DIPLOMARBEIT

Optimierung von Roaming in IEEE 802.11 Wireless Netzwerken

ausgeführt zur Erlangung des akademischen Grades
eines Diplom-Ingenieurs unter der Leitung von

ao. Univ. Prof. Dipl.-Ing. Dr.techn. Thilo Sauter

am

Institut für Computertechnik (E384)
der Technischen Universität Wien

durch

Stefan Feirer, BSc
Matr.Nr. 0926020
Carminweg 6/9/18, 1210 Wien

Wien, am 05.11.2016 _____

Kurzfassung

Die Anzahl an mobilen Geräten wie Laptops, Smartphones und Tablets nimmt immer mehr zu. Neben dem Einsatz im privaten Bereich steigt auch die Nutzung solcher Geräte in der Arbeitswelt und im industriellen Umfeld. Für eine ununterbrochene Datenverbindung sorgen in einer lokalen Umgebung zumeist WLAN (Wireless Local Area Network) und Standards der Familie IEEE 802.11. Reicht im privaten Umfeld meist ein einzelner Access Point um für eine ausreichende Funkverbindung zu sorgen, so ist im Bürobetrieb oder in der industriellen Automation eine ganze Infrastruktur an Zugangspunkten nötig. Dies bietet den Vorteil, dass immer ein bestmöglicher Empfang gewährleistet werden kann, auch wenn sich ein Client in Bewegung befindet. Problematisch bei diesem Szenario ist jedoch, dass beim Verlassen der Reichweite eines Access Points die Verbindung zu einem anderen Access Point umgeschaltet werden muss. Dieser Vorgang wird als Roaming bezeichnet. Von einem solchen Roaming-Vorgang wird grundlegend gefordert, dass dieser möglichst kurz dauert, um den Einfluss auf eine laufende Datenverbindung so gering wie möglich zu halten. Darüber hinaus muss ein Roaming transparent für höhere Schichten ablaufen. Das heißt am Client laufende Applikationen dürfen vom Wechsel zu einem anderen Access Point keine Notiz nehmen. Momentan wird bei der Verbindungsumschaltung meist kein richtiges Roaming durchgeführt. Vielmehr passiert ein Abbruch gefolgt von einem Neuaufbau der Verbindung. Dies führt unweigerlich dazu, dass laufende Applikationen ebenfalls unterbrochen werden.
Die Aufgabe dieser Arbeit besteht darin, zu untersuchen, ob und wie ein solcher Roaming-Prozess unter der Verwendung des Standards IEEE 802.11 realisiert werden kann. Dazu wurde ein Algorithmus entwickelt, der ein effizientes aber vor allem transparentes Roaming im Bezug auf Applikationen ermöglicht. Zum Test der Funktion und Leistung des Verfahrens kam ein eigens entworfener Aufbau, bestehend aus zwei Access Points und einem Client zum Einsatz. Es konnte erreicht werden, dass der entworfene Roaming-Algorithmus unter der Verwendung der Standard-Erweiterungen IEEE 802.11k und IEEE 802.11r proaktive und transparente Roaming-Abläufe mit einer Dauer im Bereich von 100 ms ermöglicht.

Abstract

The number of mobile devices like laptops, smartphones and tablets increases steadily. Not only on the consumer market but in office and industrial applications more and more mobile devices are used. For a continuous data connection in local areas Wi-Fi and the standards of the IEEE 802.11 familiy are widely-used. In private areas usally a single access point (AP) provides a radio signal sufficiently that all mobile devices can be handled. In office or industrial areas a complex infrastructur consisting of many APs is needed to cover the whole area with Wi-Fi. The advantage of this setup is that in the whole area an adequate signal can be received even if a client is moving. But problems can occur in this scenario if a client leaves the range of one AP. The existing connection has to be switched to another AP. This process is called roaming. The basic requirement of a roaming procedure is that the duration of the switch has to be as short as possible to guarantee that ongoing data transmissions are not influenced. An other requirement is that roaming should happen transparently to upper layers. This means running applications must not recognize that the connection has changed to an other AP. In current available solutions there are no real roaming procedures but a loss of the old and an establishment of a new connection. This results in abruption of running applications.
The focus of this work is to identify if the required roaming process is possible in IEEE 802.11 networks. Therefore an algorithm was developed which supports efficient roaming regarded to applications. For testing purposes an setup consisting of two APs and one client was built. Measurements proofed that under the use of the developed algorithm combined with the IEEE 802.11 amandments IEEE 802.11k and IEEE 802.11r proactive and seamless roaming procedures without the nessecity of scanning radio frequencies are possible. The overall performance of handovers was improved significantly and handovers with a duration of approximately 100 ms were achieved.

Danksagung

Es ist Zeit, mich an dieser Stelle bei meiner Familie und meinen Freunden für die Begleitung, Unterstützung und manchmal auch Aufmunterung während des Schreibens dieser Arbeit und über das gesamte Studium hinweg, zu bedanken. Besonderer Dank gilt meinen Eltern, die mir dieses Studium ermöglichten und immer ein offenes Ohr für mich hatten. Ebenfalls gebührt meiner Freundin Angelika unendlich großer Dank, dass sie mir in einer anstrengenden Zeit immer den Rücken gestärkt und meine Launen abgefangen hat. Darüber hinaus möchte ich mich bei Herrn Prof. Sauter für Betreuung der Diplomarbeit und bei allen Kollegen der Mission Embedded GmbH, die mich in fachlichen Fragen unterstützten, bedanken.

Inhaltsverzeichnis

Abkürzungen

AES	Advanced Encryption Standard
AP	Access Point
BLOB	Binary Large Object
BSS	Basic Service Set
BSSID	Basic Service Set Identifier
CCKM	Cisco Centralized Key Management
CCMP	Counter Mode with Cipher Block Chaining Message Authentication Code Protocol
EAP	Extensible Authentication Protocol
EAPOL	Extensible Authentication Protocol over Local Area Network
ESS	Extended Service Set
FTIE	Fast Transition Information Element
GSM	Global System for Mobile Communications
IAPP	Inter Access Point Protocol
IETF	Internet Engineering Task Force
LAN	Local Area Network
MAC	Medium Access Control
MDIE	Mobility Domain Information Element
MIB	Management Interface Base
MLME	MAC Sublayer Management Entity
MSK	Master Session Key
MIC	Message Integrity Check
OKC	Opportunistic Key Caching
PKC	Proactive Key Caching
PMK	Pairwise Master Key
PMKID	Pairwise Master Key Identifier
PTK	Pairwise Transient Key
RADIUS	Remote Authentication Dial-In User Service
RCPI	Received Channel Power Indicator
RSSI	Received Signal Strength Index
RSN	Robust Security Network
SCTP	Stream Control Transmission Protocol
SME	Station Management Entity
SSID	Service Set Identifier
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
WDS	Wireless Distribution System
WEP	Wired Equivalent Privacy

WEXT	Wireless Extension
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access

1 Einleitung

Mobilität, ständige Erreich- und Verfügbarkeit sind in der heutigen Arbeitswelt eine Grundvoraussetzung um konkurrenzfähig zu sein. Eine ununterbrochene Verbindung zu verschiedenen Netzwerken ist vor allem im Dienstleistungs- und Industriebetrieb wichtig. So besagt eine Studie [6] der Statistik Austria aus dem Jahr 2014, dass bereits 19.2 % aller Beschäftigten von ihrem Arbeitgeber mit einem mobilen Endgerät ausgestattet sind. Die zunehmende Daten- und Applikationshaltung auf Firmenservern oder auf externen Plattformen macht eine durchgängige Netzwerkverbindung unumgänglich, um die Arbeit im Bürobetrieb verrichten zu können. Auch in automatisierten Industrieanlagen, wie beispielsweise in Hochregallagern, werden vermehrt die Vorteile von drahtlosen Anbindungen ausgenutzt. Aufgrund vielfältiger Anwendungsmöglichkeit halten also drahtlose Technologien Einzug in eine Vielzahl von Lebensbereichen.

Im Bereich von lokalen Netzwerken ist der WLAN Standard IEEE 802.11 weltweit verbreitet. Im Jahr 2016 wurden sogar erstmals mehr Daten über WLAN-Verbindungen zu mobilen Geräten übertragen als über zellulare Technologien [Sys16]. Kaum ein mobiles Endgerät, wie Laptop, Smartphone oder Tablet wird heute ohne diesem Standard auf den Markt gebracht. Sind im privaten Gebrauch von WLAN vor allem ein hoher Datendurchsatz und Sicherheit von Bedeutung, so werden im geschäftlichen Umfeld erweiterte Anforderungen an die Netzwerkinfrastruktur gestellt.

Eine durchgängige und zuverlässige Datenverbindung, auch wenn sich ein Client in Bewegung befindet, ist ein wesentlicher Punkt. Gerade im Bürobetrieb, wo Mitarbeiter mit Mobilgeräten wie Smartphones oder Laptops laufend ihren Standort wechseln, darf die Verbindung zum Netzwerk nicht unterbrochen werden, da ein Verbindungsabbruch Auswirkung auf eine Vielzahl von Applikationen haben kann. Ebenfalls kritisch sind Systeme, in denen Roboter eingesetzt werden, die sich autonom bewegen können. Man denke an industrielle und automatisierte Fertigungsanlagen oder roboterunterstütze Lagerhaltung und Warendistribution um nur einige Anwendungsgebiete zu nennen. Entscheidend bei den zuvor genannten Anwendungen ist, dass sich mobile Geräte in einer Infrastruktur, bestehend aus einer Vielzahl von Access Points (APs) bewegen und auch während der Bewegung eine durchgängige Verbindung zum Netzwerk bestehen muss, um einen reibungslosen, effizienten und sicheren Betrieb gewährleisten zu können.

1.1 Motivation

Ein besonders kritischer Task beim Betrieb eines WLAN-Netzwerkes ist das Roaming zwischen APs, also das Umschalten der Netzwerkverbindung von einem zu einem anderen AP, wenn sich

ein Client in Bewegung befindet. Neben der zeitlich kritischen Anforderung dürfen natürlich Sicherheitsaspekte nicht außer Acht gelassen werden. Somit muss die drahtlose Verbindung neben Performance auch einen starken Schutz gegen unerlaubte Zugriffe bieten. Bisher musste ein Kompromiss zwischen einem hochperformanten und einem hoch sicheren Netzwerk eingegangen werden, da starke Sicherheitsmethoden viel Rechenzeit benötigten und in deren Folge erhebliche Verzögerungen verursachten. So kann beispielsweise mit einem ungesicherten oder mit Wi-Fi Protected Access 2-Personal (WPA2 - Personal) gesicherten WLAN eine gute Roaming Performance erreicht werden, jedoch ist ein Einsatz in einer professionellen Umgebung aufgrund fehlender Sicherheit, aber auch Verwaltbarkeit, nicht möglich. Andererseits kann ein durch WPA2-Enterprise gesichertes Netzwerk die Gefahr von Angriffen drastisch verringern. Allerdings ist der 802.1X Authentifizierungsprozess sehr umfangreich, was wiederum eine schlechte Roaming Performance mit sich bringt.

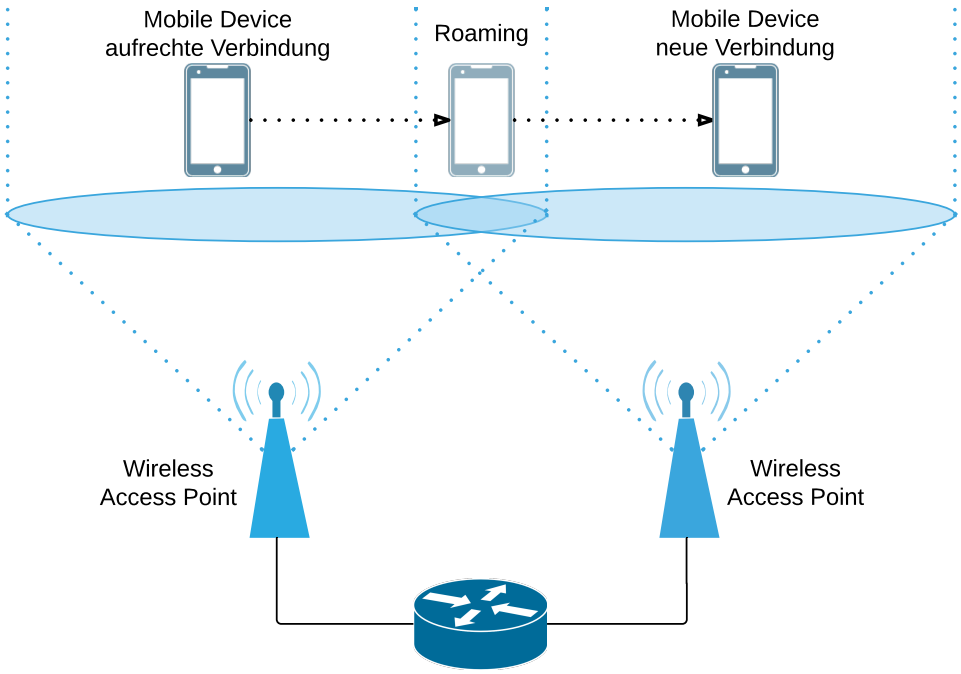

Abbildung 1.1: Szenario einer Verbindungsumschaltung von einem zu einem anderen AP

Einleitend soll nun näher auf den Roaming-Prozess eingegangen werden. Wie zuvor erwähnt, handelt es sich bei Roaming um jenen Prozess, bei dem ein Client die WLAN-Verbindung von einem AP innerhalb seiner Reichweite zu einem anderen AP in seiner Reichweite wechselt, ohne dass dabei die Netzwerkverbindung unterbrochen wird (siehe Abb. 1.1). Voraussetzung für Roaming in einem IEEE 802.11 Netzwerk ist, dass beide APs dem gleichen Extended Service Set (ESS) angehören. Ein ESS entsteht dann, wenn mehrere APs den selben Service Set Identifier (SSID) verwenden um ein Netzwerk zu erzeugen. Eine SSID ist wiederum die Kennung eines WLANs. Anders als zum Beispiel bei klassischen Mobilfunkverfahren wie Universal Mobile Telecommunications System (UMTS) oder Global System for Mobile Communications (GSM) wird der Handover-Prozess nicht von der Basisstation sondern vom Client gesteuert. Das bedeutet, dass die Roaming-Entscheidung letztendlich immer beim Client und nie beim AP liegt. Der AP kann den Client lediglich in seiner Roaming-Entscheidung beeinflussen und unterstützen, indem

er ihm Informationen über mögliche Roaming-Stationen übermittelt.

Ebenfalls ist im Standard IEEE 802.11 festgelegt, dass ein Client zu jedem Zeitpunkt nur eine aufrechte Verbindung zum Netzwerk haben darf [Soc12]. Grund dafür ist, dass ein AP Daten aus dem Netzwerk an den Client weiterleiten muss. Die Daten können jedoch nur dann den Weg zum Client finden, wenn es nur eine aufrechte Verbindung, also nur einen Pfad zum Client gibt. Dies bedeutet wiederum, dass beim herkömmlichen Roaming der bestehende Datenpfad unterbrochen werden muss, um einen Wechselprozess initiieren zu können.

Eine weitere Anforderung an den Handover ist, dass dieser für höhere Schichten transparent ablaufen muss. Das heißt, dass auch Verbindungen, die in höheren Schichten, beispielsweise TCP-Verbindungen oder Applikationen aufgebaut wurden, nicht unterbrochen oder gar abgebrochen werden.

In [BK12] wird Roaming in drei Kategorien, je nach Anforderung an den Handover, eingeteilt. **Fast Handover** fordert eine Minimierung der Dauer der Verbindungsunterbrechung. Der Paketverlust, welcher durch die Unterbrechung entsteht, ist dabei nicht von Bedeutung. **Smooth Handover** hingegen befasst sich mit der Minimierung des Paketverlusts während der Verbindungsunterbrechung, wobei die Dauer der Verbindungsunterbrechung nicht im Fokus ist. **Seamless Handover** fordert eine Umschaltung ohne Qualitätsabnahme der Verbindung, also eine Kombination aus den beiden Kategorien von Fast und Smooth Handover.

Die grundlegende Abfolge eines Roaming-Prozesses ist der Abbildung 1.2 zu entnehmen. Der Ablauf wird in die drei Phasen Entdeckungsphase, Authentifizierungsphase und Assoziierungsphase gegliedert [PCKC07b]. In der Entdeckungsphase erfolgt die Suche nach einem AP. Dazu müssen die verschiedenen Funkkanäle gescannt werden. T_{Scan} beschreibt dabei die Dauer der Entdeckungsphase. In der Authentifizierungsphase, angegeben mit T_{Auth}, werden je nach verwendeter Methode diverse Nachrichten zwischen Client und AP ausgetauscht um zu prüfen, ob der Client berechtigt ist, Zugang zum Netzwerk zu erhalten. In der Assoziierungsphase registriert sich ein Client bei einem AP um vollen Netzwerkzugang zu erhalten. Dieser abschließende Prozess im Verbindungsaufbau ist wichtig, damit der AP in weiterer Folge Datenpakete richtig zustellen kann. Dabei ist zu beachten, dass ein Client immer nur mit einem einzigen AP verbunden sein kann. Die Dauer dieser Phase wird in Abbildung 1.2 mit T_{Asso} angegeben. Eine genauere Analyse dieser drei Phasen erfolgt im Abschnitt 2.

Es sei aber an dieser Stelle angemerkt, dass der Großteil der Gesamtverzögerung im Roaming-Prozess auf die Entdeckungs- und Authentifizierungsphase zurückzuführen ist. Sowohl die Suche nach einem Roaming-Ziel als auch die Authentifizierung können, je nach eingesetzter Methode, eine nahtlose Datenübertragung verhindern.

1.2 Aufgabenstellung und Methodik

Die Aufgabe dieser Arbeit besteht darin zu analysieren, wie Roaming in einem WLAN-Netzwerk zeitlich beschleunigt werden kann. Dazu soll im ersten Schritt eine genaue Analyse der zeitlichen Abfolge von Verbindungsaufbauten in einem Netzwerk, das den Standard IEEE 802.11 verwendet, erfolgen.

Das schlussendliche Ziel dabei ist eine Verbindungsumschaltung in unter 100 ms um den Betrieb von Applikationen wie Audio- und Videotelefonie und zeitkritische Datenübertragungen in industriellen Anlagen gewährleisten zu können. Die zeitliche Anforderung von 100 ms ergibt sich

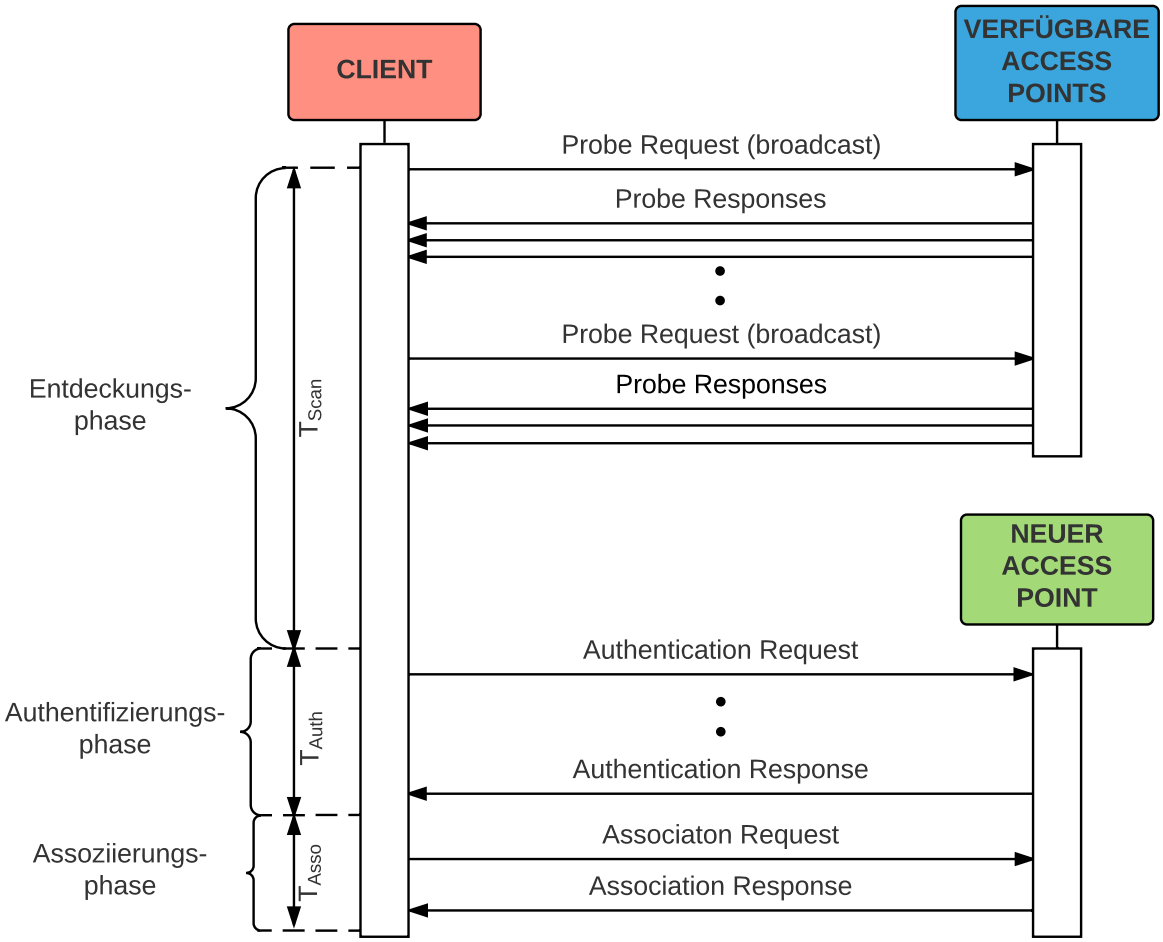

Abbildung 1.2: Ablauf des Verbindungsaufbaus zu einem IEEE 802.11 Wireless LAN Netzwerk [BK12]

aus der Vorgabe der ITU-T, dass die Gesamtverzögerung des Audiosignals in IP-Netzen zwischen zwei Gesprächspartnern maximal 150 ms sein darf [IT03]. Da bei dieser Vorgabe mit Gesamtverzögerung jedoch die Zeit zwischen Sprecher und Hörer, also die komplette Übertragungskette definiert ist und der Roaming-Vorgang nur einen Teil dieser Kette ist, wurde die zeitliche Anforderung dementsprechend geringer gewählt.

Grundlegend sollen bestehende Lösungen für Roaming in WLAN-Netzwerken verglichen werden. Des Weiteren wird eine Ausarbeitung stattfinden, ob beziehungsweise welche der Technologien in der Lage ist die definierten Anforderungen zu erfüllen. Dazu wird ein Testsystem zum Vergleich verschiedener Roaming-Varianten aufgebaut. Der Aufbau ist so zu realisieren, dass das Roaming-Verhalten in unterschiedlichen Varianten gemessen werden kann.

In Kapitel 2 erfolgt eine detaillierte Ausarbeitung, wie sich die Dauer eines Roaming-Vorgangs auf die verschiedenen Phasen aufteilt. Darüber hinaus werden verschiedene Möglichkeiten betrachtet, wie die Dauer bei der Verbindungsumschaltung verkürzt werden kann. Ebenfalls findet eine Untersuchung von derzeit noch ungelöste Probleme beim Roaming statt. Neben zwei IEEE 802.11 Erweiterungen werden auch proprietäre Lösungen und bestehende Ansätze zu einer verbesserten Roaming-Performance gegenübergestellt. Eine Gliederung der untersuchten Technologien erfolgt nach der entsprechenden Phase im Roaming-Prozess, in der eine Lösung eine zeitliche Beschleu-

nigung bringt. Dies kann entweder in der Entdeckungsphase oder in der Authentifizierungsphase sein.

In Kapitel 3 wird das Lösungskonzept zur Implementierung und Konfiguration der WLAN-Hardware beschrieben. Dazu wurde ein Roaming-Algorithmus entwickelt und implementiert. Dieser Prozess bringt unter der Verwendung der IEEE 802.11 Erweiterungen IEEE 802.11k und IEEE 802.11r eine verbesserte Roaming-Performance. Ebenfalls wird in diesem Kapitel die Umsetzung auf einem linuxbasierten Betriebssystem beschrieben. Dazu wird ein kurzer Überblick über WLAN in derartigen Architekturen gegeben. Darüber hinaus erfolgt die Beschreibung des Aufbaus des Testsystems zur Messung des Roaming-Verhaltens.

In Kapitel 4 findet die Aufbereitung der Messergebnisse statt. Es wird die Performance der entwickelten Roaming-Methode präsentiert. Darüber hinaus wird dieses Verfahren mit verschiedenen anderen Roaming-Varianten verglichen. In weiterer Folge findet auch eine Untersuchung der Vor- und Nachteile der vorgestellten Methode statt. Dazu erfolgt eine Analyse, welche Probleme die entwickelte Methode birgt.

Abschließend werden in Kapitel 5 Schlüsse aus den Messungen gezogen und diese interpretiert. Möglichkeiten zur Verbesserung und Weiterentwicklung des entwickelten Roaming-Algorithmus sind ebenfalls Teil dieses Kapitels.

2 Stand der Technik

In Kapitel 1.1 wurde festgelegt, einen Roaming-Prozess in die drei Phasen Entdeckungsphase, Authentifizierungsphase und Reassoziierungsphase zu unterteilen (siehe Abbildung 1.2). Einleitend folgt nun eine detaillierte Beschreibung dieser drei Phasen. Weiterführend werden im folgenden Kapitel verschiedene Möglichkeiten zur Verkürzung des Handoffs diskutiert.

In der Entdeckungsphase erfolgt die Suche, aber auch die Entscheidung, ob nach einem neuen AP gesucht werden muss. Die Notwendigkeit nach einem neuen AP zu suchen, kann verschiedene Gründe haben. Ein abrupter Verbindungsverlust, sinkende Verbindungsqualität oder ein AP, der höheren Datendurchsatz bietet. Diese und noch weitere Aspekte könnten der Entscheidung, nach einem Roaming-Ziel zu suchen, zugrunde liegen. Grundsätzlich zu unterscheiden ist jedoch, ob umgehend ein neuer AP gesucht werden muss, weil die bestehende Verbindung unterbrochen wurde, oder ob lediglich nach einem möglichen Roaming-Ziel Ausschau gehalten wird, um sich auf eine mögliche Verbindungsumschaltung vorzubereiten.

Die Festlegung anhand welcher Parameter ein Roaming durchgeführt wird, ist überdies keine leichte Aufgabe. Eine Möglichkeit ist der Received Signal Strength Index (RSSI) Wert. Dieser ist ein Indikator für die Empfangsfeldstärke an der Empfangsantenne. Dient der RSSI-Wert als Entscheidungsgrundlage, zu welchem AP eine Verbindung aufgebaut werden soll, so verbindet sich der Client mit dem AP mit dem höchsten RSSI-Wert. Diese Verbindung wird dann so lange gehalten, wie es die Empfangsqualität zulässt. Fällt diese unter einen Minimalwert, bricht die Verbindung ab [ZWB05]. Eine neue Verbindung findet schließlich wieder zum AP mit dem höchsten RSSI-Wert statt.

Die tatsächliche Suche nach möglichen Verbindungspunkten kann nach zwei unterschiedlichen Methoden passieren. Es kann sowohl passiv als auch aktiv nach einem neuen AP gesucht werden. Im passiven Modus wartet der Client auf die vom AP zyklisch ausgesandten Beacon Frames (Leuchtfeuer), in denen ein AP Informationen zu der von ihm angebotenen WLAN-Verbindung verbreitet. In der Standardkonfiguration werden Beacon Frames im Abstand von 100 ms übermittelt. Um somit alle 11 Kanäle (bei Verwendung von IEEE 802.11b und IEEE 802.11g) vollständig scannen zu können, ist eine Dauer von über 1 s notwendig. Wird der Standard IEEE 802.11a verwendet, müssten sogar 32 Kanäle gescannt werden [PCKC07a].

Im aktiven Modus sendet der Client selbst sogenannte Probe Requests aus und wartet auf Probe Responses von APs, welche sich in seiner Reichweite befinden und einen kompatiblen Modus anbieten. Dies läuft so ab, dass der Client auf einem bestimmten Kanal eine Anfrage schickt und gleichzeitig den sogenannten Probe Timer startet. Erreicht der Probe Timer die MinChannelTime, also die minimale Zeit die auf einem Kanal gesucht wird, so kann angenommen werden, dass

dieser Kanal von keinem AP genutzt wird. Es folgt ein Kanalwechsel. Falls der Client Aktivität auf dem Kanal feststellt, so wartet er maximal bis zur MaxChannelTime, also der maximalen Zeit auf einem Kanal, auf einen Probe Response. Im Probe Response gibt der AP sämtliche Eigenschaften, wie die unterstützten IEEE 802.11 Protokolle (z.B.: 802.11a, b, g, n), Authentifizierungsmöglichkeiten, Datenraten und die SSID bekannt. Die Suche nach einem neuen AP ist jedenfalls eine schwierige Angelegenheit und kann bis zu 90 % des gesamten Roaming-Prozesses in Anspruch nehmen [BK12].

Dazu ist in der Abbildung 2.1 die Aufzeichnung eines Verbindungsaufbaus inklusive dem Scan-Prozess zu einem ungesicherten Netzwerk ersichtlich. Es handelt sich dabei um eine mit Wireshark durchgeführte Messung. Dazu wurde ein Aufbau bestehend aus einem Client und einem AP verwendet. Die Aufzeichnung erfolgte direkt am AP mit einem dafür vorgesehenen Messinterface, welches auch zur Ferndiagnose hergezogen werden kann. Wie der Spalte *Time* zu entnehmen ist, vergehen zwischen dem ersten Probe Request vom Client (Nachricht Nr. 1), also dem Start des Scans bis zum Start der Authentifizierung (Nachricht Nr. 15, Authentication Request) 2.513 s. Der gesamte Aufbau der Verbindung, also bis zum Association Response (orange) dauert gesamt 2.599 s. Dies bedeutet, dass im Fall des Vebindungsaufbaus zu einem offenen Netzwerk fast die gesamte Dauer zum Finden eines APs notwendig ist. Die Authentifizierung und Assoziation zum Netzwerk dauert dann weitere 86 ms. In dieser Messung nimmt die Dauer des Scan-Vorgangs sogar 96.6 % des gesamten Verbindungsaufbaus in Anspruch.

No.	Time	Source	Destination	Info
1	0.000000	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=646, FN=0,
2	0.021780	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=647, FN=0,
3	0.064340	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=648, FN=0,
4	0.095102	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=649, FN=0,
5	0.135380	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=650, FN=0,
6	0.153994	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=651, FN=0,
7	0.173075	Aerohive_5a:66:da	IntelCor_c8:00:e4	Probe Response, SN=2811, FN=0
8	0.186277	Aerohive_5a:66:da	IntelCor_c8:00:e4	Probe Response, SN=2812, FN=0
9	0.195103	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=652, FN=0,
10	0.209901	Aerohive_5a:3e:da	IntelCor_c8:00:e4	Probe Response, SN=627, FN=0,
11	0.249628	Aerohive_5a:3e:da	IntelCor_c8:00:e4	Probe Response, SN=628, FN=0,
12	0.296573	Aerohive_5a:3e:da	IntelCor_c8:00:e4	Probe Response, SN=630, FN=0,
13	0.322927	Aerohive_5a:66:da	IntelCor_c8:00:e4	Probe Response, SN=2813, FN=0
14	0.384234	Aerohive_5a:3e:da	IntelCor_c8:00:e4	Probe Response, SN=631, FN=0,
15	2.513710	IntelCor_c8:00:e4	Aerohive_5a:66:da	Authentication, SN=665, FN=0,
16	2.517467	Aerohive_5a:66:da	IntelCor_c8:00:e4	Authentication, SN=256, FN=0,
17	2.597811	IntelCor_c8:00:e4	Aerohive_5a:66:da	Association Request, SN=666,
18	2.599881	Aerohive_5a:66:da	IntelCor_c8:00:e4	Association Response, SN=257,
19	2.609225	Aerohive_5a:66:da	IntelCor_c8:00:e4	Action, SN=258, FN=0, Flags=.

Abbildung 2.1: Aufzeichnung des Scan-Vorgangs und Verbindungsaufbau zu einem ungesicherten Netzwerk

In der Abbildung 2.2 ist die relative Häufigkeitsdichte einer Messreihe der Dauer des Verbindungsaufbaus zu einem ungesicherten Netzwerk zu sehen. Die gemessenen Zeiten enthalten sowohl die Dauer des Scans der Kanäle als auch des eigentlichen Verbindungsaufbaus zum Netzwerk. Die durchschnittliche Dauer betrug 2.57 s.

