.org/spec/UML/2.2/Superstructure/PDF, 2 2009. (Accessed on 07/23/2017).

[SFR03] Stevens, W. Richard, Bill Fenner und Andrew M. Rudoff: *UNIX Network Programming, Vol. 1*. Pearson Education, 3 Auflage, 2003.

[Ste06] Stevens, Hibbs &: *Implementation Issues with RFC 2131, "Dynamic Host Configuration Protocol (DHCPv4)"*. https://tools.ietf.org/html/draft-ietf-dhc-implementation-02, 06 2006. (Accessed on 07/17/2017).

[vH16] Heesch, Dimitri van: *Doxygen*. http://www.stack.nl/~dimitri/doxygen/, 12 2016. (Accessed on 07/10/2017).

[Wir] Wireshark: *CaptureSetup/VLAN*. https://wiki.wireshark.org/CaptureSetup/VLAN. (Accessed on 11/03/2016).

[YB05] Yusuf Bhaiji, Cisco Systems: *Layer 2 attacks and mitigation techniques*. http://www.sanog.org/resources/sanog7/yusuf-L2-attack-mitigation.pdf, 2005. (Accessed on 11/03/2016).