In der Authentifizierungsphase entscheidet ein AP, ob dem Client der Zugriff zum Netzwerk gewährt oder verweigert wird. Dazu wird ein Authentication Request vom Client an den AP

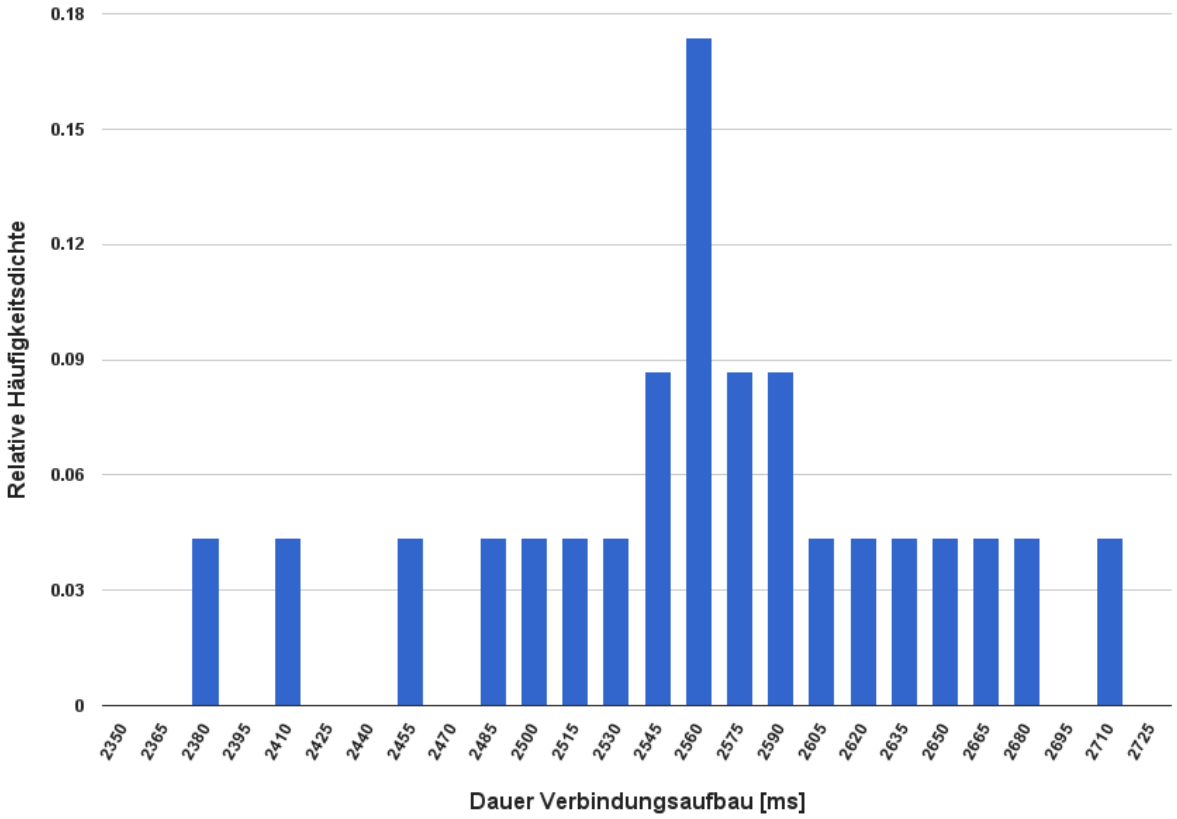

Abbildung 2.2: Relative Häufigkeitsdichte der Dauer eines Verbindungsaufbaus zu einem ungesicherten Netzwerk

gesendet. Der AP antwortet mit einem Response Frame in dem der Status der Authentifizierung übermittelt wird. Je nach Authentifizierungsmethode nimmt diese Phase unterschiedlich viel Zeit in Anspruch. Wird das Netzwerk ungesichert betrieben (siehe 2.1), so werden zwar die Authentifizierungsnachrichten geschickt, jedoch wird darin nur angegeben, dass keine Authentifizierung stattfindet.

In der Abbildung 2.1 dauert die Sendung der Authentifizierungsnachrichten im Falle des ungesicherten Netzwerkes ca. 4 ms. Werden hingegen starke Verschlüsselungs-, Authentifizierungs- und Key-Management Methoden eingesetzt, kann dies ebenfalls einige Zeit in Anspruch nehmen. So kann zum Beispiel die Verwendung des Standards IEEE 802.1X 500 ms bis 600 ms dauern [AH08]. Der Ablauf eines Verbindungsaufbaus nach dem Standard IEEE 802.1X mit einem Remote Authentication Dial-In User Service Server (RADIUS-Server) ist in Abbildung 2.11 dargestellt. Eine Messung des Verbindungsaufbaus unter Verwendung des Standards 802.1X und eines RADIUS-Servers ist der Grafik 2.3 zu entnehmen. Wie schon beim Verbindungsaufbau zu einem ungesicherten Netzwerk ist zu sehen, dass der Scan-Prozess (erster Probe Request bis zur Authentifizierung, Nachricht Nr. 1 - 17) 2.53 s in Anspruch nimmt. Der Authentifizierungsvorgang an sich dauert dann weitere 287 ms (gemessen von der ersten Authentifizierungsnachricht (Nachricht Nr. 17) bis zur letzten Key-Nachricht (Nachricht Nr. 47)). In Summe ergibt dies eine Dauer von 2.82 s für den gesamten Prozess.

No.	Time	Source	Destination	Info	Protocol
1	0.000000	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=243, …	802.11
2	0.016290	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=244, …	802.11
3	0.013928	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=244, …	802.11
4	0.039074	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=245, …	802.11
5	0.052505	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Probe Response, SN=2818…	802.11
6	0.053240	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Probe Response, SN=2819…	802.11
7	0.053642	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Probe Response, SN=2820…	802.11
8	0.068502	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=246, …	802.11
9	0.089477	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Probe Response, SN=1871…	802.11
10	0.144108	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=248, …	802.11
11	0.142249	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Probe Response, SN=2822…	802.11
12	0.164841	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Probe Response, SN=1873…	802.11
13	0.181794	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=249, …	802.11
14	0.201268	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=250, …	802.11
15	0.253971	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Probe Response, SN=1874…	802.11
16	0.257212	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Probe Response, SN=1876…	802.11
17	2.532110	IntelCor_c8:00:e4	Aerohive_5a:3e:d8	Authentication, SN=263,…	802.11
18	2.532630	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Authentication, SN=256,…	802.11
19	2.535651	IntelCor_c8:00:e4	Aerohive_5a:3e:d8	Authentication, SN=263,…	802.11
20	2.537859	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Authentication, SN=256,…	802.11
21	2.540399	IntelCor_c8:00:e4	Aerohive_5a:3e:d8	Association Request, SN…	802.11
22	2.541610	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Association Response, S…	802.11
23	2.547306	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Request, Identity	EAP
24	2.550773	IntelCor_c8:00:e4	Aerohive_5a:3e:d8	Response, Identity	EAP
25	2.557622	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Request, Protected EAP …	EAP
26	2.558182	IntelCor_c8:00:e4	Aerohive_5a:3e:d8	Client Hello	TLSv1
27	2.679212	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Server Hello, Certifica…	TLSv1
28	2.681830	IntelCor_c8:00:e4	Aerohive_5a:3e:d8	Response, Protected EAP…	EAP
29	2.694441	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Server Hello, Certifica…	TLSv1
30	2.697112	IntelCor_c8:00:e4	Aerohive_5a:3e:d8	Client Key Exchange, Ch…	TLSv1
31	2.744384	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Change Cipher Spec, Enc…	TLSv1
32	2.746598	IntelCor_c8:00:e4	Aerohive_5a:3e:d8	Response, Protected EAP…	EAP
33	2.755548	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Application Data	TLSv1
34	2.753749	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Application Data	TLSv1
35	2.754229	IntelCor_c8:00:e4	Aerohive_5a:3e:d8	Application Data, Appli…	TLSv1
36	2.765065	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Application Data	TLSv1
37	2.766252	IntelCor_c8:00:e4	Aerohive_5a:3e:d8	Application Data, Appli…	TLSv1
38	2.781114	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Application Data	TLSv1
39	2.781556	IntelCor_c8:00:e4	Aerohive_5a:3e:d8	Application Data, Appli…	TLSv1
40	2.793586	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Application Data	TLSv1
41	2.793809	IntelCor_c8:00:e4	Aerohive_5a:3e:d8	Application Data, Appli…	TLSv1
42	2.811822	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Success	EAP
43	2.813792	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Key (Message 1 of 4)	EAPOL
44	2.814591	IntelCor_c8:00:e4	Aerohive_5a:3e:d8	Key (Message 2 of 4)	EAPOL
45	2.817036	IntelCor_c8:00:e4	Aerohive_5a:3e:d8	Key (Message 2 of 4)	EAPOL
46	2.819504	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Key (Message 3 of 4)	EAPOL
47	2.819837	IntelCor_c8:00:e4	Aerohive_5a:3e:d8	Key (Message 4 of 4)	EAPOL

Abbildung 2.3: Aufzeichnung des Scan-Vorganges und Verbindungsaufbaus zu einem mit 802.1X gesicherten Netzwerk mit einem RADIUS-Server

Der Abbildung 2.4 ist die relative Häufigkeitsdichte einer Messreihe der Dauer des Verbindungsaufbaus zu einem mit dem Standard IEEE 802.1X gesichertern Netzwerk zu entnehmen. Die gemessenen Zeiten enthalten sowohl die Dauer des Scans der Kanäle, die Authentifizierung und den Verbindungsaufbau. Die durchschnittliche Dauer lag bei 2.84 s.

In der Reassoziierungsphase findet nach erfolgreicher Authentifizierung die Wiederherstellung der Verbindung statt. Diese wird durch einen Reassociation Request Frame vom Client an den neuen AP gestartet. Der AP wiederum bestätigt oder verweigert den Verbindungsaufbauversuch in einem Reassociation Response Frame. Wurde die Verbindung vom AP zugelassen, so kann der Client wieder Nutzdaten über die WLAN-Verbindung übertragen.

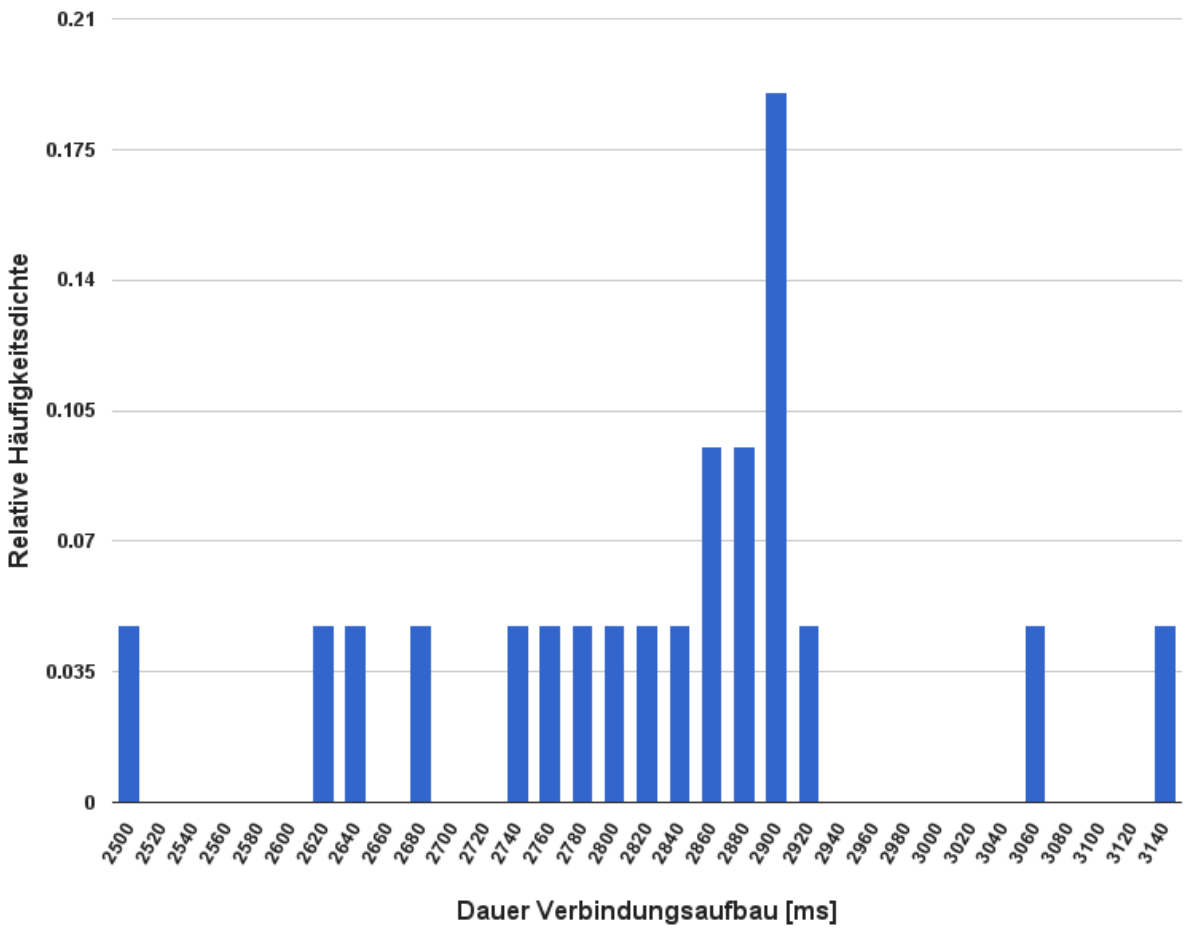

Abbildung 2.4: Relative Häufigkeitsdichte der Dauer eines Verbindungsaufbaus zu einem mit 802.1X gesicherten Netzwerk

Der Vollständigkeit halber soll an dieser Stelle auch noch der Verbindungsaufbau zu einem mit WPA2-Personal gesicherten Netzwerk angeführt werden (siehe Abbildung 2.5). Der Scan-Prozess dauert 2.58 s, vergleichbar mit den Varianten zuvor (Nachricht Nr. 1 - 20). Die Authentifizierungsphase fällt aufgrund des verwendeten Verfahrens WPA2-Personal mit 39 ms (Nachricht Nr. 21 - 30) wesentlich kürzer als jene bei Verwendung von IEEE 802.1X und WPA2-Enterprise aus. Dies liegt daran, dass keine Kommunikation zu einem RADIUS-Server notwendig ist, sondern die Ableitung und Einigung auf die verwendeten Schlüssel mit Hilfe eines 4-Way-Handshakes funktioniert.

Im Diagramm 2.6 ist die relative Häufigkeitsdichte einer Messreihe der Dauer des Verbindungsaufbaus zu einem mit dem Standard WPA2-PSK gesichertern Netzwerk ersichtlich. Neben der Dauer des Scans ist auch die Authentifizierung in den Messungen enthalten. Es wurde eine durchschnittliche Dauer von 2.57 s ermittelt.

Der vorangegangenen Beschreibung des Handover-Prozesses ist zu entnehmen, dass ein erheblicher Zeitaufwand in der Entdeckungs- und Authentifizierungsphase notwendig ist. Dies ist in der Entdeckungsphase die Entscheidung, wann ein Roaming-Prozess initiiert wird inklusive der Suche

No.	Time	Source	Destination	Info	Protocol
1	0.000000	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=141, …	802.11
2	0.014701	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=142, …	802.11
3	0.011814	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=142, …	802.11
4	0.024392	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Probe Response, SN=2756…	802.11
5	0.031282	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=143, …	802.11
6	0.039353	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Probe Response, SN=2757…	802.11
7	0.042788	Aerohive_5a:3e:d7	IntelCor_c8:00:e4	Probe Response, SN=1468…	802.11
8	0.044322	Aerohive_5a:3e:d7	IntelCor_c8:00:e4	Probe Response, SN=1470…	802.11
9	0.072715	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=144, …	802.11
10	0.115178	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=145, …	802.11
11	0.136299	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=146, …	802.11
12	0.137380	Aerohive_5a:3e:d7	IntelCor_c8:00:e4	Probe Response, SN=1471…	802.11
13	0.155229	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=147, …	802.11
14	0.160076	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Probe Response, SN=2759…	802.11
15	0.177795	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Probe Response, SN=2760…	802.11
16	0.180151	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Probe Response, SN=2762…	802.11
17	0.190663	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=148, …	802.11
18	0.228806	IntelCor_c8:00:e4	Broadcast	Probe Request, SN=149, …	802.11
19	0.253915	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Probe Response, SN=2763…	802.11
20	0.289741	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Probe Response, SN=2764…	802.11
21	2.580963	IntelCor_c8:00:e4	Aerohive_5a:66:d7	Authentication, SN=161,…	802.11
22	2.582549	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Authentication, SN=256,…	802.11
23	2.588177	IntelCor_c8:00:e4	Aerohive_5a:66:d7	Association Request, SN…	802.11
24	2.589249	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Key (Message 1 of 4)	EAPOL
25	2.589561	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Association Response, S…	802.11
26	2.605969	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Key (Message 1 of 4)	EAPOL
27	2.609409	IntelCor_c8:00:e4	Aerohive_5a:66:d7	Key (Message 2 of 4)	EAPOL
28	2.610568	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Key (Message 3 of 4)	EAPOL
29	2.617920	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Key (Message 3 of 4)	EAPOL
30	2.620186	IntelCor_c8:00:e4	Aerohive_5a:66:d7	Key (Message 4 of 4)	EAPOL
31	2.628563	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Action, SN=258, FN=0, F…	802.11

Abbildung 2.5: Aufzeichnung des Scan-Vorganges und Verbindungsaufbaus zu einem mit WPA2-Personal gesicherten Netzwerk

nach einer neuen Roaming-Station. In der Authentifizierungsphase kann es, je nach eingesetztem Verfahren, ebenfalls zu einer erheblichen Zeitverzögerung kommen. Daraus folgt, dass ein Handover nur dann schneller passieren kann, wenn die Dauer, sowohl der Entdeckungs- als auch der Authentifizierungsphase, verkürzt werden kann.

Dazu werden im folgenden Kapitel existierende Konzepte zur Beschleunigung des Roamings und deren Vor- und Nachteile vorgestellt und diskutiert. Eine Gliederung erfolgt, je nachdem an welcher Stelle ein angeführtes Konzept im Roaming-Vorgang einen Bonus für die Verbindungsumschaltung bringt. Im ersten Teil werden somit Technologien verglichen, die die Entdeckungsphase verkürzen und im zweiten Abschnitt Technologien, die die Dauer der Authentifizierungsphase reduzieren. Es sei an dieser Stelle angemerkt, dass eine umfassende zeitliche Beschleunigung des Roamings natürlich nur unter Verbesserung beider Phasen erfolgen kann.

2.1 Entdeckungsphase

Wie eingangs erwähnt wurde, ist die Entscheidung ob ein Roaming Prozess initiiert werden soll und die zuvor oder gleichzeitig ablaufende Suche nach möglichen Roaming-Zielen eine Aufgabe, die erhebliche Zeit in Anspruch nehmen kann. Dieser Ablauf passiert in der sogenannten

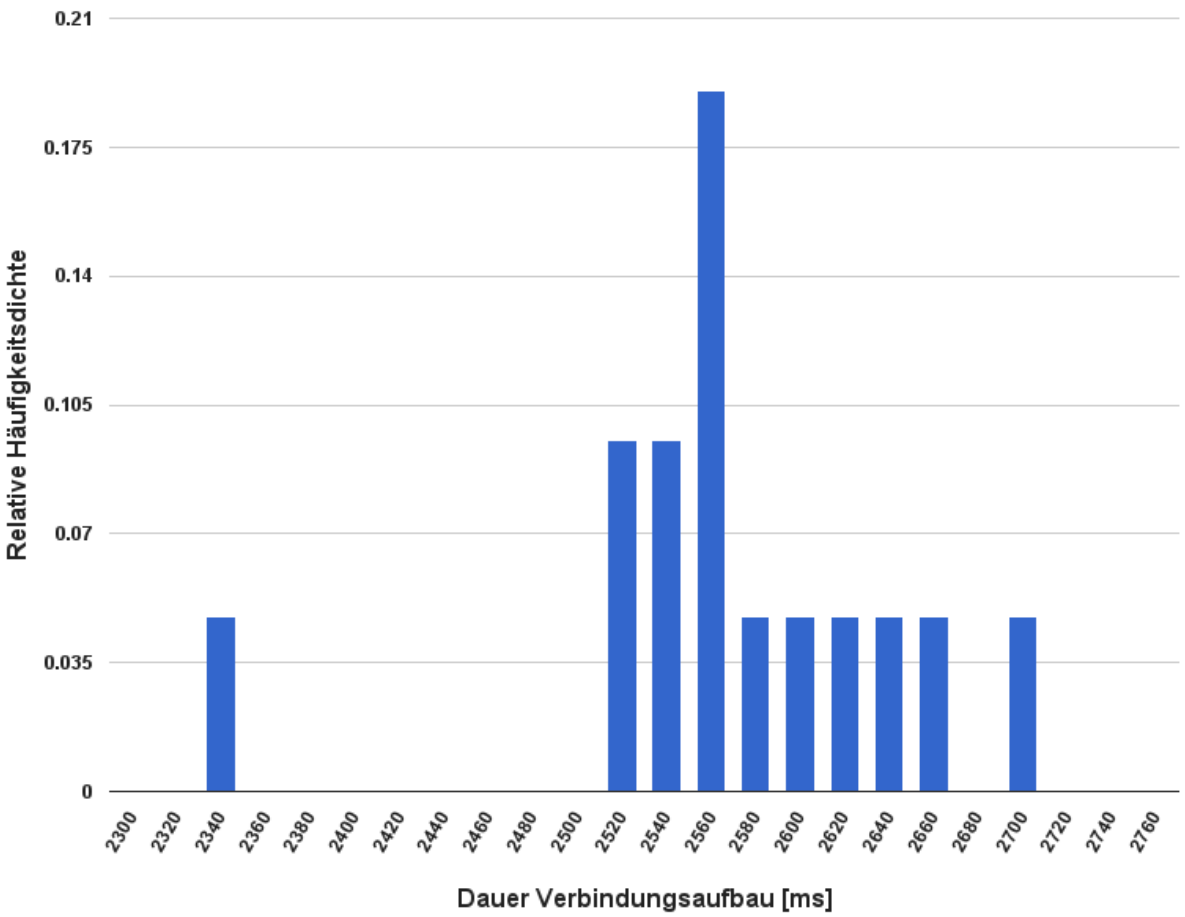

Abbildung 2.6: Relative Häufigkeitsdichte der Dauer eines Verbindungsaufbaus zu einem mit WPA2-PSK gesicherten Netzwerk

Entdeckungsphase (siehe Abbildung: 1.2). Der Client muss aufgrund der vorliegenden Daten entscheiden, ob ein Roaming durchgeführt wird. In weiterer Folge muss ein Roaming-Ziel, also ein AP, bestimmt werden, zu dem sich der Client verbinden kann.

Es gibt mehrere Konzepte, die sich mit der Verbesserung der Entdeckungsphase auseinandersetzen. Zugrunde liegt jedoch all diesen Methoden, dass eine Entscheidungsfindung nur durch Kenntnis des Netzwerks, im Speziellen der funktechnischen Umgebung des Clients, verbessert werden kann. Dies kann entweder passieren, in dem sich der Client selbst über seine Umgebung informiert, sprich auf anderen Funkkanälen nach benachbarten Stationen sucht, oder aber dass es eine zentrale Instanz gibt, welche diverse Informationen über die Netzwerkstruktur sammelt und bereit stellen kann. Zusammengefasst bedeutet das, dass für eine bessere Roaming-Entscheidung Informationen gesammelt werden müssen.

Des Weiteren müssen auch Überlegungen, wann die Suche nach einem neuen AP gestartet werden soll, getätigt werden. Im Standardfall wird dies durch das Unterschreiten eines Grenzwertes des RSSI ausgelöst. Jedoch sind auch hier andere Möglichkeiten, wie zum Beispiel die Beobachtung von Paketverlusten möglich [PCKC07a].

Grundsätzlich wäre auch eine Lösung mittels einer zentralen Steuerungsinstanz denkbar, wie sie zum Beispiel im Mobilfunk bei den Technologien UMTS und GSM zum Einsatz kommt. Dort entscheidet nicht der Client sondern ein zentraler Knoten über das Roaming. Dies hat den Vorteil, dass ein zentraler Steuerungsknoten aufgrund seiner Kenntnisse über das Netzwerk optimalere Entscheidungen treffen kann. Beispielsweise gibt es das Konzept des Personal AP [BK12], [ZWB05]. Es bietet eine zentrale Steuerung durch den sogenannten Access Controller. Dabei verwaltet der Access Controller für jeden verbundenen Client einen virtuellen Personal AP. Beim Roaming eines Clients von einem physikalischen AP zu einem anderen wird der virtuelle AP von AP zu AP weitergereicht. Das Konzept bedingt jedoch eine massive Änderung der Infrastruktur, wenn gleich beim Client keine Änderung notwendig ist. Das Konzept baut auf die 802.11 Erweiterung 802.11F auf, die als Trial-Use Recommended Practice, nicht aber als Standard spezifiziert wurde. Die Erweiterung 802.11F, Inter AP Protocol (IAPP), sollte die Interoperabilität von APs verschiedener Hersteller verbessern. Jedoch wurde diese 2006 aufgrund sicherheitskritischer Aspekte bei der Kommunikation zwischen APs zurückgezogen [DS14].

Ebenfalls wurde mit dem Gesamtkonzept FlexWare ein hybrides Netzwerk, bestehend aus kabelgebundener und kabelloser Technologien vorgestellt [SJB09]. Das System ist für den Einsatz im industriellen und automatisierten Umfeld gedacht. Die Architektur sieht dabei eine Gliederung in drei Schichten vor. Die unterste Schicht stellt die Anbindung der Endknoten, also Sensoren und Aktuatoren, dar. Diese werden mittels sogenannten FlexWare APs, die sich in der zweiten Schicht befinden, an das Netzwerk angebunden. In der dritten Schicht befindet sich mit dem FlexWare Controller der Koordinationspunkt für Echtzeitanforderungen, Roaming und Lokalisierung von Clients. Darüber hinaus stellt der Controller die Verbindung zum Backbone dar. Das Roaming wird dabei durch den Controller koordiniert. Dieser kann durch die Kenntnis der Standorte der Clients und der Reichweiten der APs nahtlose Roaming-Vorgänge bereitstellen.

Im Standard IEEE 802.11 ist festgelegt, dass die Roaming-Entscheidung immer beim Client liegt [Soc12]. Das heißt ein Roaming-Prozess wird im Wireless LAN immer vom Client und nie vom Netzwerk initiiert. Es besteht aber sehr wohl die Möglichkeit, dass ein zentraler Knoten einem Client Informationen über seine Umgebung bereitstellt und somit indirekt Einfluss auf den Roaming-Prozess haben kann.

Im folgenden Kapitel werden unterschiedliche Lösungen, alle mit dem Ziel schnellere und bessere Roaming-Entscheidungen zu treffen, vorgestellt. Des Weiteren werden Vor- und Nachteile der Technologien erarbeitet. Grundsätzlich können Daten die für die Roaming-Entscheidung nützlich sind auf verschiedenen Wegen gewonnen werden. Dies kann entweder über das Scannen der Kanäle erfolgen oder über Abfragen der Informationen von anderen Stationen. Sprich es kann aus Sicht des Clients unterteilt werden in Aggregation von Informationen und Verteilung von Information zwischen Stationen. Eben diese Unterteilung findet auch in den nachfolgenden Abschnitten statt. Die analysierten Varianten werden in selbige Abschnitte eingeteilt.

2.1.1 Aggregation von Informationen

Wie in der Einleitung erwähnt, kann die Verbesserung des Roaming-Prozesses nur durch Nutzung von zusätzlicher Information erlangt werden. Es werden nun Möglichkeiten der Informationsgewinnung diskutiert, die dann als Grundlage für eine verbesserte Roaming-Performance dienen.

Eine Variante ist, die verschiedenen Funkkanäle eines IEEE 802.11 WLAN-Netzwerkes nach Informationen zu scannen. Wie zu Beginn des Kapitels 2 erläutert wurde, kann ein Scan sowohl

passiv als auch aktiv passieren. Viele Ansätze um den Scan zu verkürzen gehen dahin, dass im aktiven Scanmodus nur eine Auswahl an Kanälen, jedoch nicht alle vorhandenen Kanäle gesannt werden. Man bezeichnet dies als Selective Scan Handoff [LC06]. Im Gegensatz dazu gibt es auch den Full Scan Handoff, bei dem alle vorhandenen Funkkanäle abgesucht werden. Je nach dem ob das passive oder aktive Verfahren eingesetzt wird, werden die Kanäle nach Beacon Frames (passiv) oder Probe Responses (aktiv) abgesucht. Im folgenden Abschnitt werden einige Verfahren, welche entweder das Selective Scan Handoff oder das Full Scan Handoff Konzept einsetzen, diskutiert.

Beim Verfahren SyncScan handelt es sich um einen Full Scan Handoff mit optimiertem Kanalwechsel [RS05]. Dies bedeutet, dass die verschiedenen WLAN-Funkkanäle kontinuierlich nach verfügbaren APs gescannt werden. Die Verbesserung liegt in der Taktung des Kanalwechsels. Dieser wird zeitlich so durchgeführt, dass unmittelbar nach dem Wechsel auf einen bestimmten Kanal eine Beacon-Message (Leuchtfeuer-Nachricht) empfangen wird. Sprich der Kanalwechsel wird zeitlich so getaktet, dass auf jedem Funkkanal ein möglichst kurzer Scan notwendig ist um Beacon Messages zu empfangen. Der große Nachteil dieser Methode ist, dass die Scan-Optimierung auf den Kenntnissen, wann welcher AP auf welchem Kanal seine Beacon Messages aussendet, beruht. Das heißt es muss die genaue Konfiguration der APs zur Verfügung stehen. Von dieser Einschränkung kann natürlich im Allgemeinen nicht ausgegangen werden. Darüber hinaus müssen die APs über eine Zeitsynchronisation verfügen. Dies macht die Methode für den breitflächigen Einsatz nicht nutzbar, wenngleich auch der Vorteil besteht, dass auch ein Client mit 802.11-Standardhardware diese Methode verwenden könnte.

Eine andere Methode, die sich mit dem Scannen der Funkkanäle beschäftigt, ist jene des Smooth Handoff [LC06]. Die Idee dabei ist, den aktiven Sendekanal kurz zu verlassen um auf einem anderen Kanal einen Scan nach APs durchzuführen, bevor wieder auf den ursprünglichen Sendekanal zurückgekehrt wird. Die Autoren definierten ihrerseits die Anforderung, dass die Beeinflussung der eigentlichen Datenübertragung so gering wie möglich sein soll. Um dies zu gewährleisten, sollte die Entdeckungsphase in Teile zerlegt werden um den Paketverlust und den Jitter klein zu halten. Das heißt, dass die eigentliche Datenübertragung immer wieder für Zeitabschnitte von 50 ms unterbrochen wird um nach alternativen APs zu suchen. Bei dieser Methode handelt es sich um einen Full Scan Handoff. Im Testaufbau wurde eine Teilung in 11 Abschnitte gewählt. Das heißt, dass in jedem Scan-Ablauf einer von 11 Kanälen gescannt wurde. Die Scan-Dauer eines Kanals wurde mit 50 ms angegeben und die Scan-Pause, also die Zeit, die zur eigentlichen Datenübertragung verwendet werden konnte, wurde ebenfalls auf 50 ms gesetzt. Dies ergab eine Dauer von $11 * (50ms + 50ms) = 1.1s$ um einen vollständigen Scan durchzuführen. Die Entscheidung, ob nach Roaming-Zielen gesucht werden soll, wird aufgrund des RSSI-Wertes entschieden. Aufgrund eines Thresholds, welcher sich der funktechnischen Umgebung anpasst, wird der Start der Entdeckungsphase ausgelöst. Fällt der RSSI-Wert unter einen maximalen Threshold, so wird die Entdeckungsphase und somit der Scan der Kanäle gestartet. Um einen Hin- und Herwechsel zwischen zwei APs zu vermeiden, wurde zusätzlich eine Untergrenze für die Differenz zwischen den RSSI-Werten des neuen und alten AP festgelegt. Die Autoren konnten in einem Testaufbau nachstellen, dass sowohl der Paketverlust als auch die Paketverzögerung verringert werden konnten. Im Test konnten Paketverzögerungen von durchschnittlich 48.1 ms gemessen werden. Mit einer weiteren Verbesserung hinsichtlich des Abbruchs der Entdeckungsphase, wenn ein passender Roaming-Partner gefunden wurde, konnte die durchschnittliche Paketverzögerung sogar auf 33.8 ms gesenkt werden. Ein weiterer Vorteil dieser Methode ist die Verwendbarkeit von 802.11 Standardequipment. Außerdem bedarf es keiner Änderung der APs. Nichts desto trotz stößt das Verfahren vor allem in Gebieten mit geringer AP Dichte an seine Grenzen. So kann es im

schlimmsten Fall passieren, dass ein kompletter Scan aller Kanäle durchgeführt werden muss, was bedeutet, dass die Entdeckungsphase 1.1 s dauert.

Eine Verbesserung des Scans kann auch mit dem Zwischenspeichern und der Wiederverwendung von Scan-Ergebnissen erreicht werden [SFRS04]. Dazu werden Scan-Resultate zwischengespeichert und bei der Rückkehr zu einem bereits besuchten Ort wiederverwendet. Konkret wurde bei einer bestimmten Lokation gespeichert auf welchen Kanälen APs zur Verfügung standen. Bei erneutem Aufsuchen des gleichen Ortes wurden nicht mehr alle, sondern nur mehr zuvor gespeicherte Kanäle gescannt. Der Nachteil dieser Methode ist klarerweise, dass ein kompletter Scan notwendig ist, wenn beim Durchsuchen der gespeicherten Kanäle keine Treffer gefunden wurden. Dies tritt beispielsweise auf, wenn die gespeicherte Information nicht mehr aktuell war.

2.1.2 Verteilung von Information

Wie im Kapitel 2.1.1 erläutert wurde, bedingt ein Scan der Funkkanäle immer eine Unterbrechung der Nutzdatenübertragung. Somit stellt sich die Frage, wie alternativ Daten, die für eine bessere Roaming-Entscheidung gebraucht werden, gewonnen werden können. Es gibt dazu schon eine Reihe von Ansätzen, Simulationen und Implementierungen die im folgenden Kapitel beschrieben werden.

Einer davon ist der Cooperative Handover Algorithmus [AKT08]. Dieser nutzt Kanal- und Nachbarinformationen, um einen schnelleren Handover zu ermöglichen. Es findet dabei eine Unterscheidung zwischen Beacon Report und Neighbor Report statt. Der Beacon Report sieht den Informationsaustausch zwischen zwei Clients vor. Dabei werden Daten über verfügbare APs, deren Kommunikationskanäle, BSSIDs usw. ausgetauscht. Der Neighbor Report wird für den Informationsaustausch zwischen Client und AP aber auch zwischen APs genutzt. Die Simulationen dieses Ansatzes ergaben durchschnittliche Handoff Delays im Bereich von 20 ms bis 25 ms.

Ein weiteres Konzept, das mit Hilfe eines Neighbor List Protocols die Dauer der Entdeckungsphase verkürzen soll, ist jenes des Lightweight Algorithm for Fast IEEE 802.11 Handover [MW12]. Die Idee dabei ist, das Kanal-Scannen komplett zu vermeiden. Stattdessen können Mobile Stationen direkt mit APs aus der Neighbor List Verbindungen aufbauen. Die Neighbor List baut jeder AP für sich auf. Dazu zeichnet ein AP beim Verbindungsaufbau zu einem Client auf, zu welchem AP dieser Client zuvor verbunden war. Die Information des zuvor verbundenen APs erhält der aktuelle AP aus dem Reassociation Request des Clients. Die Neighbor List wird schließlich als Broadcast, wobei der Neighbor Report aus dem Standard 802.11k verwendet wird, weitergegeben. Die Simulation ergab Handover-Zeiten von durchschnittlich 43 ms [MW12].

Eine andere Möglichkeit ist ein Algorithmus, der mit Hilfe von Neighbor Graphs und Non-Overlap Graphs eine Verbesserung des Handoffs erzielt [SMA04]. Der Ansatz dabei ist, die Anzahl an Kanälen, die gescannt werden müssen, zu reduzieren. Ein AP wird von einem anderen AP dann als Nachbar gelistet, wenn ein Client von diesem AP zu ihm wechselt. Beispielsweise wechselt ein Client vom AP1 zum AP2. Nun listet der AP2 den AP1 als Nachbar. Darüber hinaus speichert AP2, auf welchen Kanälen AP1 verfügbar ist. Dies kann später genutzt werden um den Scan von nicht genutzten Kanälen zu vermeiden. Die Struktur, die dabei aufgebaut wird, wird als Neighbor Graph bezeichnet. Die zweite Struktur ist der sogenannte Non-Overlap Graph. Dieser beinhaltet die Information, dass ein Client nicht gleichzeitig mit ausreichender Verbindungsqualität in der Reichweite von beiden APs ist. Der Informationsgehalt des Non-Overlap Graphs ist jener, dass ein Client, welcher einen Probe Response von einem AP empfängt, dessen Reichweite nicht mit

jener eines anderen APs überlappt, ausschließen kann, dass der zweite AP in Reichweite ist. Der Client kann somit auf die Suche nach diesem AP verzichten (man nennt dies auch pruning) und somit die Anzahl an zu scannenden Kanälen reduzieren. Dies verringert wiederum die Dauer der Entdeckungsphase. Eine Testimplementierung ergab, dass durch den Einsatz von Neighbor Graphs die Dauer der Entdeckungsphase im Gegensatz zu einem Full Scan um 80.7 % reduziert werden kann. Durch die Erweiterung des Neighbor Graphs um Pruning, also um die Nutzung des Non-Overlap Graphs konnte die Dauer der Entscheidungsphase sogar um 83.9 % im Vergleich zum Full Scan verkürzt werden [SMA04].

Wie diesem Kapitel zu entnehmen ist, können mit Hilfe von Neighbor Graphs große Verbesserungen erzielt werden. Der Nachteil all dieser beschriebenen Konzepte ist, dass keines davon standardisiert ist. Der Standard 802.11k deckt viele der zuvor beschriebenen Methoden ab. Darüber hinaus wurde der Standard bei mehreren Konzepten bereits zur Übertragung von Nachbarinformationen genutzt und soll deshalb im folgenden Kapitel 2.1.3 genauer beschrieben werden [MW12], [AKT08].

2.1.3 Radio Resource Management of Wireless LANs - IEEE 802.11k

Der Standard IEEE 802.11k bietet Stationen die Möglichkeit ihre Umgebung aus funktechnischer Sicht zu messen und zu verstehen. Dies ist die Grundlage für eine effiziente Nutzung der WLAN Infrastruktur. Für bewegte Clients ist diese Information wichtig, um entscheiden zu können, wann zu einem anderen AP gewechselt werden soll. Dazu werden jedoch nicht nur die Messdaten des einen Clients, sondern auch Daten weiterer Clients im Netzwerk verwendet. Grundsätzlich stellt der Standard IEEE 802.11k ein Service zur Messung und zum Informationsaustausch von Funkparametern bereit. Dabei können Clients lokale Messungen der Funkumgebung durchführen, Anfragen an andere Funkstationen senden, sowie Anfragen von anderen Stationen beantworten. Die Informationen werden darüber hinaus höheren Schichten zur Verfügung gestellt um von einer breiten Masse an Applikationen genutzt werden zu können. Dafür wurden eine Reihe von Messtypen festgelegt, die in diesem Kapitel beschrieben werden [Soc08b].

- **Beacon:** Mittels Beacon Requests/Reports kann eine Station von einer anderen Station eine Liste an APs abfragen, die auf einem bestimmten Kanal erreichbar sind. Die Messung kann passiv, aktiv oder im Beacon Table Mode erfolgen. Im passiven Modus wird ein Timer gesetzt, welcher die Dauer der Datenaufzeichnung festlegt. Für diese Dauer werden von der messenden Station auf dem festgelegten Kanal alle Beacon Nachrichten, Probe Respones und deren Received Channel Power Indicators (RCPI) aufgezeichnet. Im aktiven Modus sendet die messende Station auf dem festgelegten Kanal einen Probe Request und zeichnet wiederum alle Beacon Messages, Probe Requests und RCPIs auf. Im Beacon Table Mode wird die gesamte gespeicherte Beacon-Information für alle unterstützten Kanäle einer festgelegten SSID und BSSID zur Verfügung gestellt, ohne dafür zusätzliche Messungen durchzuführen.

- **Measurement Pilot:** Ist eine Nachricht ähnlich der Beacon Nachricht. Sie wird ebenfalls vom AP periodisch ausgesandt, enthält aber nur einen Teil der Informationen des Beacon Frames. Der Measurment Pilot dient zur Unterstützung der Clients beim Scannen der Kanäle.

- **Frame:** Das Frame Request/Report Paar gibt Auskunft über den gesamten Datenverkehr und die Anzahl der Frames, die eine angefragte Station empfangen und gesendet hat. Diese Anfrage wird ebenfalls von einem AP ausgesandt.

- **Channel Load:** Das Channel Load Request/Report Paar liefert die Auslastung eines Kanals, gemessen von einer Station. Der Channel Load Request wird dazu von einem AP an einen Client gesandt. Der Client antwortet darauf, wie ein bestimmter Kanal aus seiner Sicht ausgelastet ist.

- **Noise Histogram:** Ein AP sendet an einen Client die Anfrage eines Rausch-Histogramms um von diesem eine Messung zu erhalten. Die Aufzeichnung des Clients erfolgt, ohne dass dabei Daten gesendet oder empfangen werden.

- **STA Statistics:** Bei der STA Statistics handelt es sich ebenfalls um eine Anfrage, die ein AP an einen Client sendet. Dieser liefert eine ausführliche Statistik, die diverse Parameter, wie die durchschnittliche Zugriffszeit, Anzahl der übermittelten Nachrichten, Anzahl der fehlgeschlagenen Nachrichten usw. zurück.

- **Location:** Mit dieser Nachricht kann der AP den geografischen Standort (Längengrad, Breitengrad, Höhe) eines Clients abfragen. Dabei kann einerseits der Standort der Station selbst, oder jener eines anderen Clients abgefragt werden.

- **Measurement Pause:** Die Measurement Pause Nachricht bietet die Möglichkeit eine Messung zu verzögern. Auf die Measurement Pause folgt kein Response.

- **Neighbor Report:** Ein Client kann von einem AP abfragen, ob es benachbarte APs gibt, die die gleiche Netzwerkkonfiguration wie der aktuelle AP anbieten. Diese Information soll Clients die Möglichkeit geben, benachbarte APs, welche für ein Roaming zur Verfügung stehen zu finden, ohne dabei einen Kanal-Scan durchführen zu müssen.

  Ein Beispiel für einen Neighbor Report Response ist der Abbildung 2.7 zu entnehmen. Der Neighbor Report Request wird in einem 802.11 Management Action Frame gesendet. Dazu wird der Categorie Code 5 (Radio Measurement) und ein Action Code 4 (Neighbor Report Request) verwendet. Im Neighbor Report Response wird ebenfalls der Categorie Code 5 (Radio Measurement), jedoch als Action Code 5 (Neighbor Report Response) angegeben. Im dargestellten Response sind die BSSID (f0:9c:e9:5a:66:d9), die Kanal Nummer 5 und weitere Informationen zur BSSID ersichtlich. Es wird angegeben, ob der AP erreichbar ist, ob dieser eine gesicherte Verbindung anbietet und welche weiteren Eigenschaften der benachbarte AP bietet. Im der Response Nachricht wird für jeden Nachbar-AP ein eigenes Element eingefügt.

- **Link Measurement:** Diese Anfrage gibt dem AP die Möglichkeit von einem Client Informationen über die Qualität der Funkverbindung zu erfragen.

- **Transmit Stream/Category Measurement:** Diese Nachricht bietet für einen AP die Möglichkeit die Qualität einer Datenübertragung zu einem Client mit Quality of Service Anforderungen, zu erfragen. Dabei kann ein AP beispielsweise auch Bedingungen angeben, wann ein Report vom Client an den AP zurückgeschickt werden soll. Somit kann der AP vom Client fordern, dass dieser, falls eine Verbindung unter eine bestimmte Qualitätsgrenze fällt, einen Report schicken soll.

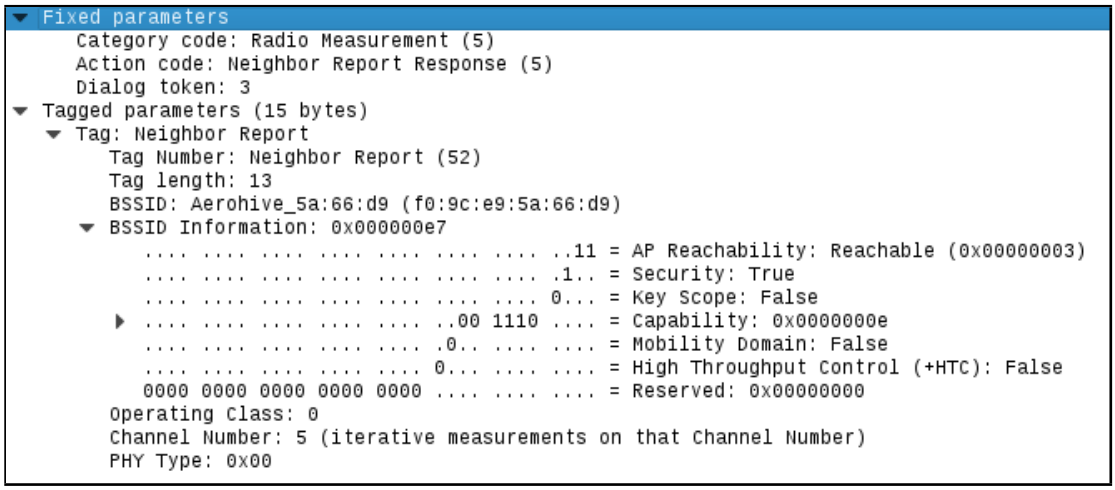

Abbildung 2.7: Beispiel eines 802.11k Neighbor Report Responses - aufgezeichnet mit Wireshark

Der 802.11k Standard deckt somit den Austausch und die Bereitstellung von Informationen bezüglich der funktechnischen Umgebung eines Clients ab. Was jedoch nicht behandelt wird, ist die Auswahl eines konkreten APs für das Roaming. Das heißt, die tatsächliche Auswahl eines Ziel-APs beim Handover wird offen gelassen. Dies bedeutet weiter, dass es schlussendlich am Hersteller von Netzwerkhardware oder an Applikationen liegt, ob die Informationen, die mittels 802.11k gesammelt werden auch für die Beschleunigung des Roamings eingesetzt werden.

Eine Lösung, die den Standard IEEE 802.11k nutzt, ist der Proactive Neighbor Caching Algorithmus [SFRS04]. Dabei wird eine Frequent Handoff Region (FHR) definiert, also ein Bereich in dem alle APs liegen, zu denen in näherer Zukunft gewechselt werden kann. Falls ein Client einen Authentication Request an den AAA-Server (Authentication, Authorization, Access Control) sendet, so schickt dieser die Authentication Response an alle APs im FHR. Somit ist der Client bei allen APs im FHR vorauthentifiziert und braucht beim Handover keine Authentifizierung mehr durchzuführen. Die FHR legt jedoch keinen AP fest, zu welchem bevorzugt gewechselt werden soll. Das führt dazu, dass gerade bei größeren WLAN-Netzwerken eine erhebliche Last für den AAA-Server und das Netzwerk entsteht, da bei allen APs in der FHR eine Vorauthentifizierung durchgeführt werden muss.

Eine anderes Konzept ist das der Proactive Key Distribution [MSJ+04]. Proactive Key Distribution ist dem zuvor erläuterten Proactive Neighbor Caching sehr ähnlich. Jedoch wird ein Neighbor Graph benutzt um die Anzahl an APs im FHR an die der Authentication Response gesendet wird, zu reduzieren. Der Nachteil dabei ist jedoch, dass der Neighbor Graph statisch gehalten wird und somit in der Praxis nicht verwendbar ist.

Die Methode Selective Neighbor Caching stellt wiederum eine Verbesserung des Proactive Neighbor Caching dar [PJKC05]. Dabei passiert eine Gewichtung der verschiedenen APs. Eine Vorauthentifizierung findet nur dann statt, wenn das Gewicht größer einer definierten Grenze (threshold) ist. Problematisch dabei ist, dass das Gewicht nur aus alten Verbindungen berechnet wird und dadurch ein Ungleichgewicht entstehen kann. Dies kann dazu führen, dass sich die Netzwerklast auf wenige APs aufteilt.

Eine Erweiterung ist das Konzept des Adaptive Neighbor Caching [YMW08]. Diese basiert auf der Überlegung, dass sich Personen meistens entlang bestimmter Pfade und nur selten zufällig

bewegen. Daraus folgt, dass die Möglichkeiten des Handoffs auf wenige APs eingeschränkt werden können. Dazu wird aus der Neighbor List eine Kandidatenliste extrahiert, die jene APs enthält, die wahrscheinlich als Roaming-Partner zur Verfügung stehen.

Abschließend kann gesagt werden, dass die Erweiterung 802.11k einige Funktionen mit sich bringt, die für schnelleres Roaming grundlegend sind. Der Standard stellt also eine breite Palette an Möglichkeiten zur Analyse der Netzwerkinfrastruktur da, jedoch bleibt die Entscheidung, zu welchem AP tatsächlich gewechselt werden soll, unbeantwortet. Die Roaming-Performance hängt letztendlich davon ab, wie gut die Informationen die mit den Nachrichten aus dem Standard gewonnen werden können, genutzt werden. Zur Zeit der Verfassung dieser Arbeit gab es tatsächlich nur sehr wenige Systeme die den Standard vollständig unterstützten. Eine teilweise Unterstützung gab es zwar auf Seiten der Netzwerkinfrastruktur Hersteller. Auf der Client-Seite war jedoch kaum Hardware zu finden, die den Standard bereitstellten. Auch die Unterstützung aktueller Betriebssysteme war nur sehr eingeschränkt gegeben.

2.2 Authentifizierungsphase

Wie in der Einleitung des Kapitels 2 beschrieben wurde, ist die Authentifizierungsphase die zweite Phase, in der eine Verkürzung des Roaming-Prozesses erfolgen kann. Grundlegend gibt es dabei zwei Möglichkeiten um Zeit einzusparen. Die Erste wäre keine oder wenig komplexe Sicherungsmechanismen zu verwenden. Damit würde die Authentifizierungsphase komplett oder zumindest teilweise entfallen. Dies ist natürlich in einem professionellen Netzwerk, auf dem sensible Daten übertragen werden, unmöglich. Die Risiken, die ein schlecht oder gar nicht gesichertes Netzwerk birgt, dürfen nicht in Kauf genommen werden. Deshalb wird in weiterer Folge auf die Beschreibung von Verfahren wie Wired Equivalent Privacy (WEP) verzichtet, da diese als veraltet und nicht sicher gelten [BHL06]. Der Ablauf beim Verbindungsaufbau zu einem ungesicherten Netzwerk soll an dieser Stelle jedoch kurz ausgeführt werden. Dieser dient zu Vergleichszwecken und stellt den kürzest möglichen Ablauf eines Verbindungsaufbaus in einem IEEE 802.11 Netzwerk dar und ist der Abbildung 2.8 zu entnehmen.

Die Kommunikation beginnt ausgehend vom Client mit einer Probe Request Nachricht. In dieser gibt der Client die unterstützten Datenraten und weitere 802.11 Eigenschaften, wie die Unterstützung vom Standard 802.11n an. Der AP antwortet mit einer Probe Response Nachricht, in der dieser die SSID, unterstützte Datenraten und Verschlüsselungsmethoden und weitere 802.11 Eigenschaften, die von diesem bereitgestellt werden, angibt. Dann erfolgt der Open System Authentication Request des Clients. Der AP antwortet darauf mit einem Open System Authentication Response. Dieser ist nicht gleichbedeutend mit den WPA2 oder 802.1X Authentifizierungsmethoden. Diese erfolgen erst nach dem Verbindungsaufbau zum Netzwerk. Ursprünglich war diese Authentifizierungsnachricht für den WEP-Standard vorgesehen. Dieser gilt jedoch, wie zuvor bereits angemerkt, als unsicher und wird deshalb nicht mehr genutzt. Das bedeutet, dass der AP in der Antwort meistens die Authentifizierung als offen angibt. Die eigentliche Authentifizierung folgt dann nach der Assoziation. Im Association Request gibt der Client schließlich an, zu welchem AP eine Verbindung aufgebaut werden soll. Darüber hinaus wird dem AP die gewählte Authentifizierungsmethode bekanntgegeben. Ist der AP mit den geforderten Einstellungen des Clients einverstanden, generiert er für diesen eine Association ID und übermittelt die ID in einer Association Response Nachricht.

Die zweite Möglichkeit Roaming zu beschleunigen, ist die Reduktion der Nachrichten, die zwischen Client und AP beim Verbindungsaufbau ausgetauscht werden müssen. Dazu werden im

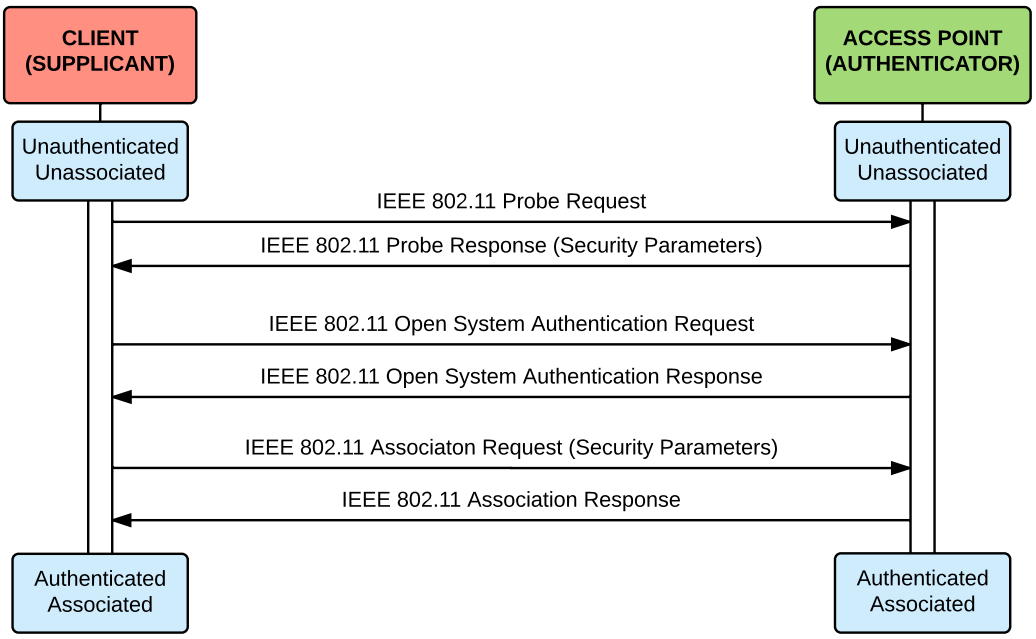

Abbildung 2.8: Ablauf des Verbindungsaufbaus mit einem ungesicherten 802.11 Netzwerk

folgenden Kapitel nun verschiedene Verfahren, welche den WPA2 Standard verwenden, im Bezug auf deren Verwendbarkeit für ein schnelles Roaming gegenübergestellt. Aktuell gilt der Standard WPA2 als sicher, solange bei der Konfiguration bestimmte Regeln, wie ein ausreichend komplexes Passwort, beachtet werden [CC15]. WPA2 implementiert den Sicherheitsstandard 802.11i für Funknetzwerke. Im Standard 802.11i, der auch den Namen Robust Security Network (RSN) trägt, wurde der 802.11 Standard um die Protokolle Temporal Key Integrity (TKIP) und Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP) erweitert [Soc12]. WPA ist eine Implementierung des TKIP Protokolls und WPA2 eine Implementierung des CCMP Protokolls. Zusammengefasst beinhaltet die Erweiterung 802.11i die Protokolle

- TKIP für Datenverschlüsselung und Message Integritätscheck (MIC) basierend auf dem Rivest Cipher 4 (RC4) Algorithmus
- Advanced Encryption Standard-CCMP (AES-CCMP) für Datenverschlüsselung und MIC basierend auf dem AES
- 802.1X Authentication Extensible Authentication Protocol (EAP) zur Authentifizierung von Clients und zur Verteilung der benötigten Schlüssel

Dabei ist EAP ein Standard der Internet Engineering Task Force (IETF) und unterstützt verschiedenste Verfahren wie RADIUS, EAP-Transport Layer Security (EAP-TLS), digitale Zertifikate (EAP-Message Digest 5 (MD5)) oder SIM-Karten Authentifizierung (EAP-SIM) [ABV+04]. EAP ist für die Authentifizierung und TKIP und CCMP für Datenverschlüsselung und Integrität von Nachrichten zuständig. All diese Verfahren werden im 802.11i Standard verwendet.

Es folgt ein Vergleich verschiedener Methoden, alle mit dem Ziel das Roaming in einem WLAN-Netzwerk zu beschleunigen.

2.2.1 Wi-Fi Protected Access Pre-Shared Key

Das Verfahren WPA/WPA2 Pre-Shared Key, auch bezeichnet als WPA/WPA2-Personal basiert darauf, dass der Master Key sowohl dem AP als auch allen Clients bekannt ist [wif12]. Das heißt der Master Key muss, bevor eine Verbindung aufgebaut werden kann, manuell konfiguriert werden. Die Nachrichtenabfolge sieht ebenfalls die Open Authentication und Association Frames vor, bevor ein 4-Way-Handshake durchgeführt wird (siehe Abbildung 2.9).

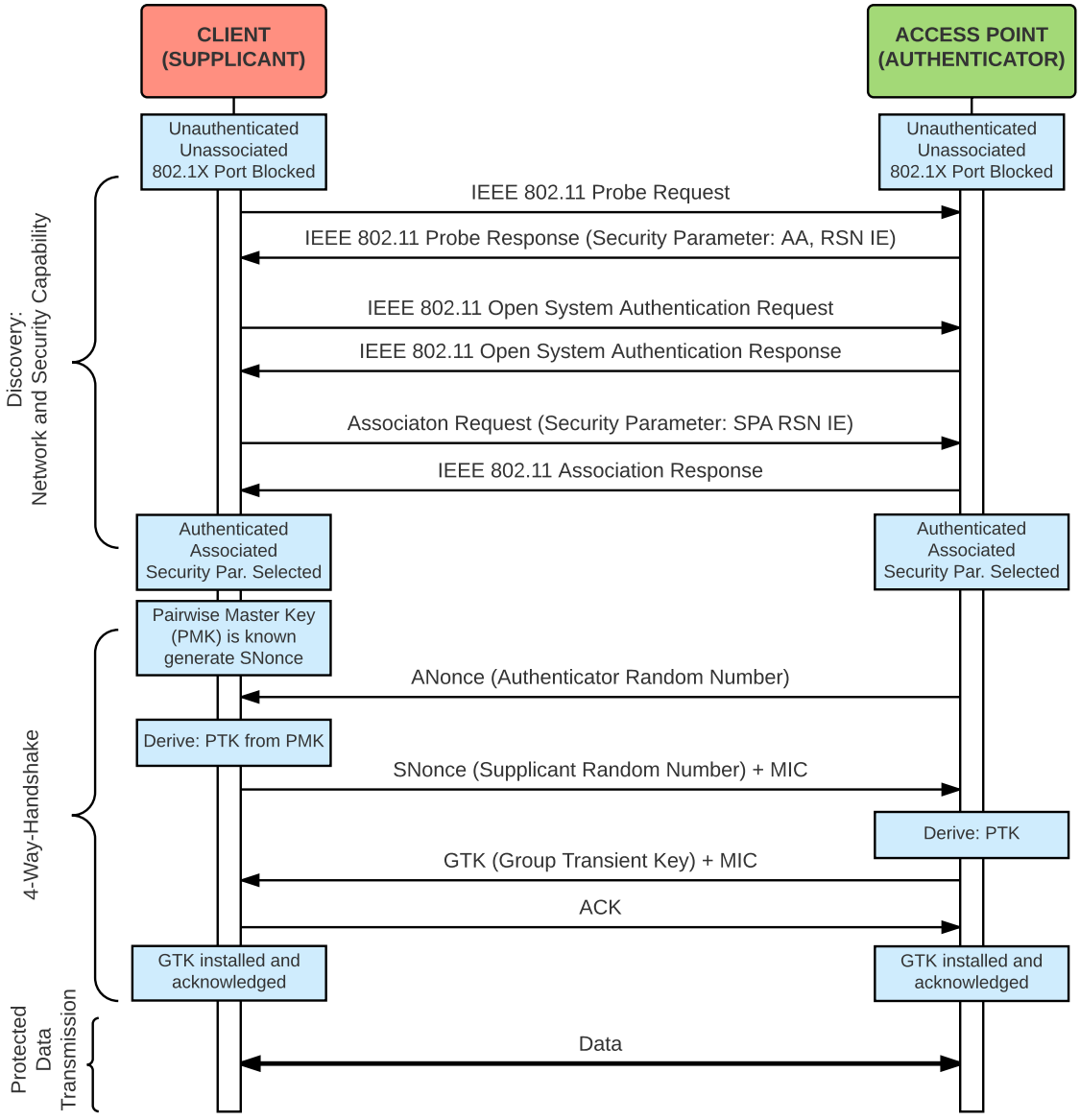

Abbildung 2.9: Ablauf des Verbindungsaufbaus unter der Verwendung von WPA2-Pre-Shared Key

Der 4-Way-Handshake ist dazu da um einerseits den sogenannten Pairwise Transient Key (PTK) zur Verschlüsselung von Unicast-Verbindungen und andererseits den Group Temporal Key (GTK) zur Verschlüsselung von Multicast-Verbindungen zu generieren. Die Erzeugung des PTK erfolgt

über eine Pseudo-Random Funktion. Diese benötigt den PMK, ANonce, SNonce, und die MAC-Adressen sowohl des Clients als auch des APs um den PTK generieren zu können.

Der 4-Way-Handshake beginnt mit dem Senden der Authenticator Random Number (ANonce) vom AP an den Client. Dies ist eine vom AP generierte Zufallszahl. Nun ist der Client im Besitz aller benötigten Informationen und kann den PTK ableiten. An den AP wird die Supplicant Random Number (SNonce), also eine vom Client erzeugte Zufallszahl zurückgeschickt. Nun hat auch der AP alle Informationen um den PTK ableiten zu können. Darüber hinaus nutzt der AP den mitgeschickten MIC um die Integrität der Nachricht zu überprüfen. Dies bedeutet, dass nach der Übertragung der zweiten Nachricht sowohl der Client als auch der AP den PTK haben und somit Unicast-Verbindungen verschlüsseln können. Im zweiten Schritt wird nun GTK vom AP an den Client übermittelt. Diese Nachricht wird bereits mit dem PTK verschlüsselt übertragen. Der Client antwortet mit einer abschließenden Acknowledgement Nachricht um dem AP mitzuteilen, dass die temporären Schlüssel installiert wurden. Aufgrund der manuellen Verteilung des Master Keys fällt eine Authentifizierung mittels EAP weg.

Dies ist jedoch auch gleichzeitig der Nachteil dieses Verfahrens. In einem Netzwerk mit einer großen Anzahl an Clients ist die Verteilung des Schlüssels aufwendig. Der Vorgang der manuellen Schlüsselverteilung muss bei Änderung des Schlüssels wiederholt werden. Darüber hinaus ist eine Rechteverwaltung für einzelne Clients nicht möglich. Das heißt es ist beispielsweise nicht möglich Clients nachträglich den Zugang zu verwehren. Man denke an folgendes Szenario, das in Firmennetzwerken häufig auftritt: Ein Mitarbeiter verlässt das Unternehmen. Der Zugriff dieses Mitarbeiters zum WLAN soll ab diesem Zeitpunkt nicht mehr möglich sein. Dazu müsste ein neuer Master Key konfiguriert werden und an alle Clients verteilt werden.

Darüber hinaus besteht bei Verwendung von WPA-PSK folgendes Sicherheitsproblem. Jeder Client verschlüsselt seine Nachrichten mit einem eigenen PTK. Der PTK wird jedoch vom Master Key abgeleitet, was wiederum bedeutet, dass jeder Client im Netzwerk den PTK eines anderen Clients ableiten und somit dessen Daten entschlüsseln kann. Dazu muss ein Client lediglich den 4-Way-Handshake Prozess eines Dritten abhören. Daher ist WPA2-PSK für industrielle, sicherheitskritische Anwendungen nicht geeignet.

Zu Vergleichszwecken wurde die Dauer des Verbindungsaufbaus mit einem mit dem Standard WPA2-PSK gesicherten Netzwerk ebenfalls gemessen. Die relative Häufigkeit der Dauer ist der Abbildung 2.10 zu entnehmen. Die Messungen umfassten die Authentifizierungs- und Assoziierungsnachrichten, sowie den 4-Way-Handshake. Im Gegensatz zu Abbildung 2.6 wurde bei dieser Messung der Scan-Prozess ausgeklammert. Im Mittel ergab sich eine Dauer von 25 ms. Somit kann WPA2-PSK durchaus als schnelles Verfahren bezeichnet werden, ist jedoch aufgrund der zuvor erwähnten Einschränkungen und Risiken nur für den Gebrauch in Heimnetzwerken denkbar.

2.2.2 Wi-Fi Protected Access Full Authentication - 802.1X

Full Authentication bedeutet, dass ein Authentifizierungsprozess nach dem Standard 802.1X durchgeführt wird [Soc12]. Dazu gibt es eine eigene Instanz, die für die Authentifizierung zuständig ist. Dies kann zum Beispiel ein RADIUS-Server oder auch ein Diameter-Server sein.

Der grundsätzliche Ablauf einer Authentifizierung nach dem Standard 802.1X ist in Abbildung 2.11 skizziert. Nach dem erfolgreichen Austausch des Probe Request/Response Paares, des Authentication Request/Response Paares und des Association Request/Response Paares beginnt der

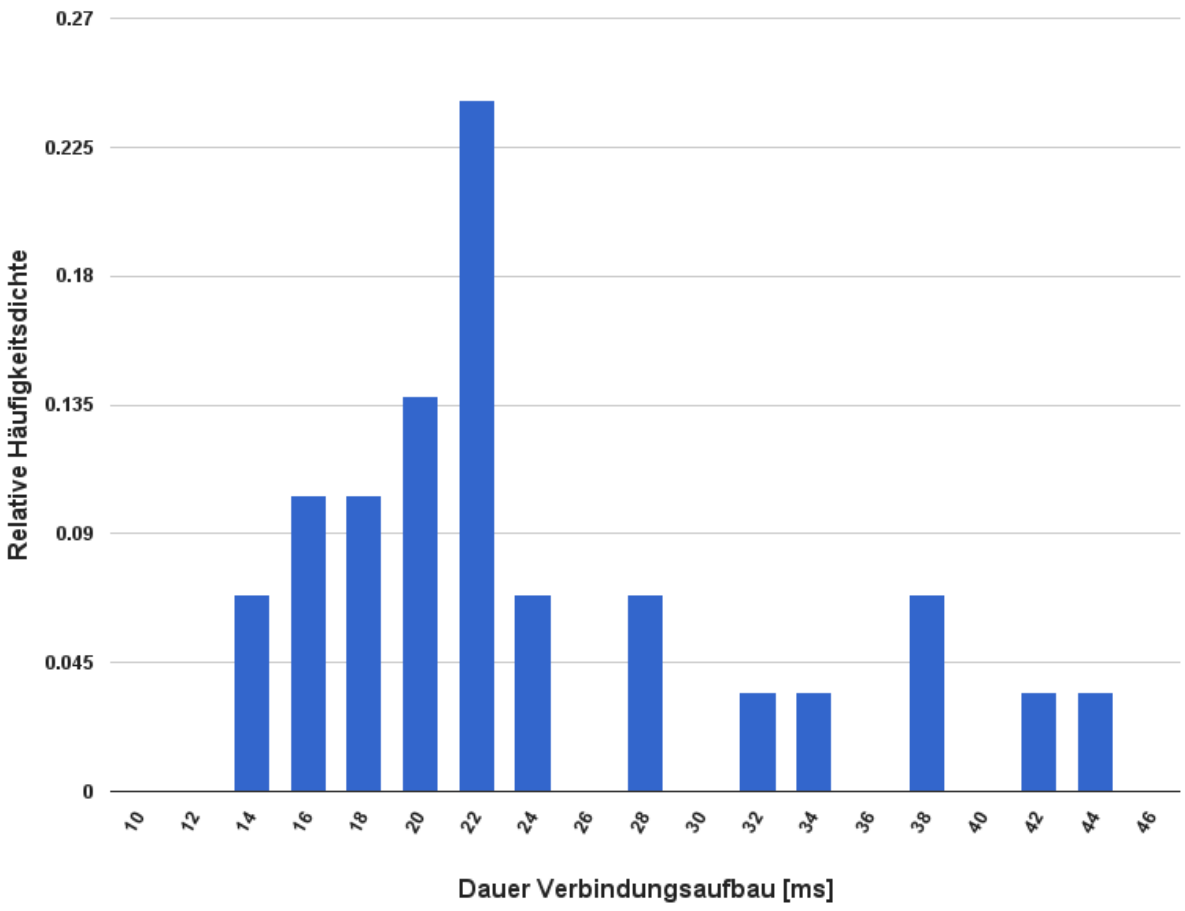

Abbildung 2.10: Relative Häufigkeitsdichte der Dauer eines Verbindungsaufbaus zu einem mit WPA2-Personal gesicherten Netzwerk

eigentliche 802.1X EAP Ablauf. Im Association Request wird dazu eine bestimmte Authentifizierungsmethode, ein Key-Management Verfahren und eine Verschlüsselungstechnik festgelegt. Im Falle der Abbildung 2.11 wird zur Authentifizierung EAP eingesetzt. Nach erfolgreicher Authentifizierung am RADIUS-Server sendet selbiger den Master Session Key (MSK) an den AP. Danach findet ein 4-Way-Handshake zur Ableitung des PTK und des Group Transient Key (GTK) statt. Die Ableitung erfolgt analog dem Verfahren WPA/WPA2 Pre-Shared Key. Der grundsätzliche Unterschied liegt darin, dass der Master Key, nicht wie bei WPA/WPA2 Pre-Shared Key vorab manuell verteilt wird, sondern dass dieser mit dem EAP Verfahren zwischen Client und Authentifizierungsserver ausgehandelt wird.

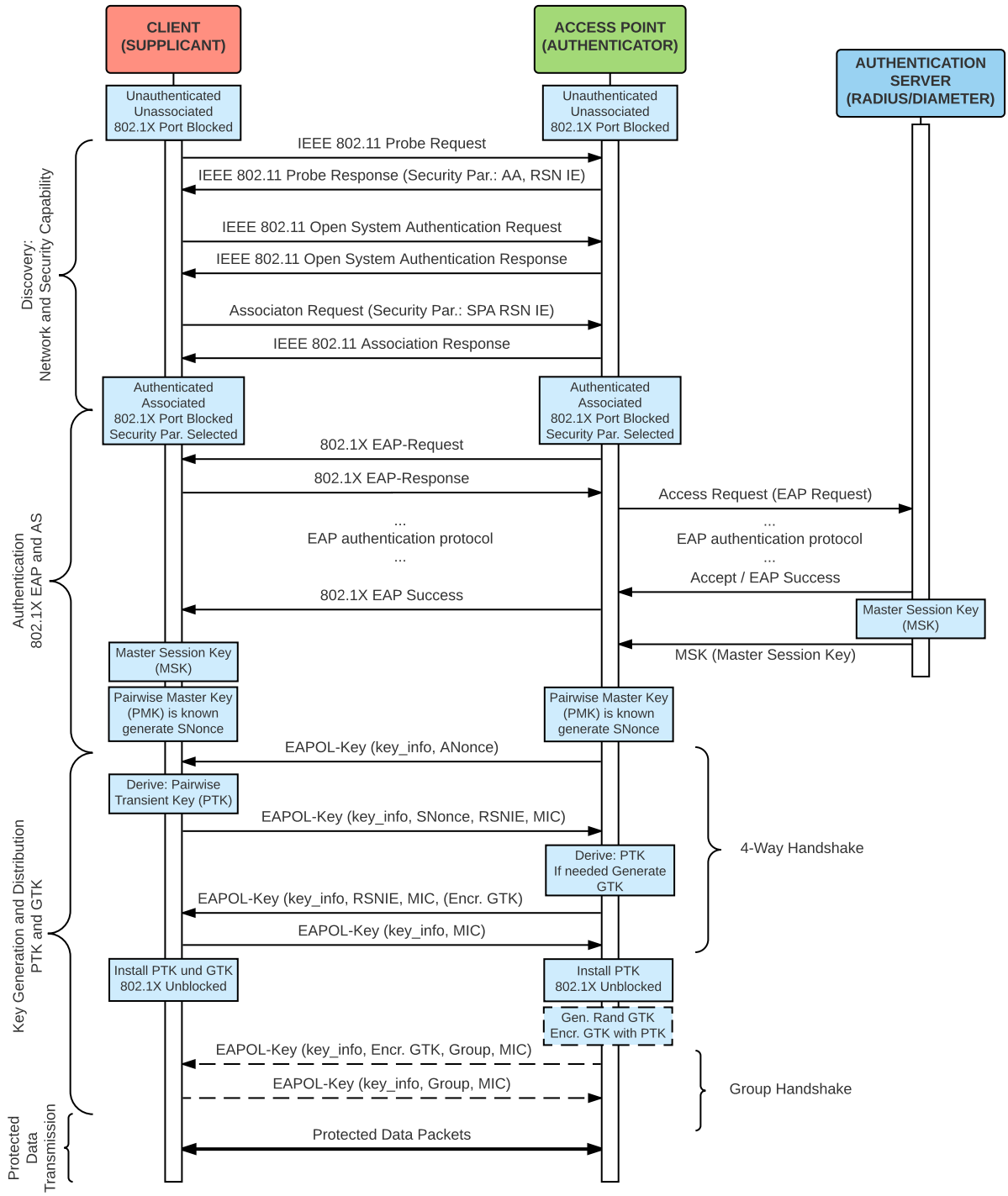

Abbildung 2.11: Verbindungsaufbau nach dem Standard 802.11i unter der Verwendung von EAP als Authentifizierungsmethode [TTHJ09]

Dazu wird im Folgenden kurz die Notwendigkeit der aufwändigen Schlüssel Hierarchie im 802.11 Standard beschrieben [Soc12]. Grundsätzlich gibt es verschiedene Verbindungssituationen, die

sich in ihren Sicherheitscharakteristiken unterscheiden. Die erste Situation ist der Unicast, also eine 1 zu 1 Verbindung, wie es beispielsweise bei WLAN die Verbindung zwischen einem Client und einem AP ist. Die andere Situation ist die des Multicasts, wo eine Verbindung zu mehreren Stationen besteht, also eine 1 zu n Verbindung. Daten, die über eine Unicast-Verbindung gesendet werden, dürfen auch nur für die beiden Verbindungspartner lesbar sein. Hierfür gibt es einen eigenen Key, den Pairwise Key, der nur einem bestimmten Client und einem bestimmten AP bekannt sind. Der Pairwise Key wird also verwendet um eine 1 zu 1 Verbindung zu sichern. Für eine Multicast-Verbindung muss es wiederum einen eigenen Schlüssel geben, der allen Stationen, die zur Multicast-Gruppe gehören, zur Verfügung steht. Dieser wird als Group Key bezeichnet. Zusammengefasst gibt es also zwei Schlüssel-Hierarchien, die Pairwise Key Hierarchie für 1 zu 1 Verbindungen und die Group Key Hierarchie für 1 zu n Verbindungen.

Der Pairwise Master Key (PMK) wird vom Master Session Key (MSK) abgeleitet. Des Weiteren wird der PTK vom PMK abgeleitet. Der GTK wird wiederum vom Group Key abgeleitet. Zur Verschlüsselung werden schlussendlich der PTK für die unidirektionale Verbindung zwischen Client und AP und der GTK für die Multicast-Gruppe eingesetzt. Es sei an dieser Stelle noch angemerkt, dass auch die transienten Schlüssel noch weiter unterteilt werden um sowohl für die Verschlüsselung als auch für die Sicherung der Datenintegrität einen eigenen Schlüssel verwenden zu können. Dies soll hier jedoch nicht weiter ausgeführt werden.

Im 802.11i Ablauf passiert die Einigung auf den PTK mittels eines 4-Way-Handshakes. Der GTK ist der letzte Schritt im 802.11i Verbindungsaufbau und wird als Group Handshake bezeichnet. Wie dem Ablaufdiagramm 2.11 zu entnehmen ist, handelt es sich beim 802.11i Verbindungsaufbau um einen aufwändigen Prozess, der je nach physikalischem Standort des Authentifizierungsservers eine Dauer von mehreren 100 ms übersteigen kann [TTHJ09].

Eine Beispielmessung (siehe 2.3) ergab eine Dauer von 287 ms für den Authentifizierungsvorgang (gemessen von der ersten Authentifizierungsnachricht (Nachricht Nr. 17) bis zur letzten Key-Nachricht (Nachricht Nr. 47)). Die relative Häufigkeit einer Messreihe des Verbindungsaufbaus zu einem mit WPA2-Enterprise gesicherten Netzwerk ist dem Diagramm 2.12 zu entnehmen. Die Messung ergab eine durchschnittliche Dauer von 265 ms für den Ablauf des gesamten Authentifizierungsprozesses.

2.2.3 Wi-Fi Protected Access Extensible Authentication Protocol Session Resumption

Viele Authentifizierungstypen die unter dem EAP-Protokoll angewendet werden können, basieren auf TLS [GME11]. TLS verwendet jedoch einen aufwendigen Handshake-Prozess um eine sichere Verbindung zwischen einem Client und einem Authentifizierungsserver herzustellen. Der Handshake setzt dabei ein serverseitiges Zertifikat voraus, um eine Authentifizierung des Servers gegenüber des Clients zu ermöglichen. Nachdem sich der Server beim Client authentifiziert hat, muss sich umgekehrt auch der Client beim Server authentifizieren. Bei EAP-TLS geschieht dies mit einem clientseitigen Zertifikat. In weiterer Folge muss noch ein 4-Way-Handshake zur Ableitung der Keys erfolgen.

Falls eine erfolgreiche Authentifizierung stattgefunden hat, besteht jedoch die Möglichkeit die TLS Session, sowohl am Client als auch am Server zu speichern. Das bedeutet, dass bei einer erneuten Authentifizierung, beispielsweise nach einem Roaming-Prozess, auf die gespeicherte Session zurückgegriffen werden kann. Somit kann der Prozess um das Authentifizierungsverfahren

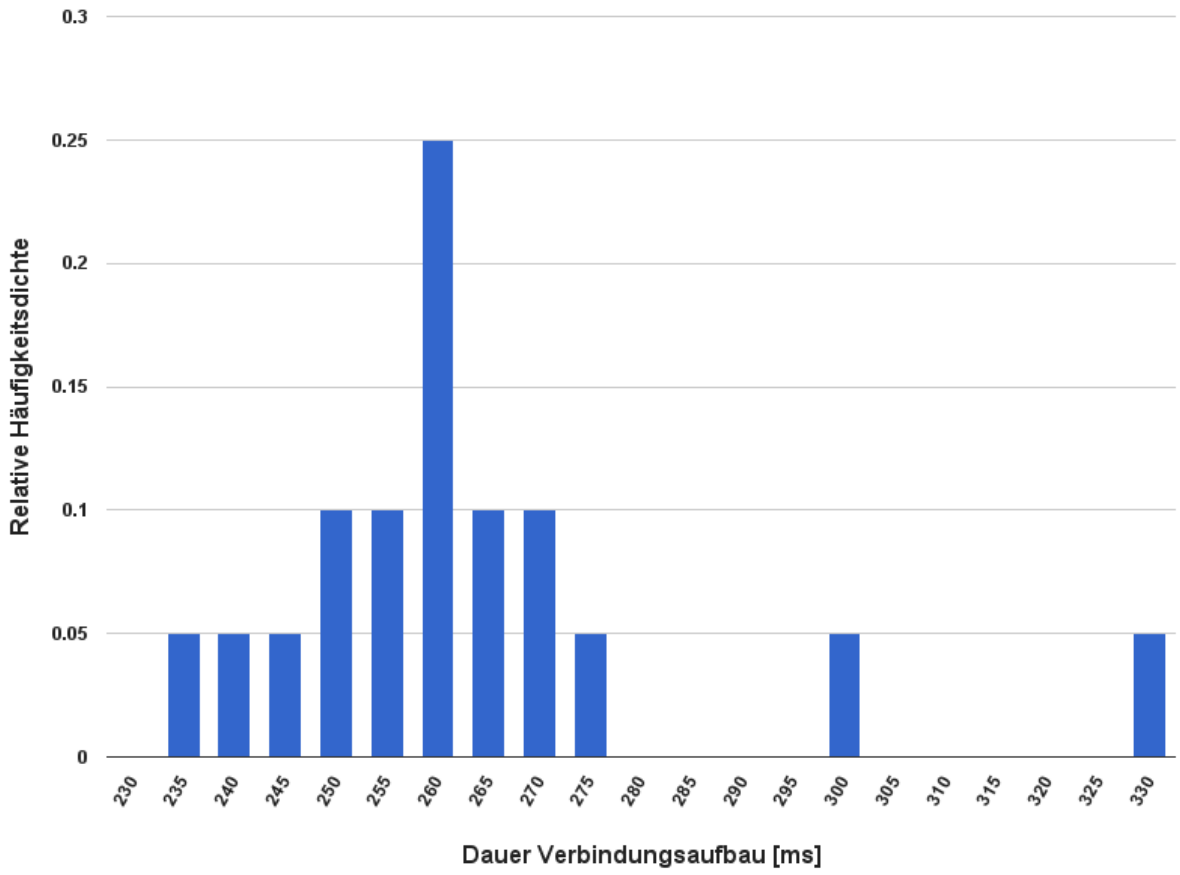

Abbildung 2.12: Relative Häufigkeitsdichte der Dauer eines Verbindungsaufbaus zu einem mit WPA2-Enterprise gesicherten Netzwerk

verkürzt werden. Es folgt eine Reduktion der auszutauschenden Nachrichten um 50 % [5]. Darüber hinaus ist die Session Resumption transparent für die WLAN-Infrastruktur und erscheint als herkömmliche 802.1X Authentifizierung.

Es sind dabei Roaming-Zeiten von unter 300 ms möglich [5]. Kritisch kann sich jedoch der physikalische Standort des Authentifizierungsservers auswirken. Nichts desto trotz reicht die erzielbare Performance nicht aus um Echtzeit-Applikationen wie Voice over IP zu betreiben.

2.2.4 Wi-Fi Protected Access Key Caching

Pairwise Master Key Caching ist eine Möglichkeit, die ursprünglich 2004 in der 802.11 Erweiterung 802.11i standardisiert wurde. Die Erweiterung wurde 2007 in den 802.11-2007 Standard und schlussendlich 2012 in den 802.11-2012 Standard übernommen [Soc12].

Beim PMK Caching speichern sowohl Client als auch AP eine erfolgreiche Authentifizierungs-Session zwischen. Diese können für spätere Verbindungen wiederverwendet werden. Falls ein Client zu einem AP wechselt, mit dem er zuvor schon eine vollständige Authentifizierung nach dem Standard 802.1X erfolgreich durchgeführt hat, kann die gespeicherte Session wiederverwendet

werden. Der Verbindungsaufbau kann somit um die Authentifizierung verkürzt werden und es kann direkt mit dem 4-Way-Handshake zum Austausch und Ableitung der Schlüssel fortgefahren werden. Der exakte Roaming-Prozess läuft so ab, dass der Client den sogenannten PMK-Identifier (PMKID) einer zwischengespeicherten Session mittels eines RSN Information Elements innerhalb des Reassociation Request an den AP schickt. Falls auch der AP die alte Session gespeichert hat, ist die Authentifizierung somit nicht nötig.

Der gesamte Ablauf ist in Abbildung 2.13 dargestellt. Dabei findet ein Wechsel der Verbindung von AP1 zum AP2 statt. Es ist zu sehen, dass im Reassociation Request der PMKID an den AP übermittelt wird. Dadurch muss keine erneute Authentifizierung stattfinden. Danach folgt der 4-Way-Handshake zum Ableiten der Schlüssel. Laut [5] sind mit PMK Caching Roaming-Abläufe im Bereich von unter 100 ms möglich.

Es sei angemerkt, dass eine zwischengespeicherte Session immer nur zwischen einem bestimmten Client und einem bestimmten AP funktioniert. Eine gespeicherte Session kann nicht zwischen AP weitergereicht werden. Dies bedeutet, dass beim initialen Verbindungsaufbau zu einem AP immer die vollständige 802.1X Authentifizierung ablaufen muss. Es ist außerdem zu beachten, dass Clients und AP nur eine bestimmte Anzahl an Sessions zwischenspeichern können. Des Weiteren laufen Sessions nach einer bestimmten Zeit ab. Das bedeutet wiederum, dass dann erneut eine vollständige Authentifizierung notwendig ist. In Netzwerken, wo oft nur zwischen wenigen APs gewechselt wird, kann PMK Caching jedoch einen immensen Vorteil bringen.

Weitere Verfahren sind das WPA2-Proactive Key Caching (WPA2-PKC) und das Opportunistic Key Caching.

PKC baut auf das in 2.2.4 besprochene Verfahren PMK Caching auf. Der Vorteil ist jedoch, dass eine zwischengespeicherte Session (PMKSA) über APs hinweg genutzt werden kann. Voraussetzung dafür ist, dass die APs entweder mit dem gleichen WLAN Controller verbunden sind, oder dass sie der gleichen AP Group angehören. Bei PKC wird eine initiale Authentifizierung nach dem 802.1X Standard an einem zentralen Punkt (z. B. am WLAN Controller) gespeichert. Will ein Client nun im gleichen ESS einen Roaming-Vorgang durchführen, so berechnet dieser einen neuen PMKID für den neuen AP, basierend auf der BSSID des neuen APs. Der Client sendet schließlich die berechnete PMKID an den neuen AP. Dies passiert, wie schon bei PMK, mit dem RSN Information Element innerhalb des Reassociation Requests. Der neue AP fragt nun beim zentralen Knoten die PMKID, welche er vom Client empfangen hat, an. Aus den Daten vom zentralen Knoten berechnet der AP erneut die PMKID um sie mit jener des Clients vergleichen zu können. Stimmen diese überein, kann der Authentifizierungsprozess übersprungen werden und es kann direkt mit dem 4-Way-Handshake zur Ableitung der Schlüssel fortgefahren werden. Dies bedeutet, dass ein Client zwar für jeden AP einen neuen PMKID berechnen muss, jedoch der zwischengespeicherte PMKSA wiederverwendet werden kann. Somit kann auf die vollständige Authentifizierung nach dem 802.1X Standard verzichtet werden. Laut [5] ist die Performance ähnlich dem PMK, also im Bereich von unter 100 ms. Zu beachten ist jedoch, dass PKC nicht standardisiert ist und somit die Implementierungen vom jeweiligen Hersteller abhängen und variieren können.

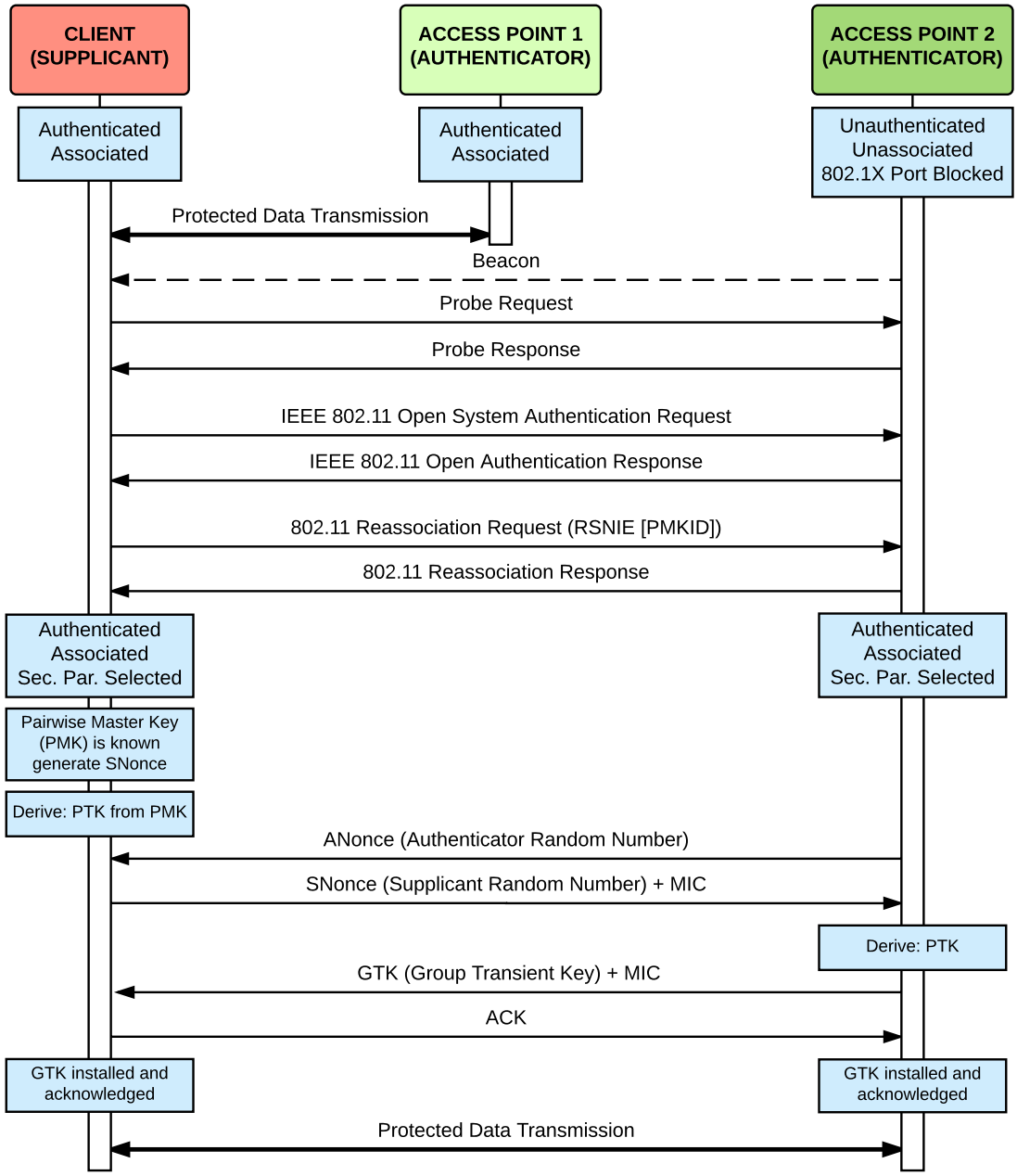

Abbildung 2.13: Ablauf des Verbindungsaufbaus unter der Verwendung von WPA2 Key Caching

2.2.5 Cisco Centralized Key Management

Bei Cisco Centralized Key Management (CCKM) handelt es sich um ein proprietäres Protokoll von Cisco [1]. Das Konzept hinter CCKM ist das Caching von Schlüsseln. Der Schlüssel wird dabei sowohl auf dem Wireless Distribution System (WDS) Master als auch auf dem Client zwischengespeichert. Es gibt einen zentralen Knoten, der die Rolle des WDS inne hat. Dies kann ein autonomer AP, eine Wireless LAN Solution Engine (WLSE) oder ein WLAN Controller sein.

Der Roaming-Ablauf passiert so, dass ein Client, welcher zu einem anderen AP wechseln will,

die Re-Key Nummer inkrementiert und aus der BSSID des gewünschten APs einen neuen PTK ableitet. Für die Übertragung des Re-Keys ist ein proprietäres Information Element notwendig. Der AP fragt dann den neuen PTK beim WDS Master an und antwortet dem Client mit einem Association Frame. Das bedeutet, dass beim CCKM sowohl die EAP Authentifizierungsphase als auch der 4-Way-Handshake im Gegensatz zum 802.11i Ablauf eingespart werden. Somit sind Roaming-Abläufe von unter 50 ms möglich [5].

Der Ablauf von CCKM wird an dieser Stelle kurz beschrieben. Der grundlegende 802.11 Verbindungsaufbau mit Authentication Request/Response und Association Request/Response läuft ab wie in Kapitel 2.2.2 und in Abbildung 2.11 ersichtlich. Ebenfalls wird nach dem EAP-Request der MSK vom RADIUS-Server bereitgestellt. Der MSK wird nun aber nicht, wie im 802.11i-Ablauf, direkt zum AP geschickt, sondern an den WLC, also den zentralen Knoten des Netzwerks. Der WLC speichert den MSK für die Dauer der Verbindung des Clients zum Netzwerk. Der MSK wird in weiterer Folge zur Ableitung der Keys benötigt. Es folgt ein 4-Way-Handshake zwischen Client und WLC um den PTK zur Verschlüsselung von Unicast-Verbindungen und den GTK zur Verschlüsselung von Multicast-Verbindungen ableiten zu können. Bis zu dieser Stelle ist der Ablauf dem 802.11i (siehe 2.11) sehr ähnlich.

Der große Unterschied erfolgt jedoch erst beim Roaming-Ablauf. Dabei sendet der Client einen einzelnen Reassociation Request zum AP bzw. zum WLC, welcher den MIC, eine sequentiell inkrementierte Zufallszahl (Nonce) und Informationen, wie die MAC Adresse des neuen APs beinhaltet um einen neuen PTK ableiten zu können. Somit hat der WLC ausreichend Informationen um den neuen PTK ableiten zu können und antwortet mit einem Reassociation Response. Der Client kann somit unmittelbar mit der Datenübertragung fortfahren.

Beim CCKM handelt es sich um eine Lösung, die eine gute Roaming-Performance erzielt, jedoch ist es ein proprietäres Protokoll, das somit auch nur auf Geräten der Marke Cisco funktioniert. Darüber hinaus ist, wie zuvor beschrieben, ein WLC notwendig. Dabei handelt es sich um Enterprise Geräte, welche im Verbund mit Lizenzen für mehrere APs preislich sehr hoch angesiedelt sind.

2.2.6 802.11r - Fast Basic Service Set Transition

Mit dem den Standard 802.11i kam es zu einem starken Anstieg der Komplexität beim Aufbau einer WLAN-Verbindung. Damit nahm auch die Dauer des Herstellungsprozesses einer Verbindung wesentlich zu. Um dieser Entwicklung gegenzusteuern wurde 2008 die Erweiterung 802.11r des 802.11 Standards eingeführt [MW12]. Dies sollte wiederum eine Minimierung der Komplexität bringen.

Der grundlegende Gedanke der Erweiterung 802.11r ist jener, den AP, zu welchem gewechselt werden soll, auf den Roaming-Vorgang vorzubereiten. Dies geschieht prinzipiell durch das Verteilen des PMK an die APs. Dies wird auch als Pre-Authentication bezeichnet. Dadurch ist es möglich, einerseits die Authentifizierung und andererseits den 4-Way-Handshake zu überspringen.

Für die Pre-Authentication am Ziel-AP gibt es zwei Möglichkeiten, Over-the-Air und Over-the-Distribution-System. Beide Varianten können jeweils mit und ohne Ressourcenreservierung durchgeführt werden, wobei die Unterstützung des Standards 802.11k für Ressourcenreservierung Voraussetzung ist [TTHJ09], [Soc08a]. Bei der Over-the-Air-Variante führt der Client die Pre-Authentication selbst, direkt am Ziel-AP, durch. Der Ablauf eines Roaming-Vorgangs unter der

Verwendung der Over-the-Air-Variante ist in Abbildung 2.14 zu sehen. Der Client sendet, nachdem ein Ziel-AP gefunden wurde, den Authentication Request direkt an den neuen AP. Dieser antwortet mit einem Authentication Response. Danach erfolgt die Umschaltung der Verbindung mittels der Reassociation-Nachrichten. War die Umschaltung der Verbindung erfolgreich, können wieder Daten gesichert über die neue Verbindung übertragen werden.

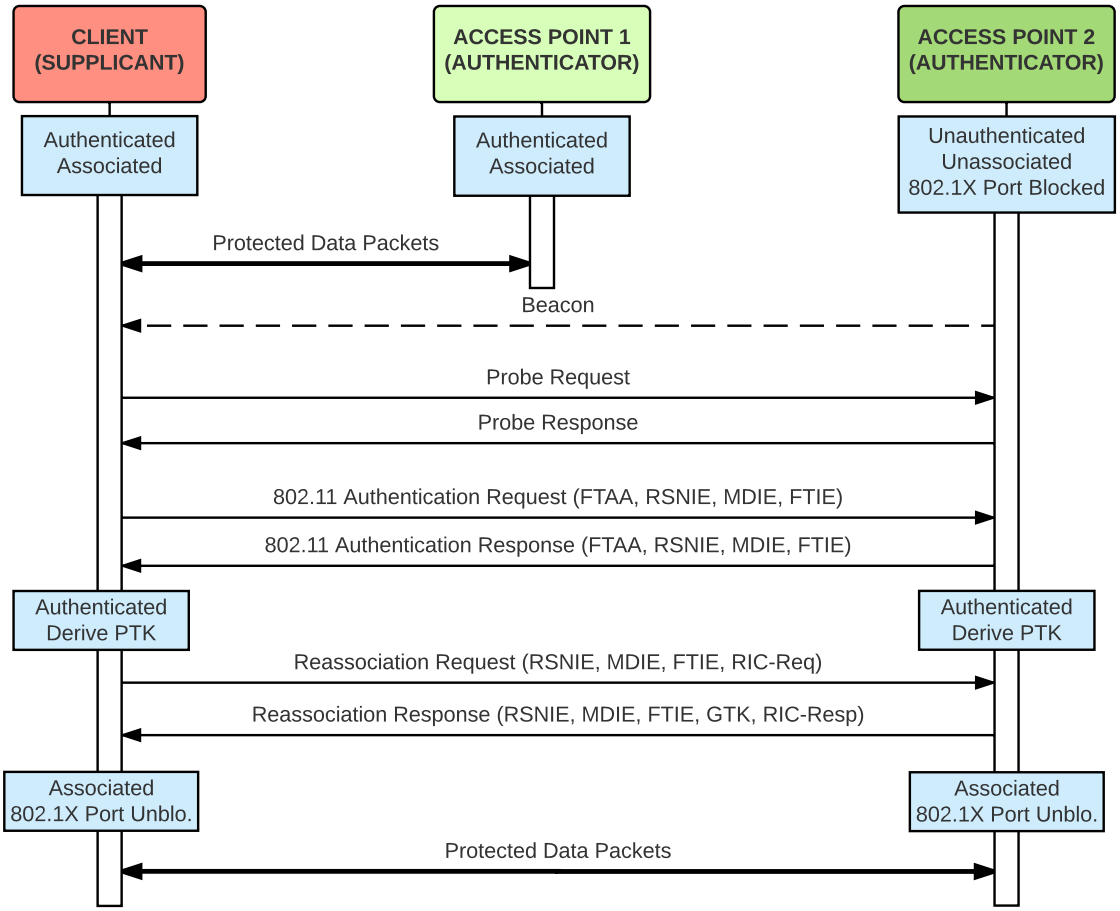

Abbildung 2.14: Ablauf der Verbindungsumschaltung unter Verwendung des Standards IEEE 802.11r und der Variante Over-the-Air [Soc08a]

Bei der Over-the-DS-Variante sendet der Client die, zur Pre-Authentication notwendigen Frames an den AP, zu dem er verbunden ist. Dieser leitet die Daten an den Ziel-AP weiter. Diese Variante ist der Abbildung 2.15 zu entnehmen. Die Reassociation Request Nachricht wird schließlich wieder direkt zum Ziel-AP gesendet.

Pre-Authentication funktioniert innerhalb einer Mobility Domain, das heißt in einem Verbund mehrerer APs innerhalb eines ESS [Nag]. Dazu müssen die APs in der Lage sein, PMKs innerhalb des Netzwerks zu verteilen, sowie Pre-Authentication durchführen zu können. Die Kommunikation zwischen den APs ist jedoch nicht Teil des Standards 802.11r. Des Weiteren ist zu beachten, dass um die Variante Over-the-DS nutzen zu können, der Client vor dem Start des Ablaufs bereits einen Ziel-AP ausfindig gemacht haben muss. Die Information über den Ziel-AP wird auch in der FT Request Nachricht, wie in der Abbildung ersichtlich, angegeben. Erst mit dieser Information ist

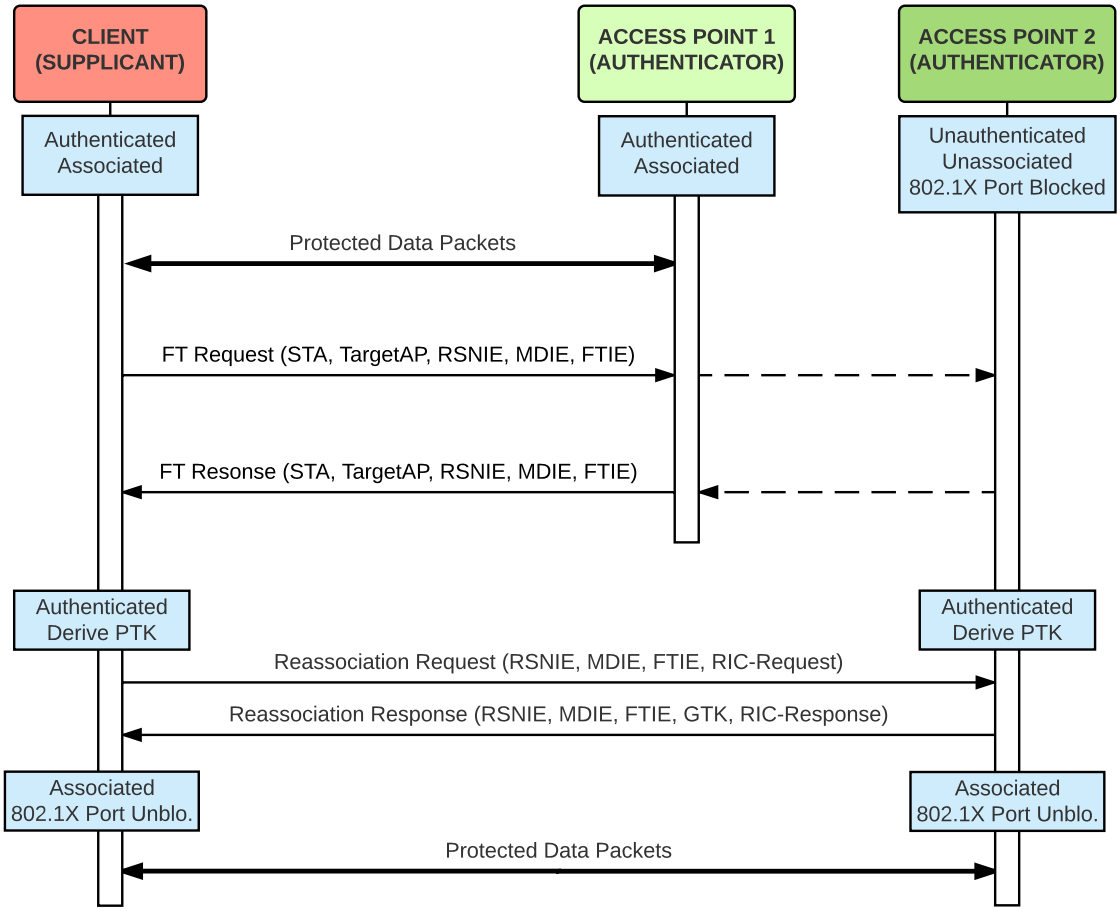

Abbildung 2.15: Ablauf der Verbindungsumschaltung unter Verwendung des Standards IEEE 802.11r und der Variante Over-the-Distribution-System [Soc08a]

es dem ersten AP möglich, die Nachricht an den richtigen AP weiterzuleiten. Es sei angemerkt, dass der 802.11r Standard sowohl WPA2-Enterprise (802.1X) als auch WPA2-Personal (PSK) unterstützt [Nag].

Der detaillierte Ablauf eines 802.11r Roamings ist in der Abbildung 2.16 und 2.17 ersichtlich und wird in weiterer Folge genauer beschrieben.

1. Bei der initialen Verbindung eines Clients zum Netzwerk tauscht der Client die Probe-, Authentication- und Association-Nachrichten mit dem AP aus um sich auf die Parameter der Datenübertragung und die Authentifizierung zu einigen.

2. Die 802.1X Authentifizierung erfolgt am RADIUS-Server. Dabei wird ein MSK generiert, von dem in weiterer Folge der PMK abgeleitet wird. Der PMK, welcher vom initialen AP, zu dem sich der Client verbunden hat, angefordert wurde, wird als PMK-R0 bezeichnet. Aus dem PMK-R0 wird der PMK-R1 abgeleitet. Sowohl PMK-R0 als auch PMK-R1 sind für jeden Client einzigartig innerhalb des Netzwerks.

3. Der PMK-R1 wird im nächsten Schritt an alle APs in der Mobility Domain verteilt. Wie

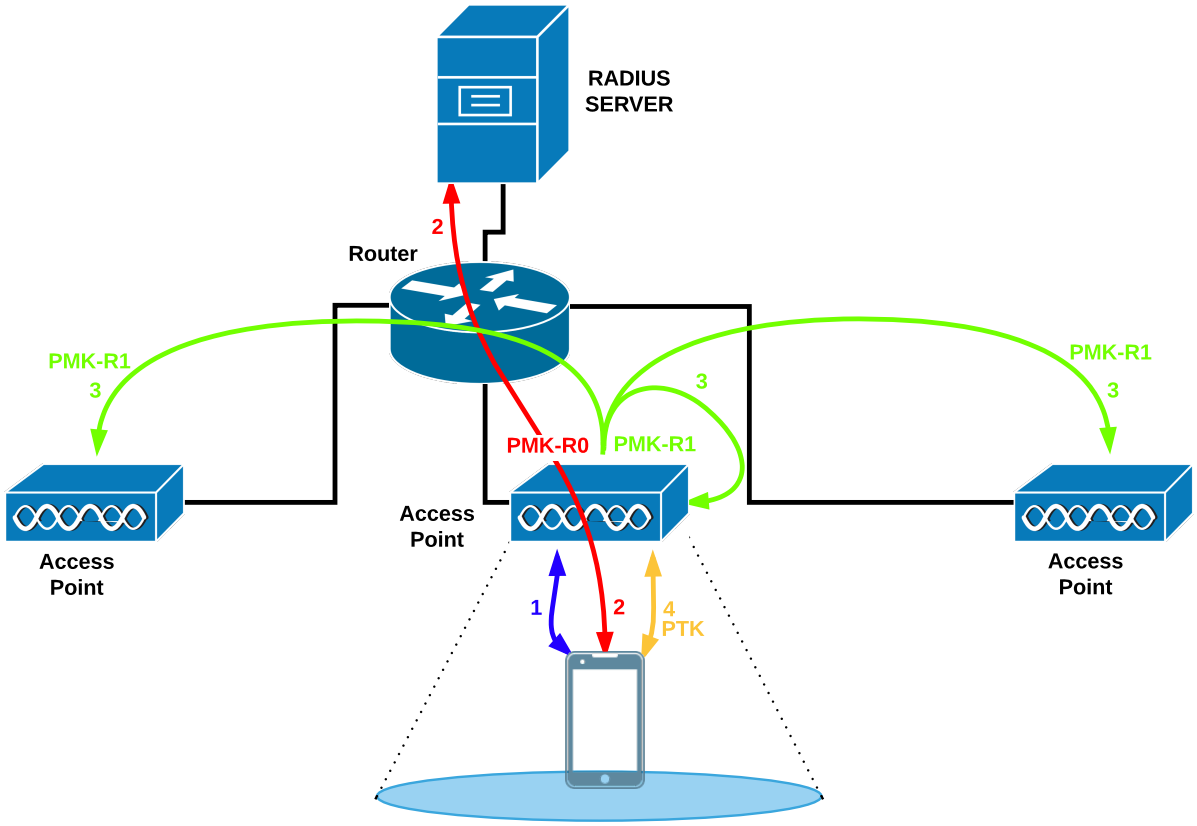

Abbildung 2.16: IEEE 802.11r - Fast Transition: Initialer Verbindungsaufbau und Authentifizierung am RADIUS-Server [Nag]

zuvor schon angemerkt wurde, ist die Verteilung der Keys an die APs nicht im Standard festgelegt.

4. Die Ableitung des Pairwise Transient Keys wird mittels eines 4-Way-Handshakes zwischen dem Client und dem AP generiert. Der PTK wird für die Datenverschlüsselung verwendet. Danach ist der Client grundsätzlich mit dem Netzwerk verbunden und kann Daten übertragen.

5. Beim eigentlichen Roaming gibt der Client dem neuen AP im 802.11 Authentication und Reassociation Request bereits einen gültigen R1KH-ID (R1 Key Holder Identifier, so wird der PMK-R1 auch bezeichnet) und eine PMKID bekannt. Akzeptiert der neue AP die Verbindungsanfrage, so können sowohl die 802.1X Authentifizierung als auch der 4-Way-Handshake ausgelassen werden. Aus diesem Grund ist Roaming nach dem 802.11r Standard auch schneller als Pre-Shared Key (PSK) Roaming, da der 4-Way-Handshake aufgrund der Bereitstellung der Daten (ANonce, SNonce, MIC, GTK), welche zur Ableitung des PTK gebraucht werden, eingespart werden. Diese Daten werden im 802.11r Standard festgelegten Fast Transition Information Element (FTIE) innerhalb der Authentifizierung und Reassoziation Nachrichten übertragen. Das bedeutet wiederum, dass nur 4 Nachrichten übertragen werden müssen, um die Datenverbindung wiederherstellen zu können. Dieser Ablauf ist auch in Abbildung 2.14 zu sehen. Es reicht nach dem Austausch der Authentifizierungsnach-

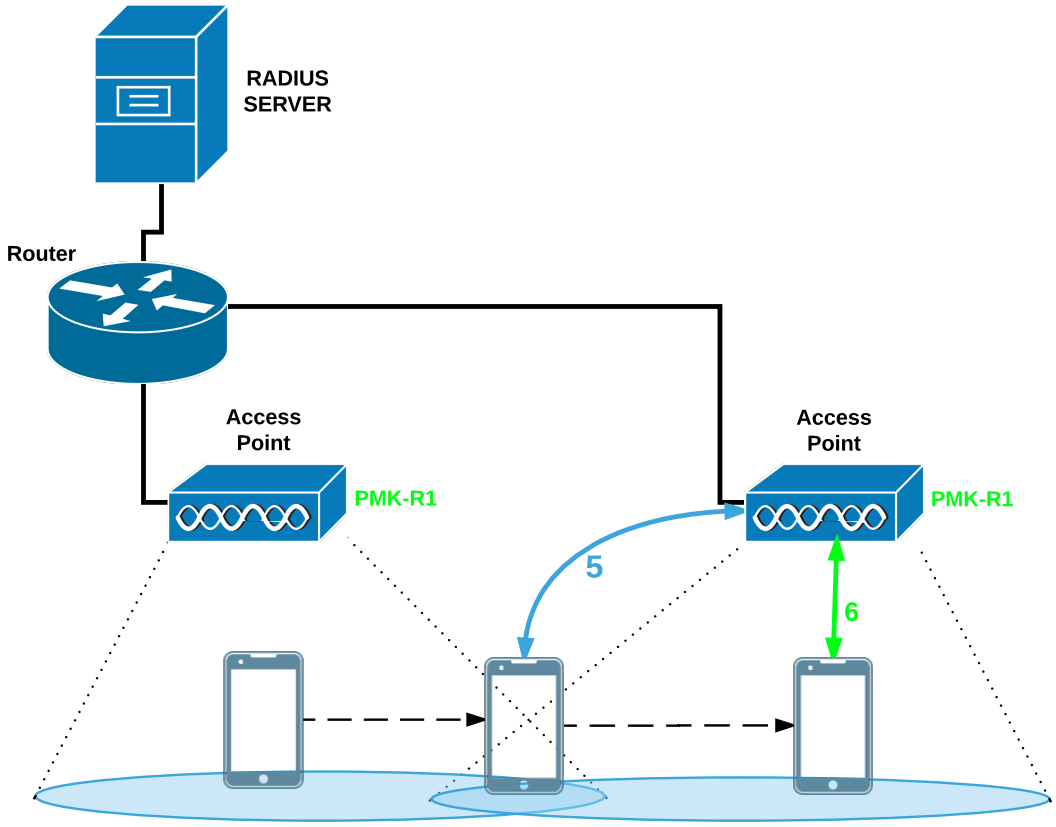

Abbildung 2.17: IEEE 802.11r - Fast Transition: Wechsel der Verbindung [Nag]

richten die Übertragung des Reassoziationsnachricht. Nach diesen vier Nachrichten ist die Verbindung zum neuen AP hergestellt und die Datenübertragung kann fortgeführt werden.

6. Die Verbindung zum neuen AP wurde somit hergestellt und es können wieder Daten über das Netzwerk übertragen werden.

Im Standard IEEE 802.11r wurde im Gegensatz zum ursprünglichen 802.11 Standard einige Änderungen und Erweiterungen der Nachrichten vorgenommen. Dabei wurden zwei Elemente neu in den Standard aufgenommen und ein Element wurde geändert. Details sind der nachfolgenden Aufzählung zu entnehmen [Nag]:

- Mobility Domain Information Element (MDIE) - NEU
- Fast Transition Information Element (FTIE) - NEU
- Robust Security Network (RSN) Information Element - GEÄNDERT

Das Mobility Domain Information Element gibt eine Liste an APs an, welche in der Mobility Domain den Standard 802.11r unterstützen. Das MDIE ist enthalten in Beacon Messages, Probe Responses, Authentication Requests/Responses, Association Requests/Responses und Reassociation Requests/Responses. Eine Beispiel eines MDIE ist ist in Abbildung 2.18 ersichtlich.

Das MDIE beinhaltet die Mobility Domain ID (MDID), welche eine Mobility Domain eindeutig festlegt. Anhand der MDID kann ein Client herausfinden, ob ein zukünftiger AP zur Mobility Domain gehört und somit 802.11r unterstützt. Darüber hinaus enthält das MDIE weitere Informationen, etwa ob das Fast Transition Resource Request Protocol unterstützt wird, welche eine Ressourcenreservierungsanfrage an einen AP erlaubt, bevor zu diesem gewechselt wird. Ebenfalls ist ersichtlich, ob eine AP die Möglichkeit von Fast Transition over the Distribution System erlaubt.

Wie zuvor schon beschrieben, enthält das FTIE Informationen, die für die Authentifizierung benötigt werden. Dazu gehören ANonce, SNonce und MIC, welche zum Ableiten des PTK gebraucht werden. Ein Beispiel ist in Abbildung 2.19 ersichtlich. Das FTIE in den Authentication und Reassociation Request/Responses enthalten. Im RSN IE werden Authentifizierungs-, Key-Management und Verschlüsselungsmethoden festgelegt. Das RSN IE wird um die beiden Methoden Fast Transition Authentication and key Management using IEEE 802.1X und Fast Transition Authentication and key Management using Pre-Shared Key erweitert. Ein Beispiel eines RSN IE ist in 2.20 abgebildet.

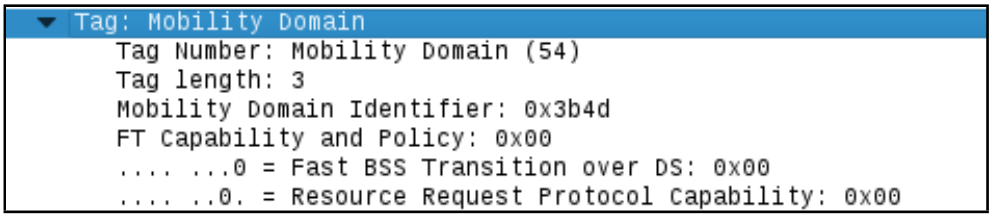

Abbildung 2.18: Beispiel eines 802.11r Mobility Domain Information Elements

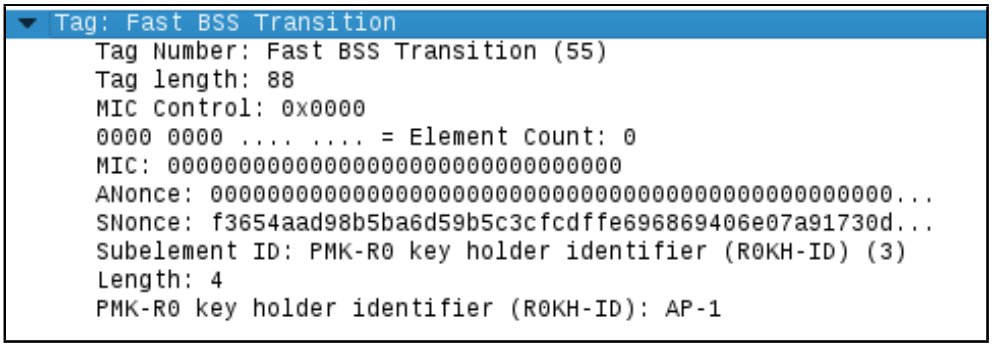

Abbildung 2.19: Beispiel eines 802.11r Fast Transition Information Elements

In der Beispielimplementierung des Standards 802.11r wurde eine Handover-Performance zwischen 27 ms und 30 ms erzielt [TTHJ09]. Ein Roaming-Ablauf bestehend aus einem Authenfifizierungs-Nachrichtenpaar und einem Reassoziierungs-Nachrichtenpaar ist in Abbildung 2.21 zu sehen. Eine Messreihe zur Ermittlung der Dauer des IEEE 802.11r Standards ergab eine durchschnittliche Dauer von 28 ms. Die relative Häufigkeit der Messung ist im Diagramm 2.22 zu sehen. Es ist zu betonen, dass die Dauer der reinen Verbindungsumschaltung somit deutlich unter den geforderten 100 ms liegt.

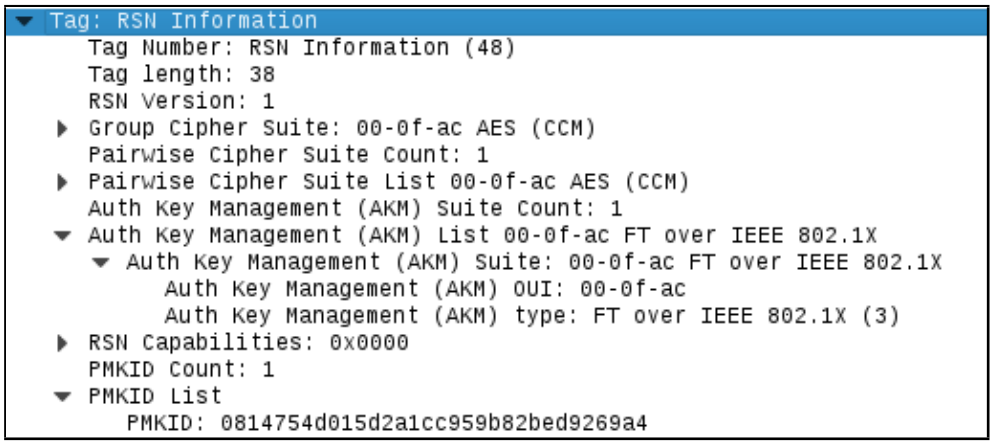

Abbildung 2.20: Beispiel eines 802.11r Robust Security Network (RSN) Information Element

No.	Time	Source	Destination	Info	Protocol
14760	55.5900…	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Action, SN=294, FN=0, …	802.11
14781	*REF*	IntelCor_c8:00:e4	Aerohive_5a:66:d9	Authentication, SN=224…	802.11
14784	0.007645	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Authentication, SN=256…	802.11
14785	0.014811	IntelCor_c8:00:e4	Aerohive_5a:66:d9	Reassociation Request,…	802.11
14788	0.020308	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Reassociation Response…	802.11
14790	0.037431	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Action, SN=259, FN=0, …	802.11

Abbildung 2.21: Aufzeichnung eines Roaming-Vorgangs unter der Verwendung von 802.11r

2.3 Zusammenfassung und Bewertung

Die detaillierte Analyse des Standes der Technik zeigte einerseits, dass Konzepte für nahtlose Roaming-Vorgänge vorhanden sind, jedoch dass aufgrund diverser Einschränkungen aktuell noch keine entsprechende Gesamtlösung verfügbar ist. Im folgenden Abschnitt werden diese Themen noch einmal kurz zusammengefasst und die aufgetretenen Problematiken aufgezeigt. Die Gliederung dabei erfolgt wieder in die zwei Phasen, Entdeckungs- und Authentifizierungsphase.

In der Entdeckungsphase muss entschieden werden, ob ein Verbindungswechsel eingeleitet wird. Ebenfalls ist es die Aufgabe dieser Phase dafür zu sorgen, sich über mögliche Roaming-Ziele zu informieren. Es wurde festgestellt, dass die Aggregation von Information bezüglich der funktechnischen Umgebung ein ausschlaggebender Punkt ist, um einen nahtlosen Raoming-Vorgang zu ermöglichen. Das bedeutet, dass sich ein Client laufend um möglich Roaming-Ziele umsehen muss, um im Falle eines Verbindungseinbruches einen Wechsel zu einem anderen AP zu ermöglichen, ohne dass laufende Applikationen unterbrochen werden. Für die Suche nach Roaming-Zielen bestehen grundsätzlich zwei Möglichkeiten. Der Client kann durch aktives Scannen der WLAN-Kanäle nach möglichen APs suchen. Um jedoch auf anderen Kanälen nach APs suchen zu können, muss die laufende Verbindung unterbrochen werden. Dies bedeutet wiederum, dass in der Zeit des Scans keine Daten übertragen werden können. Ein weiteres Problem ist die zeitliche Dauer eines Scan-Vorganges. Die große Zahl an Kanälen in Kombination mit der Zeit, die auf jedem Kanal nach APs gesucht werden muss, macht einen Einsatz dieses Verfahrens praktisch unmöglich. Darüber hinaus ist dieses Verfahren schlichtweg zu langsam um die Dynamik des Netzwerkes, gerade bei Betrachtung eine bewegten Clients, im Auge zu behalten.

Die zweite Möglichkeit Informationen über benachbarte APs zu erhalten ist jene, Information

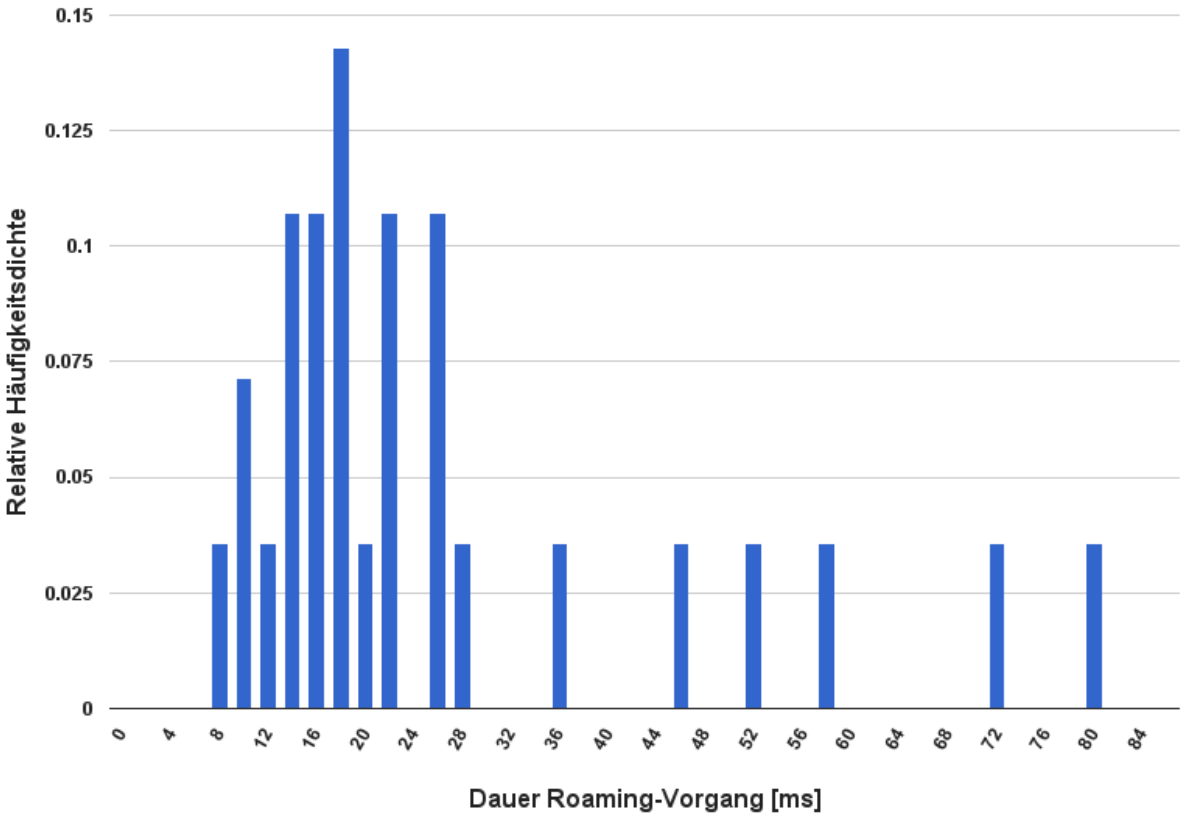

Abbildung 2.22: Relative Häufigkeitsdichte der Dauer eines Roaming-Vorgangs mit dem Standard IEEE 802.11r

von anderen Stationen abzufragen. Der klare Vorteil dieses Verfahrens liegt darin, dass dabei der aktuelle Sendekanal nicht verlassen werden muss. Das bedeutet, dass die aktuelle Verbindung dazu genutzt wird, beispielsweise vom AP abzufragen, ob es weitere APs im Netzwerkverbund gibt und wie diese erreichbar sind. Ein Scan der Kanäle ist dabei nicht nötig. Darüber hinaus ist die Beeinflussung des Datenverkehrs durch das Senden einer Request-Nachricht an und des Empfangs einer Response-Nachricht vom AP minimal. Die Erweiterung IEEE 802.11k des WLAN-Standards IEEE 802.11 bietet unter anderem die Möglichkeit Nachrichten in dem eben besprochenen Format zu übermitteln. Besonders zu betonen ist in diesem Zusammenhang, dass es sich um ein standardisiertes Verfahren handelt. Dies ist für eine herstellerübergreifende Nutzung grundlegend. Genau dieser Aspekt schließt auch die Verwendung vieler proprietärer Verfahren aus.

Die Authentifizierungsphase ist der zweite Abschnitt beim Aufbau einer Verbindung zu einem WLAN-Netzwerk. Es handelt sich dabei um jene Phase, in der seitens des Netzwerks überprüft wird, ob einem Client Zugang zum Netzwerk erteilt wird. Es wurden verschiedene Verfahren vorgestellt und deren Verwendbarkeit für einen nahtlosen Roaming-Vorgang untersucht. Der Fokus der Betrachtung lag dabei auf mehreren Aspekten. Es galt ein Verfahren zu finden, das mit dem heutigen Stand der Technik als sicher und verwaltbar gelten. Des Weiteren wird vom Verfahren eine nahtlose Verbindungsumschaltung zwischen APs gefordert. Dies bedeutet, dass ein Roaming-Ablauf für höher liegende Schichten, wie beispielsweise Applikationen, nicht merkbar ablaufen muss. Dies impliziert ebenfalls, dass sich auch die IP-Adresse bei der Umschaltung nicht

ändert. Als letzte Anforderung galt es die Verbindung in unter 100 ms von einem AP zu einem anderen zu wechseln. Da diese zeitliche Vorgabe jedoch den gesamten Roaming-Prozess betrifft, also auch die Suche nach einem Ziel-AP, sollte die Dauer der Authentifizierung im Grunde so kurz wie möglich sein.

Es erfolgte eine Untersuchung von verschiedene Authentifizierungsverfahren. Dabei zeigte sich, dass die Mehrheit dieser Verfahren aus drei Gründen nicht für einen nahtlosen Roaming-Vorgang geeignet sind. Erster Grund dabei ist das Sicherheitsrisiko, das verschiedene Verfahren mit sich bringen. In einem drahtlosen Netzwerk im industriellen Umfeld ist der Einsatz eines möglichst sicheres Verfahren Grundvoraussetzung. Ein weiteres Problem stellt die unzureichende Verwaltungsmöglichkeit dar. Diese Einschränkung wird umso eklatanter, je mehr Clients sich in einem Netzwerk befinden. Als letzte Einschränkung wurde die Dauer eines Umschaltvorgangs identifiziert. Diese macht ein nahtlosen Wechsel der Funkverbindung ebenfalls unmöglich.

Lediglich die Erweiterung IEEE 802.11r des IEEE 802.11 Standards konnte die Anforderungen eines nahtlosen Roaming-Vorgangs bereitstellen. Dabei ist anzumerken, dass dieses Protokoll nicht alleine eingesetzt werden kann. Viel mehr muss es in Kombination mit einem Standardverfahren wie WPA2-PSK oder WPA2-Enterprise verwendet werden. Dies bedeutet, dass ein initialer Verbindungsaufbau und die Authentifizierung nach einem der eben erwähnten Standards erfolgt. Die Erweiterung IEEE 802.11r bietet jedoch in weiterer Folge die Möglichkeit die Dauer von Umschaltvorgängen durch Vorauthentifizierung eklatant zu verringern.

Im Diagramm 2.23 ist dazu ein Vergleich zwischen den drei Authentifizierungsverfahren WPA2-PSK, WPA2-Enterprise und IEEE 802.11r dargestellt. Es ist zu beachten, dass die Gegenüberstellung nur die Dauer der Authentifizierung aufzeigt, ein möglicher Scan der WLAN-Kanäle ist darin nicht berücksichtigt. Offensichtlich dauert einee Authentifizierung nach dem Standard WPA2-Enterprise um ein Vielfaches länger als jene nach den Standards WPA2-PSK oder IEEE 802.11r. Obwohl die Messung der Variante WPA2-PSK noch kürzere Zeiten als jene unter Nutzung des Protokolls IEEE 802.11r, ist diese aus aufgrund von Sicherheitsrisiken nicht einsetzbar.

Abschließend scheint, dass eine Kombination der beiden Standards IEEE 802.11k und IEEE 802.11r Grundbausteine für eine Optimierung von nahtlosem Roaming sein können. Dabei steigert der Standard IEEE 802.11k die Performance der Entdeckungsphase und hilft dem Client, mögliche Roaming-Stationen zu finden. Der Standard 802.11r wiederum sorgt für eine sichere Verbindung durch die Verwendung des Standards IEEE 802.1X und ist darüber hinaus auch für eine Netzwerkinfrastruktur mit vielen Clients einsetzbar. Trotz der Verwendung eines sicheren Verfahrens sind somit schnelle Roaming-Abläufe möglich. Der Standard IEEE 802.11r liefert also eine Kombination aus schnellen und sicheren Authentifizierungsverfahren.

Ein grundlegendes Problem für nahtlose Roaming-Vorgänge wird in keinem der vorgestellten Verfahren behandelt wird. Es geht dabei um die Entscheidung, wann ein Roaming-Prozess ausgelöst werden soll. Das heißt, auf Grundlage welcher Informationen ein Client entscheidet, ob nun ein Wechsel zu einem anderen AP durchgeführt wird. Diese Frage bleibt in all den untersuchten Verfahren unbeantwortet, ist jedoch von entscheidender Bedeutung. Die Erweiterung IEEE 802.11k bietet zwar die Möglichkeit Informationen für Roaming-Entscheidungen zu erlangen, die endgültige Entscheidung wird jedoch nicht getroffen. Dies bedeutet, dass in weiterer Folge genau diese Thematik noch genauer bearbeitet werden muss. Die Herausforderung dabei ist die Auswahl einer Entscheidungsgrundlage, als die Festlegung eines Parameters aufgrund dessen ein Roaming-Vorgang durchgeführt wird. Mögliche Ansätze wären beispielsweise die Empfangsfeldstärke eines Clients, die Auslastung der APs oder die Verfügbarkeit von Nachbar-APs. Diese und noch weitere Aspekte könnten als Grundlage zur Entscheidung dienen.

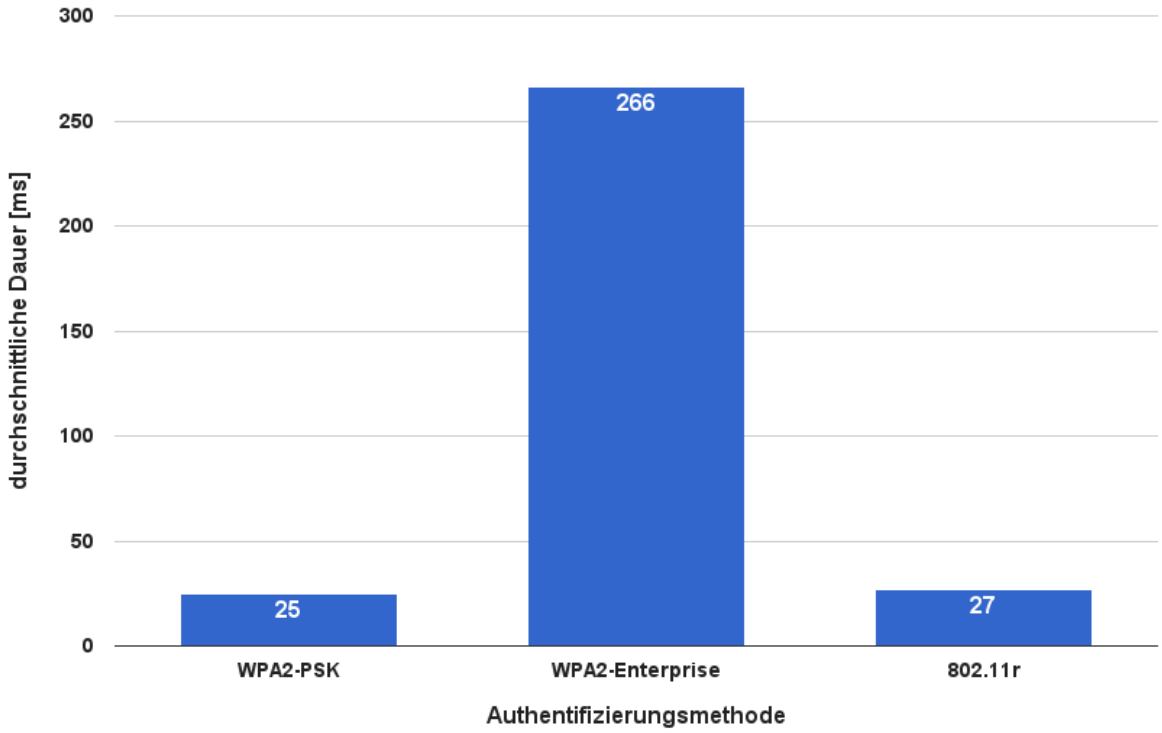

Abbildung 2.23: Gegenüberstellung der Dauer verschiedener Authentifizierungsverfahren

3 Konzept und Realisierung

Aufgrund der in Kapitel 2 gewonnen Erkenntnisse wird in weiterer Folge ein Konzept entwickelt, wie Roaming beschleunigt werden kann. Es soll sich dabei um eine Methode handeln, die die beiden grundlegenden Anforderungen unterstützt. Dies ist die zeitliche Vorgabe eine Verbindungsumschaltung innerhalb von 100 ms durchzuführen. Wie in Kapitel 1.2 beschrieben, resultiert die Anforderung aus der Vorgabe der ITU-T, dass die Gesamtverzögerungszeit eines Audiosignals bei Voice over IP maximal 150 ms betragen darf [IT03]. Da sich diese Verzögerung auf die gesamte Übertragungsstrecke, also alle dabei beteiligten Soft- und Hardwarekomponenten bezieht, wird die Anforderung an die Verzögerung, die durch einen Roaming-Ablauf entstehen kann noch darunter festgelegt.

Die zweite Anforderung ist die Gewährleistung einer nahtlosen Umschaltung, die für höhere Schichten transparent abläuft. Über die Verbindung laufende Applikationen dürfen durch den Umschaltvorgang nicht unterbrochen werden. Dazu gehört beispielsweise auch, dass sich die IP-Adresse bei der Umschaltung nicht ändert. Diese Anforderung impliziert, dass die Netzwerkinfrastruktur in der Lage sein muss, diese Umschaltung vorzunehmen. Das heißt, dass die Pakete nach einem erfolgten Roaming-Ablauf auch weiterhin an den korrekten Client zugestellt werden.

Im folgenden Kapitel findet die Ausarbeitung eines Algorithmus statt, der nahtlose Roaming-Vorgänge ermöglicht. Dies wird durch die Optimierung beider Phasen, also der Entdeckungs- auch der Authentifizierungsphase erreicht. In weiterer Folge wird die Implementierung des Roaming-Verfahrens erläutert. Ebenfalls ist der Aufbau eines Testsystems zur Messung der Roaming-Performance in diesem Abschnitt beschrieben. Dazu ist es nötig eine kurze Einführung in die WLAN-Architektur in linuxbasierten Systemen zu geben. Dies soll auch in diesem Abschnitt geschehen. Da sich die Findung von passender Hardware als komplexe Aufgabe erwies, soll auch dieser Thematik ein Unterkapitel gewidmet werden. Bei der Hardware-Auswahl galt es einerseits passende APs und andererseits einen Client mit der notwendigen Netzwerkschnittstelle und unterstützten Standards zu finden.

3.1 Konzept des Roaming-Algorithmus

Kapitel 2.1 hat die Phasen des Scan-Prozesses und der Authentifizierung als jene Abläufe identifiziert, die einen nahtlosen Roaming-Vorgang verhindern. Bei der Messung verschiedener Verfahren dauerte der initiale Scan-Prozess, also jener beim erstmaligen Suchen und Verbinden mit einem neuen Netzwerk, in der Größenordnung von 2.5 s (Details siehe Kapitel 2 bzw. Abb. 2.2, 2.4 und 2.6). Ebenfalls ist in den Abbildungen zu sehen, dass die Authentifizierungsphase, je nach verwendeter Authentifizierungsmethode, weitere Zeit in Anspruch nimmt. Beim Einsatz von WPA2-Enterprise beispielsweise schlägt sich die Authentifizierung im Mittel mit 266 ms zu Buche (siehe Abb. 2.4). Sogar bei einem Verbindungsaufbau zu einem ungesicherten Netzwerk dauert das Senden der Authentifizierungsnachrichten 86 ms (siehe Abb. 2.1). Zu den Messungen ist anzumerken, dass es sich dabei jeweils um den initialen Verbindungsaufbau handelt und nicht um Roaming-Szenarien, wo eine Verbindung von einem AP zu einem anderen umgeschaltet wird. Jedoch soll damit verdeutlicht werden, wie lange derartige Prozesse dauern können und wie weit man hier von einer nahtlosen Umschaltung und der geforderten Dauer von 100 ms entfernt ist.

Um ein Roaming innerhalb von 100 ms erreichen zu können, ist offensichtlich, dass eine Verbesserung sowohl der Entdeckungsphase als auch der Authentifizierungsphase nötig ist. Dies bedeutet, dass eine Lösung, im Wesentlichen bestehend aus zwei Teilen entwickelt werden muss. Der erste Teil der Lösung zielt auf eine Verkürzung des Scan-Prozesses und der zweite Teil auf eine Beschleunigung des Authentifizierungsprozesses ab. Wie auch schon in Kapitel 2 ausgearbeitet wurde, kann ein Roaming-Vorgang noch weiter in die folgenden Teilaufgaben zerlegt werden. Dabei sind die ersten drei Punkte in der Entdeckungsphase und der letzte Punkt in der Authentifizierungsphase angesiedelt:

1. Finden von möglichen Roaming-Zielen (Entdeckungsphase)
2. Überwachung des Status der aktuellen Verbindung (Entdeckungsphase)
3. Auslösen des Roaming-Prozesses aufgrund von bestimmten Parametern (Entdeckungsphase)
4. Authentifizierung am neuen AP (Authentifizierungsphase)

Diese vier Punkte sollen nur in einzelnen Abschnitten ausgearbeitet werden, um daraus eine Gesamtlösung für das Roaming-Problem zu erreichen.

3.1.1 Findung von Roaming-Zielen

Die Analyse des Scan-Prozesses zeigte, dass dieser der zeitaufwendigste Teil beim Durchführen eines Roaming-Vorganges ist. Dies legt nahe, dass ein Scan der Kanäle im optimalen Fall vermieden werden sollte. Ist eine komplette Vermeidung nicht machbar, dann sollte die Suche nach APs zumindest zu Zeitpunkten stattfinden, in denen die Verbindung durch den Scan nicht unterbrochen werden muss. Das bedeutet, es sollte ein Scan nicht erst dann durchgeführt werden, wenn die Verbindung schon abgebrochen ist. Die Vermeidung von Scans sollte jedoch eine hohe Priorität haben.

Wie im Abschnitt 2.1.2 ausgearbeitet wurde, gibt es neben dem Scan eine weitere Möglichkeit Informationen über benachbarte APs zu generieren. Es handelt sich dabei um die Abfrage der Information bei anderen Knoten im Netzwerk. Deshalb wird der erste Teil der Lösung zum Ziel haben,

den Scan-Prozess so gut wie möglich zu verhindern. Sollte doch ein Scan notwendig sein, ist dieser so zu takten, dass eine bestehende Verbindung nur möglichst gering beeinflusst wird. Darüber hinaus gilt es den zeitaufwändigen Scan aller WLAN-Kanäle durch eine laufende Überwachung der Verbindungsparameter zu verhindern.

Die zugrunde liegende Idee dabei ist, dass sich der Client schon vor einem möglichen Verbindungsverlust um alternative APs umschaut um im Falle eines Verbindungsabbruchs bereits einen möglichen AP, zu dem die Verbindung gewechselt werden kann, parat zu haben. Um eine noch bessere Performance bzw. Anbindung an das Netzwerk zu erreichen, kann auch schon vor dem Verbindungsverlust zu einem anderen AP gewechselt werden. Dazu informiert sich der Client regelmäßig bei jenem AP, zu dem gerade eine aufrechte Verbindung besteht, über mögliche Roaming-Ziele. Dafür wird eine Funktion aus dem Standard IEEE 802.11k (Radio Resource Management) genutzt. Der Standard stellt dafür den sogenannten Neighbor Report Request bereit. Mit dieser Anfrage vom Client an den AP ist es möglich Informationen über benachbarte APs zu bekommen, ohne die aufrechte Verbindung unterbrechen oder auf einen anderen Kanal wechseln zu müssen. Nähere Informationen zum Neighbor Report Request sind dem Kapitel 2.1.3 zu entnehmen. In der Antwort vom AP auf den Neighbor Report Request, dem sogenannten Neighbor Report Response, werden benachbarte APs, die demselben Extended Service Set, also dem gleichen Netzwerkverbund angehören, zurückgemeldet. Des Weiteren werden im Neighbor Report Response auch die SSIDs der APs und deren Sendekanäle bekanntgegeben. Damit kann auf den Scan-Prozess zur Findung möglicher Roaming-Ziele verzichtet werden. Mit den aus dem Neighbor Report Response gewonnenen Informationen kann sich der Client ein Bild über seine funktechnische Umgebung machen. Somit hat der Client die Möglichkeit sich effizient einen Überblick über mögliche APs und deren Verbindungsqualität zu verschaffen. Ein Scan der Kanäle, um mögliche Roaming-Stationen zu finden, ist somit nicht mehr nötig.

Es wurde nun festgelegt, dass sich der Client unter Verwendung des Standards IEEE 802.11k und der darin enthaltenen Funktion des Neighbor Report Request über mögliche Roaming-Ziele informiert. In weiterer Folge stellt sich die Frage, wann und wie oft eine derartige Abfrage, ob benachbarte APs verfügbar sind, passieren soll.

Um ein eine optimale Parametrisierung des Intervalls der Neighbor Report Request zu erreichen, wird einleitend erörtert, was die Folgen von zu wenigen oder zu vielen Anfragen an den AP sein könnten. Wird das Intervall zu klein gewählt, werden unnötig oft Neighbor Report Requests an den AP geschickt. Dies mag bei einem oder wenigen Clients in einem Netzwerk noch kein Problem darstellen, jedoch skaliert sich diese Thematik mit steigender Anzahl an Knoten im Netzwerk und die Belastung des APs mit Neighbor Report Anfragen steigt mit jedem Client im Netzwerk. Auf der anderen Seite kann ein zu groß gewähltes Intervall dazu führen, dass der Client der Dynamik des Netzwerkes nicht folgen kann. Dies bedeutet, dass sich die Empfangssignale der APs so schnell ändern, dass der Client davon nichts mitbekommt. Daraus folgt, dass der Client recht bald die Verbindung zum Netzwerk verlieren wird, da er einfach nicht oder nur zu spät realisiert, dass die Empfangsqualität eingebrochen ist und somit auch die Verbindung verloren wurde. Dies hat einerseits zur Folge, dass die Verbindung und auch alle darüber laufenden Applikationen unterbrochen werden. Andererseits folgt daraus, dass erneut ein initialer Verbindungsaufbau nötig ist, der viel Zeit in Anspruch nimmt.

Offensichtlich gibt es aber für den Parameter auch keinen fixen Wert. Dieser hängt stark vom Einsatzszenario und der Dynamik des Netzwerks bzw. wie schnell sich ein Client in diesem bewegt, ab. Die Reichweite von APs hängt wiederum von verschiedenen Parametern wie Sendeleistung

und verwendeter Antennen ab. Im freien Gelände wird eine minimale Reichweite von 30 m angenommen. Eine gehende Person wird mit einer Maximalgeschwindigkeit von 5 km/h, also 1.38 m/s angenommen. Dies bedeutet wiederum, dass eine Person für die 30 m 21.73 s brauchen würde. Befindet man sich in einem Gebäude, wird durch Mauern und andere Störquellen die Reichweite natürlich dementsprechend kleiner. Dies führt zur Annahme, dass die Aussendung eines Neighbor Report Requests im Intervall von 10 s im Büroumfeld ausreichend ist. Für Clients, die sich schneller bewegen, müsste selbstverständlich ein kleiner Wert angenommen werden. Diese Annahmen bringen zwar eine gewisse Einschränkung auf ein ausgewähltes Szenario mit sich, dies wird an dieser Stelle jedoch in Kauf genommen.

Es fand nun weiters eine Festlegung der Frequenz der Aussendung der Neighbor Report Nachrichten statt. Nachfolgend wird diskutiert, welche Parameter zur Überwachung der Verbindung und daraus folgend zum Auslösen eines Roaming-Vorgangs herangezogen werden.

3.1.2 Überwachung der Verbindung

Um eine laufende Verbindung überwachen zu können, muss grundlegend ein Parameter für die Qualität selbiger festgelegt werden. Als einfache und naheliegende Lösung scheint der RSSI-Wert als gut geeignet. Dieser gibt die Empfangsfeldstärke des Signals an der Empfangsantenne an. Darüber hinaus wird dieser beim Empfang von Nachrichten von einem AP regelmäßig aktualisiert und gibt somit zuverlässig Auskunft über die Qualität der Verbindung zu einem AP. Dies bedeutet, dass vor allem die Empfangsqualität des APs, zu dem aktuell eine Verbindung besteht, durchgehend aktualisiert wird.

Um nun aber einen Vergleich zu anderen APs machen zu können, bedarf es auch der Messung der RSSIs zu diesen APs. Grundsätzlich erfolgt eine Erhebung der RSSIs beim Scan der Kanäle. Wie zuvor jedoch schon ausgearbeitet wurde, gilt es diesen aufwendigen und großteils nutzlosen Prozess weitestgehend zu vermeiden. Dazu wird nun wieder die Information, die aus dem Neighbor Report Request gewonnen wurde, genutzt. Dort ist neben der Information der benachbarten APs und deren SSIDs auch angegeben, auf welchen Kanälen diese zu erreichen sind. Um nun die RSSIs zu den benachbarten APs zu messen, reicht also ein selektiver Scan auf den Kanälen auf denen laut dem Neigbor Report Request diese APs auch senden. Alle anderen Kanäle müssen nicht durchsucht werden. Dies wäre nebenbei auch völlig wertlos, da dort ohnehin keine möglichen Roaming-Stationen senden.

Dies hat zur Folge, dass dafür ein Scan zum Herausfinden der Verbindungsqualität zu einem bestimmten AP passieren muss. Der Scan beschränkt sich aber auf jene Kanäle, auf denen der Client aus dem Neighbor Report Response weiß, dass auf diesen Kanälen benachbarte APs senden. Es müssen somit nur einzelne Kanäle, je nachdem wie viele APs im Neighbor Report Response gemeldet werden, gescannt werden. Aus den aufgezeichneten RSSI-Werten der möglichen Ziel-APs muss dann weiter die Entscheidung, zu welchem AP gewechselt wird, getroffen werden. Dazu wird im folgenden Abschnitt darauf eingegangen, bei welchen RSSI-Werten das Auslösen eines Roaming-Vorganges sinnvoll ist.

3.1.3 Auslösen eines Roaming-Prozesses

Im diesem Abschnitt ist die Ausarbeitung, unter welchen Umständen ein Roaming-Prozess ausgelöst werden soll, zu finden. Grundsätzlich wird ein Roaming-Vorgang nur dann ausgelöst, wenn der RSSI-Wert der aktuellen Verbindung unter einen bestimmten Grenzwert fällt. Dies ist gleichbedeutend mit der Abnahme der Verbindungsqualität unter einen bestimmten Grenzwert. Dieser Grenzwert wird auf $-71\,\mathrm{dBm}$ festgelegt. Grund für diese Definition ist, dass die Firma Cisco $-71\,\mathrm{dBm}$ als Voraussetzung einer WLAN-Konfiguration angibt, um eine Datenrate von 54 Mbps gewährleisten zu können [2]. Dies reicht aus um sowohl Sprach- als auch Daten übertragen zu können.

Eine zweite Voraussetzung um einen Roaming-Vorgang auszulösen, ist die Verfügbarkeit eines benachbarten APs, also eines APs zu dem ein Verbindungswechsel stattfinden könnte. Die bloße Verfügbarkeit eines APs ist jedoch noch nicht ausreichend. Dieser muss auch dem gleichen ESS, also dem gleichen Netzwerkverbund angehören, damit ein Roaming-Vorgang überhaupt möglich ist. Durch die Nutzung des Standards 802.11k und des darin enthaltenen Neighbor Report Requests wird sichergestellt, dass in der Liste eines Clients an möglichen Ziel-APs nur APs sind, die auch diese Voraussetzung erfüllen.

Erfüllt nun ein potentieller Ziel-AP alle geforderten Kriterien, so wird laufend der RSSI zu diesem AP gemessen. Bietet der AP nun auch eine bessere Verbindung, als der aktuelle AP, kann ein Roaming ausgelöst werden. Dazu wird eben der RSSI-Wert der aktuellen Verbindung mit der zu einem Nachbar-AP verglichen. Dies erfordert, wie zuvor schon erwähnt, eine laufende Aufzeichnung und Auswertung der RSSI-Werte zu den benachbarten APs.

Um zu verhindern, dass bei ähnlichen RSSI-Werten der aktuellen und der potentiellen, neuen Verbindung ein ständiges Umschalten der Verbindung ausgelöst wird, wird darüber hinaus festgelegt, dass der benachbarte AP einen mindestens um $10\,\mathrm{dBm}$ besseren RSSI-Wert haben muss, damit ein Roaming-Prozess ausgelöst wird.

Dem Flussdiagramm in Abbildung 3.1 ist der entworfene Roaming-Algorithmus zu entnehmen. Im oberen Abschnitt ist das zyklische Senden von Neighbor Report Request, zur Findung von möglichen Roaming-Zielen, ersichtlich. Sobald ein Neighbor Report Response empfangen wurde, findet der Vergleich der RSSI-Werte statt. Der RSSI-Wert der aktuell aufrechten Verbindung wird dabei als RSSI bezeichnet. Des Weiteren bezeichnet RSSI_min den Schwellwert der aktuellen Verbindung. Solange diese Grenze nicht unterschritten wird, wird kein Roaming-Prozess ausgelöst. RSSI_neu bezeichnet die Qualität der Verbindung zu einem möglichen Ziel-AP und Buffer ist jener Wert, um den die Qualität der neuen Verbindung besser sein muss, als die alte, damit ein Roaming-Vorgang ausgelöst wird. Wie ebenfalls im Flussdiagramm ersichtlich, handelt es sich um einen Kreislauf. Sobald ein Roaming durchgeführt wurde, beginnt der Prozess von Neuem. Dies bedeutet, dass es sich um einen ständigen Überwachungs- und Optimierungsprozess handelt.

An dieser Stelle soll ebenfalls noch einmal die Wahl des Schwellwertes der Verbindungsqualität, ab dem ein Roaming-Prozess ausgelöst wird, diskutiert werden. Dieser wurde mit $-71\,\mathrm{dBm}$ angenommen um zumindest eine Verbindung mit einer Datenübertragungsrate von 54 Mbps gewährleisten zu können. Auch dieser Parameter ist keineswegs in Stein gemeißelt. Der Schwellwert sollte mit Bedacht auf die Anwendung, welche über die Verbindung laufen soll, gewählt werden. Handelt es sich um Anwendungen, die entsprechend weniger Datenübertragungsrate benötigen, kann dieser Wert durchaus geringer gewählt werden. Die Wahl eines konkreten Wertes erweist sich somit als äußerst schwierig. Denkbar wäre in diesem Zusammenhang natürlich auch, die Parameter variabel zu machen. Das heißt, dass beispielsweise der Client einen Parameter für die Dynamik des

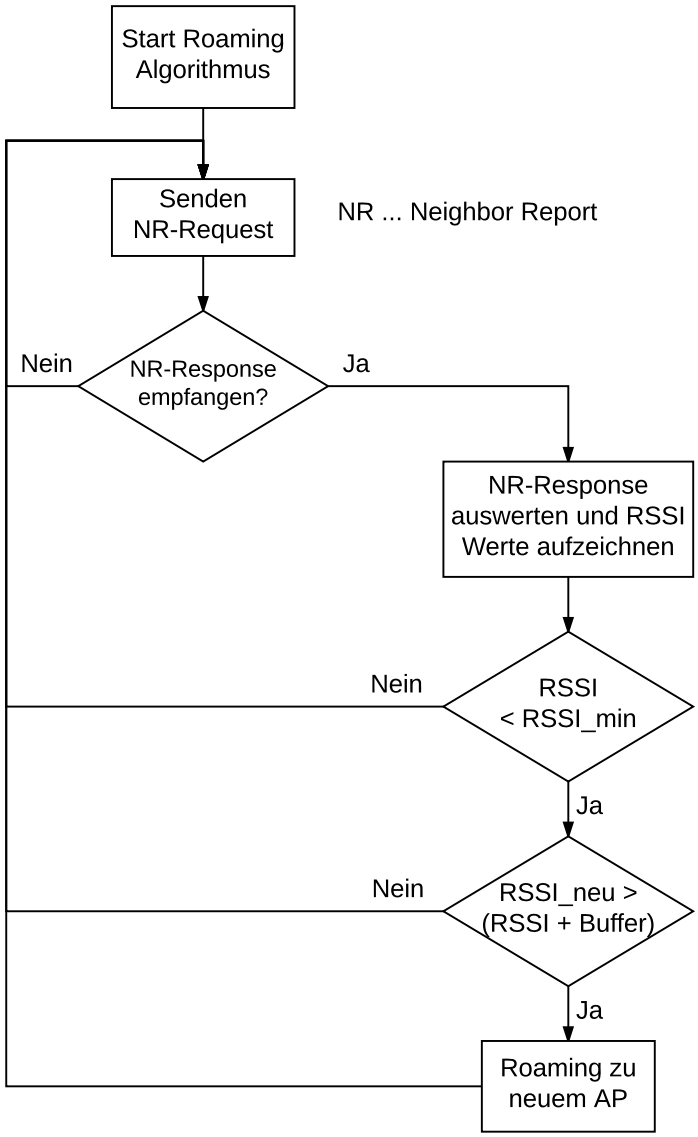

Abbildung 3.1: Flussdiagramm des entwickelten Roaming-Algorithmus auf Basis von Neighbor Report Requests und RSSI-Auswertung

Netzwerkes ermittelt und aufgrund dessen die Parameter entsprechend setzt. Dies würde eine weitergehende Untersuchung der Zusammenhänge zwischen RSSI und Datenübertragungsraten erfordern, welche an dieser Stelle nicht weiter ausgeführt werden soll.

Sind alle Voraussetzungen für die Umschaltung der Verbindung vorhanden, so löst der Client den Roaming-Prozess aus. Es erfolgt nun der Wechsel der Verbindung vom AP, zu dem eine Verbindung bestand, zum neuen AP, der mittels Neighbor Report Request ausfindig gemacht wurde. Das heißt die Verbindung zum alten AP wird getrennt und eine neue Verbindung zum Ziel-AP wird aufgebaut. Am Ziel-AP erfolgt nun neuerlich eine Authentifizierung des Clients. Auf diese Authentifizierung und die dabei verwendete Methode wird im folgenden Abschnitt eingegangen.

3.1.4 Authentifizierung am neuen AP

Aufgrund der Messungen im Kapitel 2.2.6 wird als Authentifizierungsmethode der Standard IEEE 802.1X mit einem RADIUS-Server verwendet. Um den Roaming-Vorgang zu beschleunigen wird der Standard IEEE 802.1X in Verbindung mit dem Standards IEEE 802.11r genutzt. Der Standard IEEE 802.11r schafft es dabei durch Vorauthentifizierung die Dauer und Komplexität, welche der Standard IEEE 802.1X mit sich bringt, auf ein Minimum zu reduzieren. Die Nutzung von IEEE 802.1X ist damit begründet, dass als möglicher Einsatzort eine Büroumgebung mit vielen Usern vorgesehen ist. In dieser Konstellation ist ausschließlich eine Installation mit dahinter liegendem Authentifizierungsserver einsetzbar. Beim Standard IEEE 802.11r kommt die Variante Over-the-Air zum Einsatz. Das bedeutet, dass die Vorauthentifizierung direkt am Ziel-AP passiert. Dieser Vorgang ist in der Abbildung 2.14 zu sehen. Die Variante Over-the-DS konnte aufgrund fehlenden Supports seitens der APs nicht genutzt werden.

An eine alternative Lösung, beispielsweise mit WPA2-Personal und einem einzigen Schlüssel, der allen Usern bekannt ist, ist in diesem Szenario aus Sicherheitsgründen nicht denkbar. Details zu den Protokollen IEEE 802.1X und IEEE 802.11r finden sich in den Kapiteln 2.2.2 und 2.2.6.

Der Grund für die Verwendung von IEEE 802.11k und IEEE 802.11r ist jener, dass es sich um zwei standardisierte Protokollerweiterungen handelt, die auch von diversen AP Herstellern im Businessbereich bereitgestellt werden. Der große Vorteil dieser Roaming-Lösung besteht darin, dass dazu nur Änderungen im Client, nicht aber in der Netzwerkinfrastruktur, also in den APs, notwendig sind. Die Änderungen beschränken sich des Weiteren auf die Applikationsschicht. Das heißt, es bedarf keiner Anpassungen in teilweise herstellerabhängigen Kernelmodulen. Dies bedeutet im Umkehrschluss, dass ein Einsatz nicht auf einzelne WLAN Hardware-Hersteller beschränkt ist. Die zugrunde liegende Hard- und Software muss ausschließlich den Standard IEEE 802.11 und dessen Erweiterungen IEEE 802.11k und IEEE 802.11r bereitstellen.

Da die entwickelte Methode durch ihre Einfachheit glänzt, soll im folgenden Kapitel auf die daraus resultierenden Einschränkungen eingegangen werden.

3.1.5 Voraussetzungen und Einschränkungen

Klarerweise gehen mit dem entworfenen Roaming-Algorithmus auch diverse Einschränkungen einher, die an dieser Stelle ebenfalls diskutiert werden sollen. Der folgende Abschnitt widmet sich dem Thema, welche Voraussetzungen für ein optimales Roaming-Ergebnis gegeben sein müssen. Darüber hinaus sollen zu erwartende Probleme beleuchtet und etwaige Lösungen dafür ausgearbeitet werden.

Grundlegend ist klarzustellen, dass bei einem initialen Verbindungsaufbau, also beim erstmaligen Verbinden zu einen Netzwerk, immer der standardmäßige Scan- und Authentifizierungsprozess des IEEE 802.11 Standards zu durchlaufen ist. Der Initialprozess liegt also nicht im Fokus dieser Betrachtungen. Dieser Prozess kann auch nicht umgangen werden, da sich ein Client nach dem Aktivieren eines WLAN-Moduls zuerst einen Überblick über die verfügbaren Netzwerke und deren Parameter verschaffen muss. Dazu ist es notwendig, alle vorhandenen Kanäle zu scannen, ein bestimmtes Netzwerk auszuwählen und sich in weiterer Folge zu diesem zu verbinden. Dazu muss, das vom Netzwerk festgelegte Authentifizierungsverfahren, wie z. B. nach dem Standard IEEE 802.1X, durchlaufen werden. Eine Verkürzung dieses Initialprozesses ist nicht möglich.

Die Beschleunigung des zuvor entwickelten Prozesses bezieht sich nur auf einen Umschaltvorgang von einem AP, zu dem bereits eine Verbindung bestand, zu einem neuen AP, der dem gleichen Extended Service Set, also dem gleichen Netzwerkverbund angehört. Die Einschränkung des Roamings innerhalb eines Netzwerkverbunds hat mehrere Gründe. Eine Pre-Authentifizierung mittels des Standards IEEE 802.11r ist nur innerhalb eines solchen möglich, da die Verteilung von Zugangsschlüsseln an die APs aus Sicherheitsgründen nur in einem abgeschlossenen Netzwerkverbund ablaufen darf. Ein weiterer Grund dafür ist, dass auch der Austausch von Nachbarinformationen nach dem Standard IEEE 802.11k voraussetzt, dass die einzelnen APs miteinander verbunden sind. Die Beschleunigung der Prozesse legt zugrunde, dass im Netzwerk, also zwischen den verschiedenen APs eine Verbindung besteht, über die Informationen zur effizienteren Verwaltung des Netzwerks verteilt werden können. Dazu müssen die verschiedenen APs einerseits die gleichen Standards unterstützen und andererseits müssen diese untereinander verbunden sein. Aus den gerade genannten Gründen stellt also das vorgestellte Verfahren diesbezüglich keine besondere Einschränkung dar. Es handelt sich dabei vielmehr um eine Anforderung an das Netzwerk, die notwendigen Protokolle zu unterstützen.

Des Weiteren kann es zu dem Umstand kommen, dass keine benachbarten APs verfügbar sind. Dies bedeutet klarerweise, dass kein Roaming stattfinden kann. In diesem Fall wird auf den allgemeinen Standardfall bei WLAN zurückgegriffen. Das heißt, dass eine aufrechte Verbindung so lange gehalten wird, wie es die Verbindungsqualität zulässt. Nimmt die Qualität weiter ab, der Client bewegt sich also aus dem Senderadius des APs, bricht die Verbindung irgendwann ab. Dieses Problem tritt selbstverständlich an den Grenzen der Senderadien von APs auf. In einem definierten Bereich, sei dies ein Büro oder ein anderes Gebäude, kann das Problem jedoch mit einer sinnvollen Planung und Ausstattung mit ausreichend Netzwerkinfrastruktur weitestgehend ausgeschlossen werden. Das bedeutet, dass die Standorte der APs so festzulegen sind, dass von jeder Lokation in einem definierten Bereich zumindest ein AP erreichbar ist. Dieses Problem kann somit durch ordentliche Netzplanung ebenfalls vermieden werden und stellt keine spezielle Einschränkung des entwickelten Roaming-Algorithmus dar. Vielmehr handelt es sich dabei um ein allgemeines Problem, das bei allen kabellosen Technologien zu beachten ist.

Ein weiterer Aspekt, der im entwickelten Roaming-Szenario nicht betrachtet wird, ist eine effiziente Verbindungsumschaltung aus Sicht des gesamten Netzwerkes. Beispielsweise könnte sich ein Szenario ergeben, in dem viele Clients einen guten RSSI-Wert zu einem bestimmten AP haben, da dieser an einem zentralen Standort angebracht ist. Dann kann es unter Einsatz des zuvor beschriebenen Roaming-Algorithmus im Laufe der Zeit passieren, dass alle Clients zu diesem AP verbunden sind und somit eine ungleiche Lastverteilung im Netzwerk stattfindet. Dies ergibt sich aus dem Umstand, dass der Client nur aus seiner Sicht versucht einen möglichst optimalen Zugang zum Netzwerk zu finden. Eine Sicht auf das gesamte Netzwerk bleibt ihm dabei verborgen. Der Client kann gegen dieses Problem auch nichts unternehmen, da er weder Informationen darüber hat, wie viele andere Clients zu einem bestimmten AP verbunden sind, noch wie sehr dieser AP dadurch ausgelastet ist. Dies bedeutet eine Optimierung mit dem Fokus auf dem Netzwerk kann nur durch einen zentralen Knoten im Netzwerk passieren. Dieser Knoten muss in der Lage sein, Daten von den APs zu sammeln und in weiterer Folge auch steuernd eingreifen zu können. Eine mögliche Lösung für dieses Problem wäre jene, dass ein AP Neighbor Report Responses versendet um zum Beispiel eine bessere Lastverteilung im Netzwerk zu erreichen. Es müsste beispielsweise in einem Szenario, in dem ein AP an seine Kapazitätsgrenzen stößt, unterbunden werden, dass selbiger AP von anderen APs als mögliches Roaming-Ziel angegeben wird. Somit würden keine weiteren Clients zu diesem AP wechseln. Wie zuvor angemerkt bedarf es dabei aber einer zentralen Lösung, die durch die Netzwerkinfrastruktur zur Verfügung gestellt werden muss.

Ebenfalls ist in diesem Zusammenhang anzumerken, dass ein Netzwerk hinsichtlich vieler Parameter optimiert werden kann, sei es Datendurchsatz, Antwortzeiten oder eben optimale Empfangsqualität. Außer Frage steht jedoch, dass obwohl die schlussendliche Roaming-Entscheidung beim Client liegt, der Roaming-Vorgang doch in einem gewissen Maße vom Netzwerk und den APs beeinflusst werden kann. Man denke an das zuvor erwähnte Beispiel, dass ein AP zur besseren Verteilung der Lasten im Netzwerk, bestimmte APs als mögliche Roaming-Ziele kommuniziert und gewisse APs nicht. So kann auch die Netzwerkinfrasturktur indirekt Einfluss auf den Client ausüben. Da die Optimierung, wie erwähnt, aufgrund verschiedenster Parameter erfolgen kann, müsste grundlegend untersucht werden, welche Anwendung die Optimierung hinsichtlich welcher Parameter erfordert. So könnte in einem Büro-Netzwerk die Antwortzeit ein wichtiger Parameter sein. In einem anderen Szenario wäre eventuell jedoch der Datendurchsatz von zentraler Wichtigkeit.

Darüber hinaus soll auch kurz das Thema Sicherheit betrachtet werden. Dabei wird jedoch nicht auf bekannte Risiken beim IEEE 802.11 Standard, sondern ausschließlich auf Sicherheitsrisiken, die durch das vorgestellte Roaming-Verfahren entstehen können, eingegangen. Wie zuvor erwähnt, besteht seitens des Netzwerkes die Möglichkeit einen Client mit gezielten Neighbor Report Respones in gewissem Maße zu beeinflussen. Es stellt sich die Frage, ob dies auch böswillig genutzt werden kann. Aus heutiger Sicht der Dinge ist das Risiko eher gering, da Neighbor Report Requests ausschließlich an APs geschickt werden, mit den zuvor ein Authentifizierungsverfahren durchlaufen wurde. Dies schließt aus, dass jeder beliebige AP mit böswilligen Absichten, Nachbarinformationen verteilen kann. Es wäre zwar denkbar, dass ein böswilliger Knoten manipulierte Neighbor Report Responses aussendet, um den Client auf einen ebenfalls manipulierten AP zu leiten, doch spätestens beim Versuch sich zu diesem AP zu verbinden, wird aufgrund fehlender Authentifizierung der Vorgang abgebrochen. Bei dem gerade beschriebenen Szenario kann es natürlich dazu kommen, dass der Client aufgrund des fehlgeschlagenen Verbindungsversuch zu dem manipulierten AP die Netzwerkverbindung verliert und somit eine neue Verbindung aufgebaut werden muss.

Eine größere Bedrohung wäre sicher, wenn es im Netzwerk, zu dem bereits eine Verbindung besteht, einen manipulierten AP gibt, der auch schon authentifiziert wurde. Dieses Problem ist aber ganz klar bei den Herstellern von APs und deren Software angesiedelt und es wird daher auch nicht weiter darauf eingegangen. Zusammenfassend kann das Bedrohungspotential, das durch den Roaming-Algorithmus entsteht, als gering betrachtet werden.

Zusammenfassend hat sich dieses Kapitel mit der Entwicklung eines Roaming-Algorithmus beschäftigt. Dabei wurde grundlegend der Standard IEEE 802.11k ausgewählt um aus der Sicht des Clients Informationen über benachbarte APs erlangen zu können. Dies soll verhindern, dass ein langwieriger Scan-Prozess nötig ist. Des Weiteren wurde eine konstante Überwachung der laufenden Verbindung als Notwendigkeit erarbeitet, um schon vor einem möglichen Verbindungsverlust einen Roaming-Prozess auslösen zu können. Zuletzt erfolgte eine Festlegung der Grenzwerte um optimale Roaming-Entscheidungen treffen zu können. Als Authentifizierungsmethode wurde eine Kombination aus dem Standard IEEE 802.1X beim initialen Verbindungsaufbau und dem Standard IEEE 802.11r als Grundlage für schnelle Verbindungswechsel gewählt. Abschließend wurde auf Voraussetzungen, Einschränkungen und mögliche Probleme bei der entwickelten Roaming-Methode eingegangen. Im folgenden Kapitel steht die Entwicklung eines Testaufbaus zur Performance-Messung und dem Vergleich mit anderen Technologien im Fokus.

3.2 Testaufbau

Um das entwickelte Roaming-Verfahren auch realitätsnahe testen zu können, ist es notwendig einen entsprechenden Testaufbau zu entwerfen. Der Aufbau muss einen grundlegenden Funktionstest des Roamings ermöglichen. Im Wesentlichen sollte ein Roaming-Szenario, in dem ein Client zwischen zwei APs die Verbindung wechselt, nachgestellt und verifiziert werden. Ebenfalls soll das System genutzt werden um Vergleichsmessungen zwischen verschiedenen Authentifizierungsmethoden und deren Verzögerungszeiten zu tätigen. Ein schematisches Blockdiagramm des Testaufbaus ist der Abbildung 3.2 zu entnehmen. Die beiden APs sind über eine etwaiges Netzwerk zu einem Netzwerkverbund zusammenzuschließen. Dieses Backbone-Netzwerk ist sowohl zur Übertragung von Pre-Authentifizierungsnachrichten aus dem Standard IEEE 802.11r als auch zum Austausch von Nachbar-Informationen unter den APs notwendig. Der Client kommuniziert mit den APs unter Verwendung der Protokolle IEEE 802.11b/g/n bzw. 802.11k und 802.11r. Als Anforderung für die Hardware galt, sowohl client- als auch netzwerkseitig, Geräte zu finden, die die IEEE 802.11 Erweiterungen 802.11k (Radio Resource Management) und 802.11r (Fast Basic Service Set Transition) unterstützen.

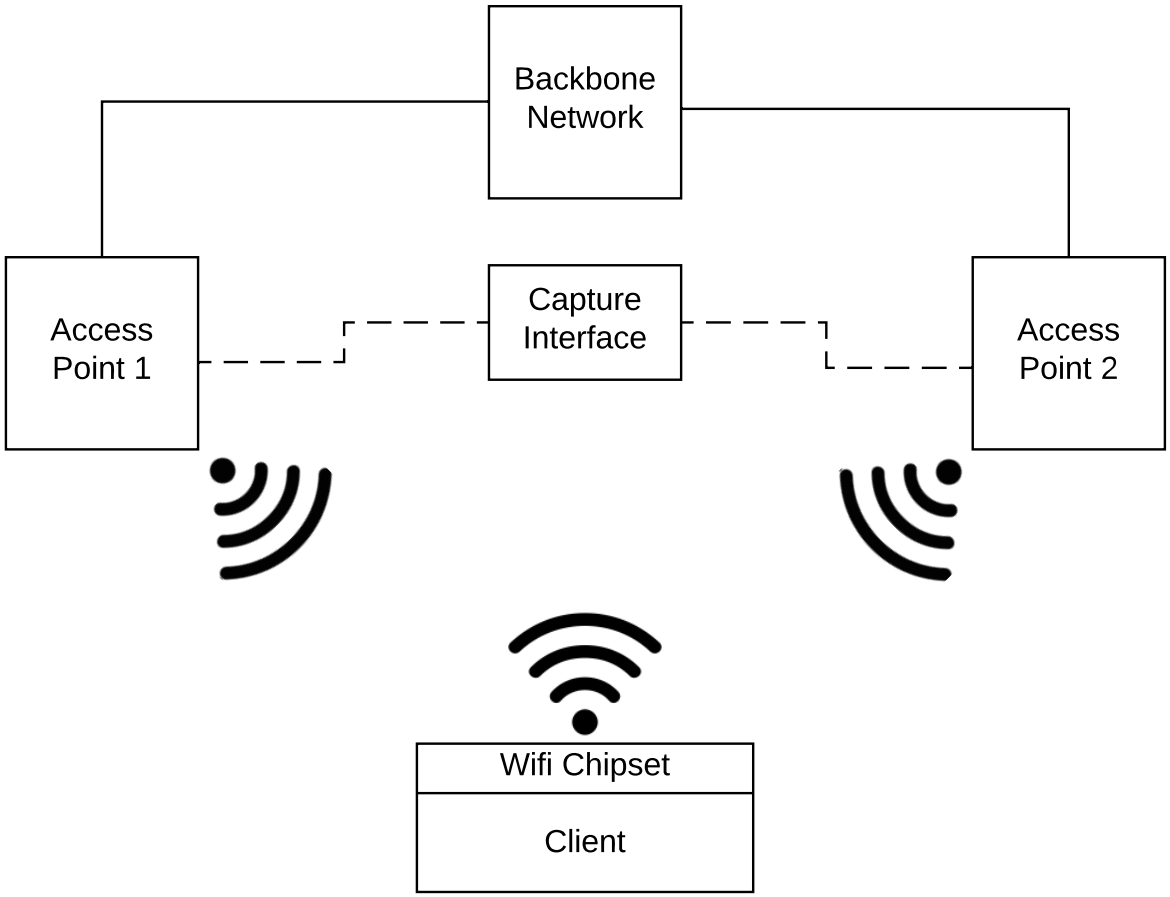

Abbildung 3.2: Blockdiagramm Testaufbau

Zusammengefasst werden an den Testaufbau folgende Anforderungen gestellt:

- Möglichkeit zur Verifikation des entwickelten Roaming-Verfahrens

- Möglichkeit zur Vergleichsmessung mit herkömmlichen Verfahren
- Bestandteile Infrastruktur:
  - 2 APs mit Unterstützung der Protokolle IEEE 802.11k und IEEE 802.11r
  - Netzwerkinfrakstruktur zum Zusammenschluss der APs
- Bestandteile Client:
  - WLAN-Chipset mit Unterstützung der Protokolle IEEE 802.11k und IEEE 802.11r
  - Computersystem zur Ansteuerung des Chipsets

Aus der Liste der Anforderungen ergab sich die Suche und Evaluierung der Komponenten für den Testaufbau. Die detaillierten Vorgaben für die APs, wie auch für das Client-System, werden dazu in den nächsten beiden Unterkapiteln behandelt. Es erfolgte ein eingehender Untersuchungs- und Vergleichsprozess unterschiedlichster Hersteller und Architekturen um mögliche Hardware-Komponenten zu finden.

3.2.1 Evaluierung APs

Wie zuvor bereits beschrieben, galt es für den Aufbau eines Testsystems geeignete APs zu finden. Diese sollten als Grundlage für Performance- und Funktionstests des entwickelten Roaming-Verfahrens dienen. Darüber hinaus musste eine dahinter liegende Netzwerkinfrastruktur geschaffen werden, um die APs zu einem Netzwerkverbund zusammenschließen zu können. Grundlegende Voraussetzung für die APs war die Unterstützung der Erweiterungen des IEEE 802.11 Standards, IEEE 802.11k (Radio Resource Management) und IEEE 802.11r (Fast Basic Service Set Transition). Darüber hinaus sollten die APs über die Möglichkeit einer Fernwartung und über die Bereitstellung von Diagnosewerkzeugen verfügen. Dies sollte beim Aufbau des Netzwerks und bei der Suche und Lösung von eventuell entstehenden Problemen helfen. Geplant ist auch, selbiges System dazu zu nutzen, die Roaming-Performance der entwickelten Roaming-Methode mit anderen Lösungen zu vergleichen. Ein Fernwartungssystem mit Analysefunktion schien als gut dafür geeignet. Zusammengefasst wurden folgende Anforderungen an die Netzwerkarchitektur gestellt:

- APs mit Unterstützung des Standards IEEE 802.11k (Radio Resource Management)
- APs mit Unterstützung des Standards IEEE 802.11r (Fast Basic Service Set Transition)
- APs mit Bereitstellung einer Fernwartungsfunktion zur Analyse des Datenverkehrs
- Netzwerkinfrastruktur zur Verbindung der APs

Da es eine Vielzahl von Herstellern Hardware für Netzwerktechnik anbietet, wurde anfangs eine Liste an möglichen Firmen erarbeitet. In weiterer Folge wurden Anfragen an die Hersteller versandt, um Geräte die die IEEE 802.11 Erweiterungen 802.11k (Radio Resource Management) und IEEE 802.11r (Fast Basic Service Set Transition) unterstützen, zu finden. Einen guten Überblick über Geräte, die diese beiden Standards unterstützen, bot die Website der Wi-Fi Alliance [7]. Bei der Wi-Fi Alliance handelt es sich um ein Konsortium an Firmen, die sich unter anderem einer besseren Interoperabilität von Netzwerk-Equipment verschrieben haben. Dazu bietet dieses

Konglomerat an Industriebetrieben eine breite Palette an Zertifizierungen. In diesem Zusammenhang war für diese Arbeit das Zertifikat Voice-Enterprise von Interesse. Geräte, die mit dem Voice-Enterprise Zertifikat ausgestattet sind, müssen unter anderem sowohl den Standard IEEE 802.11k als auch den Standard IEEE 802.11r unterstützen. Eine Produktsuche, gefiltert nach Geräten, die Voice-Enterprise fähig sind, brachte einen ersten Überblick über mögliche Hardware.

Zusammen mit den direkten Anfragen an Hersteller von Netzwerk-Hardware wurde folgende Liste (siehe Tabelle 3.1) ausgearbeitet. Die Aufzählung enthält Informationen zu diversen APs, aber auch Wireless-Controllern. Dabei ist neben der Typenbezeichnung und dem Hersteller auch angegeben, ob ein bestimmtes Gerät die geforderten Standards unterstützt. Zusätzlich ist der Tabelle zu entnehmen, ob ein Gerät auch für den Außenbereich zugelassen ist. Dort, wo die Information zu beschaffen war, ist nebenbei auch noch ein Richtpreis für das jeweilige Gerät angeführt. Hersteller, die ausdrücklich angaben, keine der beiden Standards zu unterstützen, wurden ebenfalls in die Auflistung aufgenommen. So gaben die Firmen Asus, D-Link, Lancom, Mikrotik, Proxim, TP-Link und Ubiquiti an, die benötigten Protokolle nicht zu unterstützen.

Aufgrund der Preisunterschiede zwischen Geräten für den Innen- oder Außenbereich wurde entschieden, die Suche auf die billigere Variante für den Innenbereich einzuschränken. Die Preisdifferenz zu den Geräten, die für eine Montage im Außenbereich ausgelegt sind, ist aufgrund der geforderten mechanischen Ausführung doch eklatant. Des Öfteren boten aber Hersteller den gleichen AP, einmal in einem Gehäuse für den Innenbereich und einmal für den Außenbereich, an. Die Funktion der Geräte unterschied sich ansonsten jedoch nicht.

Weitere Analysen ergaben, dass für die Verwendung der geforderten Standards bei einem Teil der Anbieter zusätzliche Hardware, in Form eines zentralen Knoten im Netzwerk, notwendig ist. Diese Hardware wäre erforderlich, um einerseits Pre-Authentifizierungsnachrichten aus dem Standard IEEE 802.11r und andererseits Nachbar-Informationen aus dem Standard IEEE 802.11k unter den APs austauschen zu können. Diese sogenannten Netzwerk-Controller befinden sich jedoch im Hochpreissegment und kosten somit ein Vielfaches der APs. Außerdem sollten im Testaufbau lediglich zwei APs verwendet werden. Diese beiden APs mit einem teuren Netzwerk-Controller zu verbinden, ist aus wirtschaftlichen Sicht nicht als Lösung denkbar. Somit konnten die Hersteller Cisco, Fortinet, Meru, Motorola/Zebra, Ruckus, SMC/Edge Core und Zyxel ebenfalls aus der Auswahl gestrichen werden. Somit grenzte sich die Auswahl auf die Hersteller Aerohive, Compex und Extrem Networks ein. Zur endgültigen Auswahl wurden folgende Geräte gegenübergestellt:

- Aerohive HiveAP 141
  - 802.11a/b/g/n Dual Radio AP, 2x2 Antenna Connectors for 2.4 GHz and 5 GHz
  - Chipsets: Qualcomm Atheros AR9344 and AR9382
- Compex MiMo Zen Series AP MMZ344
  - 802.11ac/b/g/n Dual Band Radio AP
  - Chipset: Qualcomm Atheros AR9344 and QCA9882
- Extrem Networks AP 3825i/e
  - 802.11ac/b/g/n Dual Band Radio AP
  - 3x3 Antenna Connectors MIMO implementation for high throughput

Modellbezeichnung	**Typ**	**Hersteller**	**Kosten**	**indoor/ outdoor**	**802.11k**	**802.11r**
Aerohive AP121	AP	Aerohive	600$	indoor	ja	ja
Aerohive AP230	AP	Aerohive	800$	indoor	ja	ja
Aerohive AP330	AP	Aerohive	1000$	indoor	ja	ja
Aerohive AP141	AP	Aerohive	500$	indoor	ja	ja
Aerohive AP170	AP	Aerohive		outdoor	ja	ja
		Asus			nein	nein
Cisco 2504 Series Wireless Controller	WLC	Cisco	560$		ja	ja
Cisco 2106 Series Wireless LAN Controllers	WLC	Cisco	1240$		ja	ja
Cisco AIR-LAP1262N-A-K9	AP	Cisco	600$	indoor	ja	ja
Access Point MMZ344	AP	Compex	166$	indoor	ja	ja
Access Point MMZ558	AP	Compex	200$	indoor	ja	ja
		D-Link			nein	nein
Access Point AP3825	AP	Extreme Networks	866$	indoor	ja	ja
Access Point AP3715	AP	Extreme Networks	650$	indoor	ja	ja
Access Point FortiAP-321C	AP	Fortinet	670$	indoor	ja	ja
Wireless Controller FortiOS 5	WLC	Fortinet			ja	ja
		LANCOM			nein	ja
Meru AP301	AP	Meru	800$	indoor	ja	ja
Meru MC1550	WLC	Meru	1000$		ja	ja
		Mikrotik			nein	nein
Motorola AP 6562	AP	Zebra	900$	outdoor	ja	ja
Motorola RFS4000 Wireless LAN Controller	WLC	Zebra	1000$		ja	ja
Motorola AP 7161	AP	Zebra	1700$	outdoor	ja	ja
Motorola AP 6532	AP	Zebra	600$	outdoor	ja	ja
Motorola AP 622	AP	Zebra	260$	indoor	ja	ja
		Proxim			nein	nein
Ruckus ZoneDirector 1200	WLC	Ruckus	1095$		ja	ja
Ruckus ZoneFlex R300	AP	Ruckus	350$	indoor	ja	ja
Wireless Access Controller EWS4502	WLC	Edge-Core			ja	ja
Wireless Access Controller EWS4606	WLC	Edge-Core			ja	ja
Access Point ECW7220-L	AP	Edge-Core		indoor	ja	ja
Access Point ECW7212-L	AP	Edge-Core		indoor	ja	ja
		TP-Link			nein	nein
		Ubiquiti			nein	nein
Zebra AP 7562	AP	Zebra	1100$	outdoor	ja	ja
WAC6503D-S Access Point	AP	Zyxel	400$	indoor	ja	ja

WAC6553D-E Access Point	AP	Zyxel		outdoor	ja	ja
NWA3560-N Access Point	AP	Zyxel	300$	indoor	ja	ja
NWA3550-N Access Point	AP	Zyxel		outdoor	ja	ja
NWA5123-NI Access Point	AP	Zyxel	150$	indoor	ja	ja
NXC5500 Wirless LAN Controller	WLC	Zyxel	2500$		ja	ja
NXC2500 Wirless LAN Controller	WLC	Zyxel	350$		ja	ja

Tabelle 3.1: Liste an Hardware und Herstellern und Information über Unterstützung der geforderten Standards

Für die Entscheidung zugunsten des Aerohive AP 141 war schlussendlich das von Aerohive zur Verfügung gestellte Mess- und Diagnose-Interface verantwortlich. Diese Funktion ist in der Lage sämtlichen Datenverkehr aufzuzeichnen bzw. diesen über einen externen PC live mitzuverfolgen. Neben den Daten, die über die beiden APs laufen, können mit dem sogenannten Promiscuous-Modus auch alle anderen Nachrichten, die über WLAN übertragen werden, analysiert werden. Dazu stellen die APs von Aerohive ein Capture Interface zur Verfügung, worüber die Aufzeichnung aller übertragenen Daten durch eine externe Stelle möglich ist. Es wurden zwei Modelle der Type Aerohive AP141 802.11n Dual Radio Access Point zum Aufbau des Testsystems herangezogen [AN15]. Der Aufbau erfolgte wie in Abb. 3.2 dargestellt. Die beiden APs wurden über einen zusätzlichen Router der Marke TP-Link verbunden. Über den zusätzlichen Netzwerkknoten wurde auch die Möglichkeit geboten, die Netzwerkdaten der beiden APs über einen weiteren Rechner in Echtzeit aufzeichnen zu können, ohne eines der WLAN-Interfaces dafür verwenden zu müssen.

Nach der Festlegung der Netzwerkinfrastruktur wird im nächsten Abschnitt die Evaluierung der Client-Seite beschrieben. Dazu war es notwendig einen passenden WLAN-Adapter und ein System zur Ansteuerung dessen zu finden.

3.2.2 Evaluierung Client-System

Untersuchungen ergaben, dass die Erweiterung IEEE 802.11r in diversen Linux-Distributionen als Softwarefeature implementiert ist und dazu keine über den Standard IEEE 802.11 hinaus, spezielle Hardware notwendig ist. Dies wäre eine Entscheidungsgrundlage, die Implementierung auf einem linuxbasierten Betriebssystem durchzuführen. Ein weitere Anforderung an das Betriebssystem war der frei verfügbare Sourcecode jener Teile, die den Standard IEEE 802.11 und die Authentifizierungsverfahren wie WPA implementieren, da davon auszugehen ist, dass an mehreren Stellen Anpassungen gemacht werden müssen.

Eine Analyse der Erweiterung IEEE 802.11k unter linuxbasierten Systemen ergab, dass dieses Feature sowohl in Hardware als auch in Software unterstützt werden muss. Das bedeutet, dass sowohl die Hardware 802.11k-fähig sein muss und darüber hinaus auch alle beteiligten Softwareteile den Standard beherrschen müssen. Erneut wurden über die Produktsuchfunktion der WiFi-Alliance Website mögliche WLAN-Chipsets ausfindig gemacht [7]. Die Filterung geschah wieder auf Produkte, die mit dem Voice-Enterprise Zertifikat ausgestattet waren. Bei den Chipsets konnte somit eine Einschränkung auf die Hersteller Intel, Qualcomm Atheros und Realtek/Mediatek gemacht werden. Konkret wurden folgenden Chipsets gefunden, die den Standard IEEE 802.11k unterstützen:

- Intel JC82546MDE
- Qualcomm Atheros AR5B22
- Qualcomm Atheros QCA9882
- Realtek RTL8812AU
- Realtek RTL8192CU
- MediaTek MT7610U

Im nächsten Schritt erfolgte die Suche nach Netzwerkadaptern, die mit den zuvor ausgearbeiteten Chipsets bestückt sind. Bezüglich Schnittstellen wurde die Suche auf Geräte, die entweder mit einer USB- oder PCI-Schnittstelle ausgeliefert werden, eingeschränkt. Es wurde zu jedem Chipset ein möglicher Adapter gesucht. Diese sind in der folgenden Liste zusammengefasst:

- Intel Centrino Ultimate-N 6300 - Dual Band 2x2 Wi-Fi Module
  - Chipset: Intel JC82546MDE
  - Interface: Mini-PCIe
  - Antenna Connector: 2x U.FL
  - Linux-Driver: iwlwifi
- Azurewave AW-NB110H - Dual Band 2x2 Wi-Fi with Bluetooth Module
  - Chipset: Qualcomm Atheros AR5B22
  - Interface: Mini-PCIe
  - Antenna Connector: 2x U.FL
  - Linux-Driver: ath9k
- COMPEX WLE600VX - Dual Band 2×2 802.11ac Module
  - Chipset: Qualcomm-Atheros QCA9882
  - Interface: Mini-PCIe
  - Antenna Connector: 2x U.FL
  - Linux-Driver: ath10k
- Proxim ORiNOCO USB-9100 - MIMO 2x2, 802.11ac USB Adapter
  - Chipset: Realtek RTL8812AU
  - Interface: USB
  - Antenna Connector: no
  - Linux-Driver: rtl8xxxu
- TP-LINK TL-WN8200ND - 300Mbps High Power Wireless USB Adapter
  - Chipset: Realtek RTL8192CU
  - Interface: USB

  - Antenna Connector: RP-SMA
  - Linux-Driver: rtl8192cu
- TP-LINK Archer T2UH - AC600-High-Gain-Dualband-USB-WLAN-Adapter
  - Chipset: MediaTek MT7610U
  - Interface: USB
  - Antenna Connector: RP-SMA
  - Linux-Driver: mt7610u_sta

Aus dieser Liste an verwendbaren Wireless-Adaptern wurden in weiterer Folge noch zwei Geräte entfernt. Dies war zuerst der USB-Adapter Proxim Orinoco USB-9100, da dieser über keinen Antennen Connector verfügte. Dies schien in Hinblick auf einen Testaufbau als wenig geeignet. Ebenfalls wurde das Modul Azurewave AW-NB110H gestrichen, da dieses den Linux-Treiber ath9k verwendet und dieser Treiber den Standard IEEE 802.11k nicht unterstützt. Somit blieb eine Auswahl an vier Modulen übrig, die für den Testaufbau geeignet waren. Aufgrund der Erwartung von weiteren Schwierigkeiten bei der Inbetriebnahme der Adapter wurde beschlossen, folgende vier Geräte anzuschaffen:

- Intel Centrino Ultimate-N 6300
- COMPEX WLE600VX
- TP-LINK TL-WN8200ND
- TP-LINK Archer T2UH

Nachdem die Auswahl der Netzwerkadapter erfolgt war, musste ein System zum Betrieb dieser gefunden werden. Da es sich um den Bau eine Prototyps handelt, schien ein Development-Board mit den benötigten Schnittstellen als gut dafür geeignet. Da bereits ein Board des Herstellers Variscite verfügbar war, sollte dieses für das Testsystem genutzt werden. Es handelte sich dabei um das Board des Typs Variscite VAR-SOM-MX6. Das System-on-Module wurde dabei in Verbindung mit einem Custom Board eingesetzt. Die Abbildung 3.3 bietet einen Überblick über die verfügbaren Schnittstellen. Wie zu erkennen, wird sowohl die USB- als auch PCI-Schnittstelle von diesem Board unterstützt. Als System-on-Module kam das VAR-SOM-MX6 mit folgenden Spezifikationen zum Einsatz:

- Freescale i.MX6 Series SoC Quad ARM® CortexTM-A9 Core, 1.2 GHz
- 1 GB DDR3 RAM
- 1 GB NAND Flash
- 2 x LVDS Display Interface
- HDMI V1.4 Interface
- On-board 10/100/1000 Mbps Ethernet PHY
- TI WiLink8 2.4/5GHz WLAN (802.11 a/b/g/n) / BT-BLE with optional MIMO

- 1 x USB 2.0 Host, 1 x OTG
- PCI Express 2.0 PHY
- 1 x SD/MMC
- Serial Interfaces (SPI , I2C, UART, I2S)
- Stereo Line-In / Headphones Out
- Single 3.3 V Power Supply

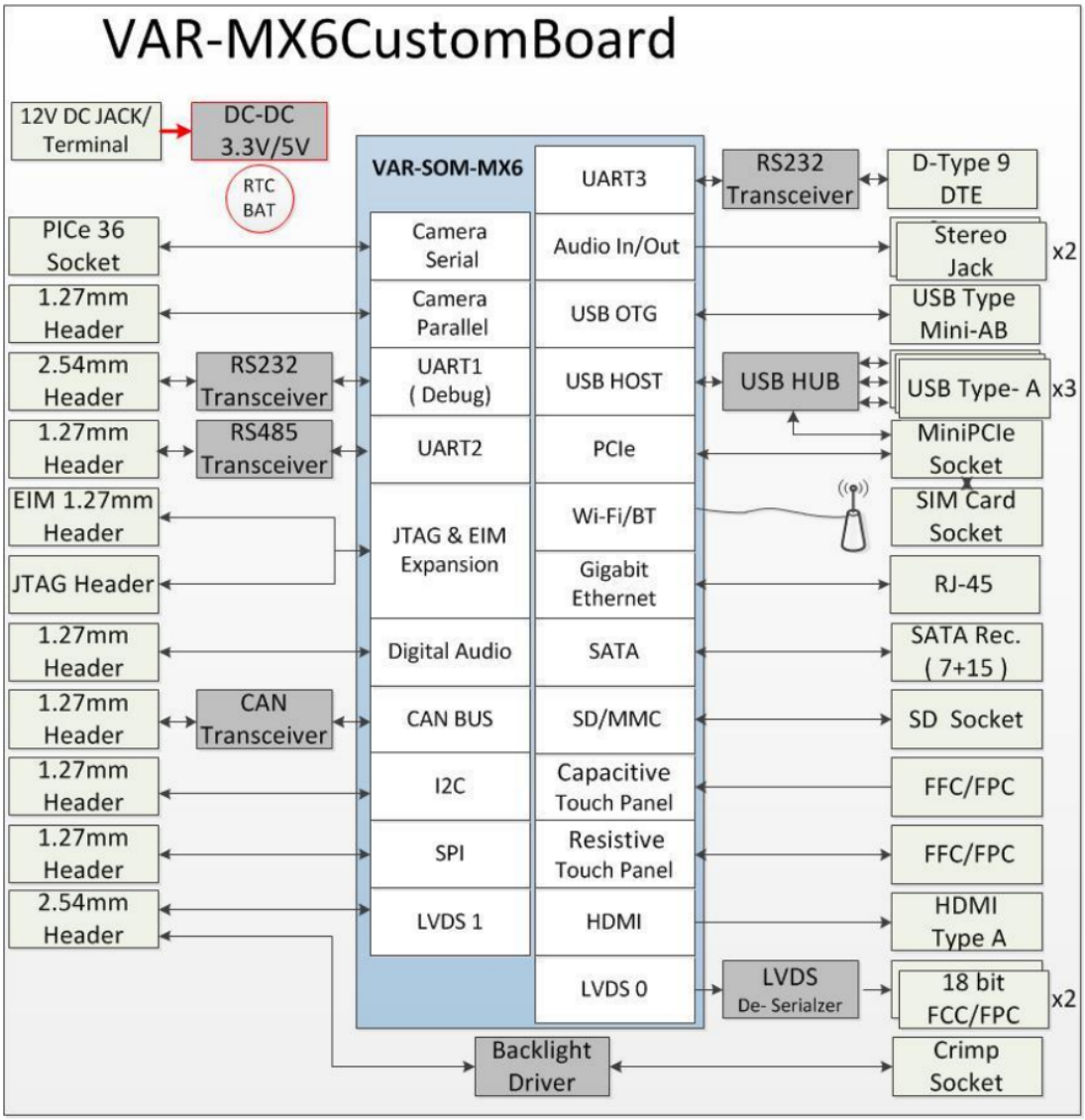

Abbildung 3.3: Blockdiagramm VAR-MX6 Custom Board [Var12]

Obwohl das Development-Board auch ein WLAN-Interface verbaut hatte, konnte dieses aufgrund der fehlenden Unterstützung des Standards IEEE 802.11k nicht verwendet werden. Somit galt es einen der ausgewählten WLAN-Adapter mit den geforderten Standards lauffähig zu bekommen.

Auf dem Development-Board kam die Linux-Kernel Version 3.14 zum Einsatz, was sich in weiterer Folge als problematisch herausstellte.

Als erstes WLAN-Modul wurde der TP-Link Archer T2UH Wifi-Stick an das Development-Board angeschlossen, wurde aber vom Betriebssystem nicht erkannt. Somit konnte vom System auch kein Treiber für dieses Modul geladen werden. Es stellte sich heraus, dass es sich um ein relativ neues WLAN-Modul handelt, welches vom Kernel zu diesem Zeitpunkt noch nicht automatisch unterstützt wurde. Eine weitergehende Recherche ergab, dass der Hersteller Treiber zum Download bereitstellte. Der entsprechende Treiber wurde geladen und für das Zielsystem kompiliert. In weiterer Folge wurde das Gerät zwar erkannt und der Treiber dafür geladen, jedoch kam es immer wieder zu Abstürzen des Betriebssystems. Die darauf folgende Analyse ergab, dass die Abstürze durch mehrere Probleme in den Kernelmodulen des zuvor installierten Treibers verursacht wurden. Dies führte zum Beschluss, dieses Modul vorerst nicht weiter zu verwenden.

Als zweiter Adapter wurde der TP-LINK TL-WN8200ND getestet. Dieser wurde, wie auch jener zuvor, über USB an das Development-Board angeschlossen. Die Erkennung des Adapters durch das Betriebssystem stellte in diesem Fall keine Probleme dar. Der dafür vorgesehene Treiber wurde automatisch geladen. Grundsätzliche Test der WLAN-Funktion verliefen ebenfalls positiv. Es sollte in weiterer Folge die Funktion der benötigten Standards überprüft werden. Dazu wurde auf den APs ein Netzwerk mit aktivierten IEEE 802.11k und IEEE 802.11r Standards erstellt. Es war jedoch nicht möglich eine Verbindung des Adapters mit diesem Netzwerk herzustellen. Eine detaillierte Analyse ergab, dass das Modul den veralteten Kernel Treiber Wireless Extension (wext) verwendet. Es stellte sich jedoch heraus, dass diese Extension die beiden benötigten Protokolle nicht unterstützte. Dazu war die Verwendung der aktuellen Netlink 80211 Schnittstelle Voraussetzung. Diese funktionierte jedoch nicht in Kombination mit dem Kernelmodul rtl8192cu. Eine weitere Untersuchung ergab die Verfügbarkeit des alternativen Kernelmoduls rtl8xxxu. Dieses sollte sowohl in Verbindung mit der Netlink Schnittstelle als auch mit dem WLAN-Adapter funktionieren. Problematisch war jedoch in weiterer Folge, dass dieses Kernelmodul erst ab einer aktuelleren Kernelversion 4.4 verfügbar war. Der erste Lösungsansatz dazu war, das Kernelmodul auf die zur Verfügung stehende Kernelversion 3.14 zurück zu portieren. Dies war jedoch nicht erfolgreich. Die zweite Lösung war umgekehrt das Development-Board auf die neue Kernelversion hochzuziehen. Nach mehreren Versuchen und Support-Anfragen beim Hersteller des Boards wurde festgestellt, dass dies nicht in absehbarer Zeit zu bewerkstelligen ist.

Der nächste Versuch erfolgte unter der Verwendung des Moduls COMPEX WLE600VX. Die Verbindung zum Development-Board erfolgte über die Schnittstelle Mini-PCIe. Der Adapter wurde erfolgreich vom System erkannt und der Kernelmodul ath10k dafür geladen. Etwaige WLAN Funtionstests verliefen ebenfalls erfolgreich, jedoch kam es bei der Überprüfung der Funktion der benötigten Standards wieder zu Problemen. Zu Netzwerken mit aktivierem IEEE 802.11k und IEEE 802.11r konnte keine Verbindung aufgebaut werden. Erneute ergab eine detaillierte Untersuchung Probleme mit der zur Verfügung stehenden Kernelversion. Auf Anfrage bei diversen Mailing-Listen für den Linux Kernel wurde angeraten, auf eine aktuelle Kernelversion umzusteigen, da die geforderten Protokolle in der Kernelversion 3.14 nur teilweise implementiert waren. Dies führte jedoch dazu, dass das Development-Board nicht verwendet werden konnte, da dafür keine aktuelle Kernelversion vom Hersteller bereitgestellt wurde.

Deshalb erfolgte die Entscheidung die weitere Entwicklung auf einen Standardrechner mit x86 Architektur zu verlagern. Verwendet wurde dabei der WLAN-Chipsatz Intel Corporation Centrino Ultimate-N 6300 [Cor11]. Dieser verwendete das Kernelmodul iwlwifi, welches wie zuvor gefordert, auf der neuen Netlink 80211 Schnittstelle aufsetzt. Es handelte sich bei dem WLAN-Adapter

zwar nicht um das aktuellste Modul, jedoch war dafür der notwendige Support für linuxbasierte Betriebssysteme gegeben. Aufgrund des Alters der Hardware wurden zwar nicht alle Features des 802.11k Standards unterstützt, jedoch war für die Implementierung ausschließlich das Feature des Neighbor Report Requests Voraussetzung. Die Unterstützung dieser Option war durch das Intel WLAN-Modul gegeben.

Darüber hinaus werden in der Tabelle 3.2 noch einmal die getesteten WLAN-Module zusammengefasst. Dazu wurden zu jedem Modell in der Spalte Beschreibung die jeweiligen Probleme und Einschränkungen dokumentiert.

Modell	**Beschreibung**
Intel Centrino Ultimate-N 6300	mit Verwendung des Linux Kernel Treiber Moduls iwlwifi in Verbindung mit der Netlink-Schnittstelle ist der eingeschränkte Einsatz von IEEE 802.11k möglich
Azurewave AW-NB110H	Problem durch Einsatz des veralteten Linux-Drivers ath9k, ath9k unterstützt IEEE 802.11k nicht
COMPEX WLE600VX	Probleme bei der Unterstützung von IEEE 802.11k im Mainline-Kernel
Proxim ORiNOCO USB-9100	kein Antenna Connector, somit ungeeignet für Testsystem
TP-Link TL-WN8200ND	der im Linux Kernel enthaltene Standard-Treiber nur in Verbindung mit wext (wireless extension) verwendbar; wext unterstützt jedoch IEEE 802.11k und IEEE 802.11r nicht neues Linux Kernel Treiber Modul rtl8xxxu würde die Standards unterstützen, ist jedoch aktuell nicht Teil des Mainline-Kernels
TP-Link Archer T2UH	Treiber nicht im Linux Mainline-Kernel enthalten proprietäres Kernel-Modul des Herstellers fehlerhaft

Tabelle 3.2: Zusammenfassung der untersuchten WLAN-Module und der dabei aufgetretenen Probleme

Aufgrund der festgestellten Probleme im Zusammespiel von Kernelversion, Kernlmodulen und damit entstehenden Abhängigkeiten wird der WLAN-Architektur unter linuxbasierten System das folgende Unterkapitel gewidmet. Es soll der grundsätzliche Aufbau und die Verbindung von Hardware und Software im Kernel-, wie auch im Userspace bezüglich der Anforderungen des Projekts betrachtet werden.

3.2.3 WLAN Architektur Linux

Wie schon im Kapitel 3.2 beschrieben wurde, ist für die Nutzung der geforderten IEEE 802.11 Erweiterungen eine lange Kette, bestehend aus verschiedenen Hard- und Softwareteilen erforderlich. Aufgrund der vielen Abhängigkeiten und Voraussetzungen wird in diesem Abschnitt die WLAN-Architektur unter linuxbasierten Systemen erläutert. Es wird ein Überblick über den generellen Aufbau mit Fokus auf jenen Teilen, die für die Implementierung des Roaming-Prozesses grundlegend sind, gegeben. Basis der folgenden Beschreibungen ist die Kernelversion 3.13. In aktuelleren Versionen des Kernels kann es durch diverse Änderungen am Kernel durchaus sein, dass die folgenden Beschreibungen nicht mehr korrekt sind. Die grundsätzliche Architektur ist jedoch gleich geblieben. Somit sollten Änderungen nur Detailbereiche betreffen.

Allgemeine Struktur

In Abb. 3.4 ist der Aufbau der WLAN-Architektur unter linuxbasierten Systemen dargestellt. Die Unterteilung erfolgt grundsätzlich in drei Schichten. Die unterste Schicht beschreibt die Hardware, darüber liegt der Kernelspace und wiederum darüber ist der Userspace angesiedelt.

In der untersten, der Hardware-Schicht, befinden sich sämtliche Dinge, die direkt mit der Hardware im Zusammenhang stehen. Zum Beispiel ist dort das verwendete WLAN-Modul oder auch diverse Speicher, wie Festplatten angesiedelt. Ebenfalls zur Hardware-Schicht wird jene Software gezählt, die direkt der Hardware zuzuordnen ist und benötigt wird, um überhaupt mit der Hardware kommunizieren zu können. Jegliche Firmware gehört in diese Gruppe. Bezeichnend für die Firmware ist, dass diese fest mit der Hardware verbunden ist. Die Hardware kann ohne die Firmware nicht verwendet werden. Des Weiteren ist anzumerken, dass es in den meisten Fällen auch keine Möglichkeit gibt diese Firmware zu ändern. Diese wird mit der Hardware geliefert und gibt vor, welche Funktionen eine bestimmte Hardware bietet. Wird eine Funktion zwar von der Hardware, jedoch nicht von der Firmware unterstützt, so kann diese auch nicht vom Kernel und darüber liegenden Schichten verwendet werden.

Darüber liegt die Kernel-Schicht. Diese interagiert direkt mit der Hardware. Der Kernel ist zentraler Bestandteil des Betriebssystems und übernimmt grundlegende Verwaltungsaufgaben. Dazu gehört die Organisation von Prozessen und Daten. Der Kernel bietet eine Schnittstelle zur Hardware, ist für Speicher-, Prozess- und Geräteverwaltung, sowie für das Dateisystem zuständig. Die Verwaltung des Kernels betrifft sämtliche Ressourcen, auf die ein System Zugriff hat. Beispielsweise darf eine Applikation aus dem Userspace nicht direkt auf die Hardware (z. B. einen Speicher) zugreifen. Der Zugriff erfolgt indirekt über den Kernel. Der Kernel bietet also dem Userspace verschiedene Funktionen um mit der darunterliegenden Hardware kommunizieren zu können. Dies bietet den Vorteil, dass dem Userspace eine abstrakte Sicht der Dinge durch den Kernel angeboten wird. Eine Applikation im Userspace muss sich beispielsweise nicht um Aufgaben, wie die gemeinsame Nutzung einer Ressource von mehreren Prozessen beschäftigen. Diese Aufgaben übernimmt der Kernel. Falls eine Applikation aus dem Userspace auf tiefere Strukturen im System zugreifen will, passiert dies über sogenannte Systemcalls, welche wiederum der Kernel zur Verfügung stellt.

Über dem Kernel liegt wiederum der Userspace. Dies ist jene Schicht, in dem gewöhnliche Applikationen laufen. Diese Applikationen können indirekt, über definierte Schnittstelle des Kernels, auf Systemressourcen zugreifen.

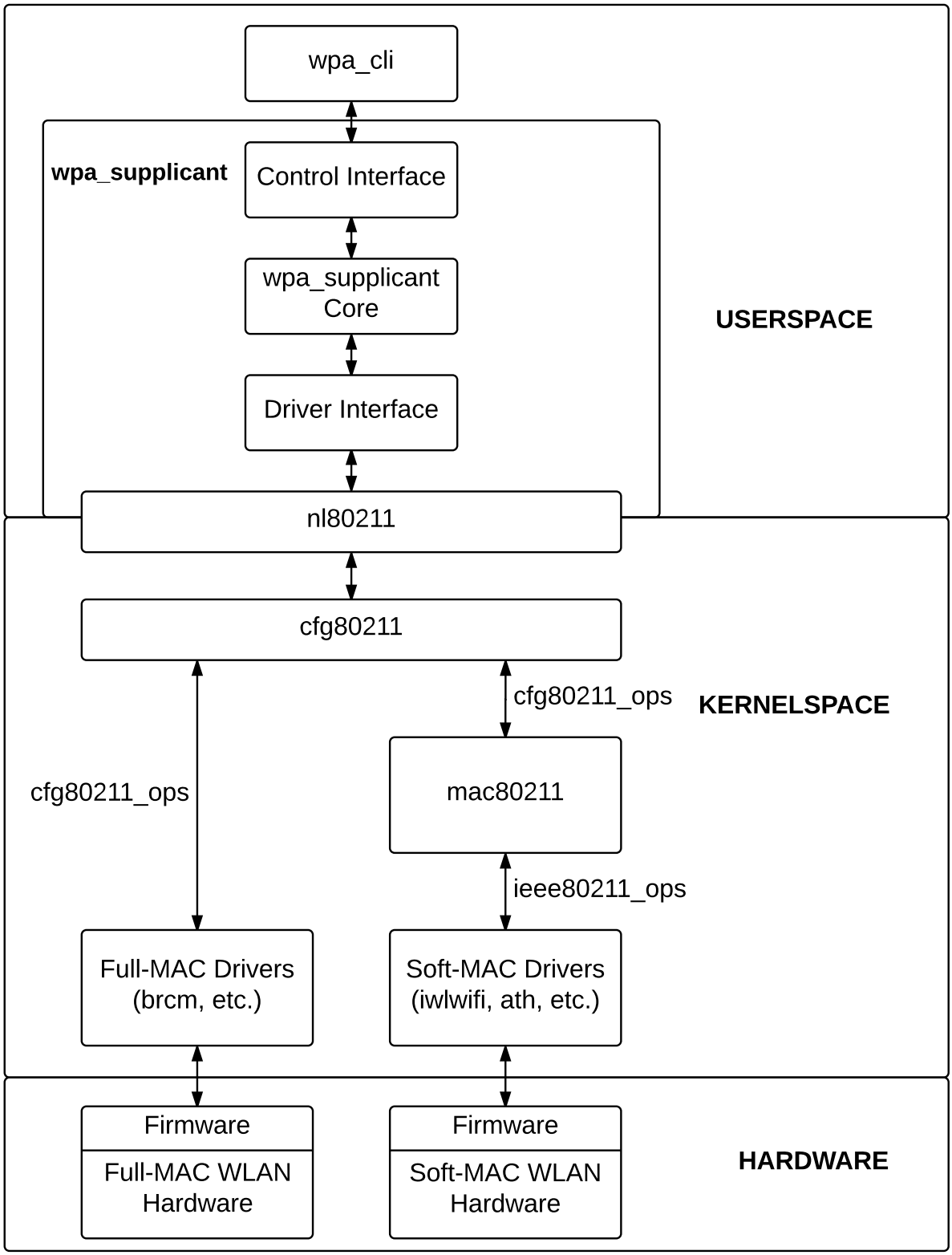

Abbildung 3.4: Blockdiagramm Linux Wi-Fi [Cor09], [4]

Hardware- und Kernel-Ebene

Auch WLAN ist in diese Schichtstruktur eingebettet. In der Hardware-Schicht ist der WLAN-Chipsatz, also die WLAN-fähige Hardware zu finden. Es gibt zwei verschiedene Typen von WLAN-Hardware. Die Unterscheidung erfolgt zwischen Soft-MAC (medium access control) und Full-MAC WLAN-Hardware. Dabei wird Hardware, die die MAC Sublayer Management Entity (MLME), also den IEEE 802.11 Standard, direkt in Hardware oder in Firmware implementiert, als Full-MAC bezeichnet. Im Gegensatz dazu wird Hardware, die die MLME in host-basierter Software implementiert, als Soft-MAC betitelt. Full-MAC-Devices kapseln somit die Komplexität des IEEE 802.11 Protokolls. Bei Soft-MAC-Devices werden hingegen nur zeitkritische Teile in der Hardware oder Firmware realisiert. Der Rest läuft am Hostsystem. In der Regel kommen vor allem in mobilen Geräten Full-MAC Geräte zum Einsatz. Dies erleichtert die Integration eines WLAN-Chipsatzes in ein komplexes System. Darüber hinaus kann bei Full-MAC Geräten Energie gespart werden, da der 802.11 Prozess in einer speziell dafür optimierten CPU abläuft.

Die Hardware wird im Normalfall über diverse herstellerspezifische Treibermodule, die im Linux-Kernel angesiedelt sind, angesteuert. Dabei ist noch anzumerken, dass bevor der Treiber mit der Hardware kommunizieren kann, im Normalfall die Firmware auf die Hardware geladen werden muss. Diese Firmware-BLOBs (binary large objects) werden vom Hardwarehersteller zur Verfügung gestellt. Diese Firmware gibt grundlegend die Funktionen der Hardware vor. Wird ein Feature oder ein Befehl nicht durch die Firmware unterstützt, so kann diese auch von darüber liegenden Schichten nicht genutzt werden.

Bei Soft-MAC Geräten wird der 802.11 Standard am Hostsystem implementiert. Die beiden parallelen Pfade, Soft- und Full-MAC sind in Abb. 3.4 ersichtlich. Es ist zu sehen, dass bei Full-MAC Geräten die Steuerung direkt erfolgt, bei Soft-MAC Geräten hingegen gibt es die zusätzliche Schicht mac80211. Diese stellt die Implementierung des IEEE 802.11 Standards in Software dar. Der Kernelspace besteht also, je nachdem ob es sich um Full-MAC oder Soft-MAC Hardware handelt aus drei oder vier Teilen. Bei Full-MAC aus dem Treibermodul (z. B. brcm für Hardware der Firma Broadcom), dem Modul cfg80211 (configuration 802.11) und nl80211 (Netlink 802.11). cfg80211 ist dabei eine API zur Konfiguration eines WLAN Gerätes. Dabei wird cfg80211 von nl80211 angesprochen. Bei Full-MAC Hardware kommuniziert cfg80211 direkt mit dem Hardware-Treiber. cfg80211 und mac80211 kommunizieren dabei mit Treibern über einen Satz an Callback Funktionen, die im Treiber implementiert sind und registriert werden, sobald dieser gestartet wird. Einige der Callback-Funktionen müssen vom Treiber implementiert werden, andere wiederum sind optional (solche die nicht von jeder Hardware unterstützt werden oder Funktionen die von mac80211 implementiert wurden und somit nicht unbedingt durch den Treiber implementiert werden müssen). Für cfg80211 sind die Callbacks im struct cfg80211_ops enthalten und für mac80211 im struct ieee80211_ops. Generell benutzen die mac80211 Version der Callbacks das Präfix ieee80211_ und die Implementierung im Treiber verwendet den Treibername als Präfix (z. B. ieee80211_tx ist die mac80211 Callback-Funktion und ath10k_tx ist die äquivalente Treiberimplementierung des ath10k Treibers).

Bei Soft-MAC Hardware hingegen gibt es zusätzlich noch die mac80211-Schicht (medium access control 802.11), die zwischen dem Hardware-Treiber und cfg80211 angesiedelt ist. Dort wiederum ist ieee80211_ops ein Satz an Funktionen, die beim Hardware-Treiber registriert werden. Als Schnittstelle zum Userspace dient nl80211. Nl80211 stellt die API zur Verwendung von netlink zur Verfügung. Dies ist wiederum eine Abstraktion des Netlink Socket Library. Nl80211 erlaubt somit einem Programm aus dem Userspace Kommandos und Konfigurationsinformation an der Kernel zu schicken (z. B. Auswahl eines Kanals oder konfigurieren der Netzwerkkarte in den

Monitormodus) und auch Information vom Kernel zurück zu erhalten (z. B. Resultate eines Scans).

Userspace-Ebene

Bis zu diesem Punkt wurden die Hardware und die Implementierung des IEEE 802.11 Standards beschrieben (entweder direkt in der Hardware bei Full-MAC oder im Kernel bei Soft-MAC). Die Steuerung aus dem Userspace erfolgt, wie bereits beschrieben über nl80211. Darüber hinaus fehlt noch die Beschreibung des Userspaces. Dort ist beispielsweise die Applikation wpa_supplicant angesiedelt. Wpa_supplicant implementiert die Standards WPA und WPA2. Die Applikation beinhaltet die Erzeugung von Schlüssel, aber auch Prozesse und Zustandsabläufe zur Authentifizierung mit einem WPA Authenticator sind dort definiert. Grob gegliedert, besteht wpa_supplicant wiederum aus drei Schichten, dem Treiber-Interface, dem wpa_core und dem Control Interface (siehe Abb. 3.4). Dabei bietet das Treiber-Interface die Schnittstelle zur Kommunikation mit nl80211. Im wpa_supplicant Kern findet sich eine große Anzahl an Funktionen wieder. Zentral ist darin eine Event Loop, welche sämtliche Abläufe steuert. Dazu gehört das Lesen von Konfigurationsdateien, die Verarbeitung von Kommandos, die über das Control Interface erfolgen, aber auch die Implementierung verschiedenster Zustandsabläufe für diverse WPA/WPA2 Authentifizierungsmethoden. Die Funktionen des WPA-Kerns sollen jedoch an dieser Stelle nicht weiter ausgeführt werden. Das Augenmerk soll auf die Kommunikation mit dem wpa_supplicant gelegt werden.

Dazu bietet wpa_supplicant ein Control Interface, das von verschiedenen Applikationen genutzt wird. Dazu zählt der Standard-Netzwerkmanager, aber auch das Tool wpa_cli. Wpa_cli ist eine Beispielapplikation, die ein großes Set an Kommandos zur Interaktion mit wpa_supplicant bereitstellt und auch genutzt werden kann, um eine Konfigurationsdatei für den wpa_supplicant zu erstellen. Neben Kommandos bietet das Control Interface weitere Möglichkeiten Statusinformationen und Event-Benachrichtigungen zu empfangen. Eine ausführliche Beschreibung des Control Interfaces findet sich unter [8]. Das Control Interface unterstützt zwei Arten der Kommunikation. Es können Kommandos abgesetzt werden, die aus einer Request Nachricht, die an das Interface geschickt wird und einem Response, die von diesem zurückkommt, bestehen. Die zweite Möglichkeit sind Nachrichten, die vom wpa_supplicant unaufgefordert gesendet werden. Um solche Nachrichten zu erhalten, muss dies über eine spezielle Funktion bekanntgegeben werden.

Linux und Roaming

Ebenfalls soll nun kurz darauf eingegangen werden, wo die einzelnen Teile des Roaming-Algorithmus angesiedelt sind. Wie zuvor beschrieben, ist für den Standard IEEE 802.11k die Unterstützung der Hardware und der dazugehörigen Firmware essentiell. Dies stellte sich vor allem beim Auswahlprozess der Hardware als große Hürde heraus, da Hardware-Hersteller zwar den 802.11k Support auf Datenblättern angaben, jedoch nicht angegeben wurde, ob die für linuxbasierte Systeme zur Verfügung gestellte Firmware diese Option ebenfalls unterstützt. Da im Testaufbau nur Soft-MAC-Hardware verwendet wurde, wird im Folgenden auch nur auf diese eingegangen. Der Standard IEEE 802.11k betrifft sowohl den MAC, SME (station management entity), MLME (MAC sublayer management entity), MIB (management information base) und das Interface für höhere Schichten [Soc08b]. Der Neighbor Report Request wird dabei in einem 802.11 Management Action Frame als Parameter übertragen (siehe Abb. 2.7). Der Request kann ebenso in Probe Requests enthalten sein. Wie zuvor schon erwähnt, muss der Standard IEEE 802.11k in

jeder Schicht implementiert sein. Dies hat natürlich zur Folge, dass die Funktionsfähigkeit von vielen Komponenten abhängt. Da für den Roaming-Algorithmus jedoch nur die Funktion Neighbor Report verwendet wurde, konnte diese Abhängigkeit ein wenig reduziert werden. Jedenfalls müssen alle drei großen Schichten (Hardware, Kernel, Userspace) den Standard unterstützen.

Der Standard 802.11r bringt ebenfalls einige Änderungen mit sich. So gibt es zwei neue und ein geändertes Inforamtion Element (Mobility Domain Information Element (MDIE) und Fast Transition Information Element (FTIE) sind neu und Robust Security Network (RSN) Information Element wurde modifiziert). Der Vorteil der Standarderweiterung 802.11r liegt jedoch ganz klar darin, dass dies keine explizite Unterstützung der Hardware fordert. Es handelt sich um reine Softwareänderungen, die einerseits den Kernel und andererseits den Userspace, konkret den wpa_supplicant betreffen.

Unter der Voraussetzung, dass sowohl Hardware, Firmware, die Kernelmodule und die Applikationen im Userspace die Erweiterungen 802.11k und 802.11r unterstützen, findet die Implementierung des Roaming-Algorithmus zum überwiegenden Teil im wpa_cli statt. Dazu werden die Kommandos, die das Control Interface des wpa_supplicant zur Verfügung stellt, verwendet (siehe Userspace Schicht in Abb. 3.4). Um den entwickelten Roaming-Algorithmus zu starten, wird der wpa_cli um einen Befehl erweitert.

4 Experimentelle Verifikation

In diesem Kapitel werden die Ergebnisse verschiedener Messungen und deren Bedeutung präsentiert. Es wird dabei ausgearbeitet, welcher Performancegewinn mit der entwickelten Roaming-Methode im Gegensatz zu verbreiteten Lösungen möglich ist und welche Vorteile aber auch Nachteile das entwickelte Verfahren bringt. Dazu erfolgt eingangs die Beschreibung eines Roaming-Ablaufs anhand einer Aufzeichnung des Netzwerkverkehrs während einem Umschaltvorgang. Anhand diesem wird die Vorgangsweise beim Roaming genauer erklärt. Danach erfolgt eine Analyse der Performance, im Falle des Roamings also die Messung der zeitlichen Dauer des Umschaltvorgangs. Darüber hinaus werden auch Roaming-Vorgänge mit laufenden Datenverbindungen analysiert und gemessen. Einleitend sollen nun grundlegende Überlegungen betreffend der Messung von Roaming-Abläufen erfolgen.

4.1 Messmethoden

Um die Performance eines Roaming-Vorganges messen zu können, muss vorab festgelegt werden, wie dieser Vorgang definiert wird, was gemessen und was außer Acht gelassen wird. Es ist weiter notwendig, sich Klarheit darüber zu verschaffen, in welcher Form Vergleichsmessungen sinnvoll und möglich sind. Wie schon eingangs beschrieben, ist Roaming das Umschalten der Verbindung von einem AP, zu dem eine aufrechte Verbindung besteht, zu einem anderen AP, der dem gleichen ESS angehört. Eine weitere Bedingung für Roaming ist, dass dieser Vorgang für Schichten über dem Data Link Layer, also Schichten über dem IEEE 802.11 Protokoll, unsichtbar ablaufen muss. Das bedeutet, dass sich beispielsweise die IP-Adresse nicht ändert oder Sessions von Applikationen nicht unterbrochen und neu aufgebaut werden müssen.

In Kapitel 2 fand eine Unterteilung des Roaming-Vorgangs in zwei Phasen statt. Dies ist einerseits die Entdeckungsphase. In dieser Phase entscheidet ein Client, dass ein Verbindungswechselvorgang eingeleitet wird und es findet die Suche nach einem möglichen Ziel-AP statt. Die zweite Phase ist die sogenannte Authentifizierungsphase. Dort wird überprüft, ob dem Client Zugang zum Netzwerk gewährt wird. Viele Vergleiche in der Literatur konzentrieren sich auf eine der beiden Phasen. Die andere wird dabei aus diversen Gründen außer Acht gelassen. Beispielsweise liegt der Fokus der Betrachtung nur auf dem Authentifizierungsalgorithmus, also jene Nachrichten von der ersten Authentifizierungsnachricht bis zur schlussendlich erfolgreich abgeschlossenen Authentifizierung (siehe dazu Abb. 1.2). Diese Dauer variiert stark nach verwendeter Authentifizierungsmethode und Aufbau des Netzwerks. Die für die verschiedenen Methoden benötigten Zeiten wurden bereits im Anschnitt 2.2 ausgearbeitet und verglichen. Das Problem, dass sich bei

dieser Betrachtung von Teilphasen ergibt, ist schlichtweg die fehlende Gesamtsicht. Auch wenn eine Authentifizierung in noch so kurzer Zeit möglich ist, bringt dies nur einen geringen Vorteil, wenn die Suche nach einem AP mehrere Sekunden dauert.

Wie auch in Kapitel 2 angemerkt wurde, kann die Suche nach APs, also der Scan-Vorgang der verschiedenen Kanäle bis zu 90 % eines Verbindsaufbaus zu einem Netzwerk bedeuten. Das heißt wiederum, dass gerade dieser, unter Umständen sehr zeitaufwändige Vorgang, nicht aus der Messung der Roaming-Performance entfernt werden sollte. Einzige Ausnahme wäre jedoch, wenn kein Scan-Prozess stattfindet. Dies soll ja beim entwickelten Roaming-Prozess der Fall sein. Einen Überblick über die möglichen Messvarianten gibt die Abbildung 4.1. Die Probe Requests und Respones sind dabei strichliert dargestellt, da diese nicht bei jeder Variante durchgeführt werden müssen. Die Dauer des Roaming-Vorgangs bei dem lediglich die Authentifizierung und Assoziation gemessen wird ist mit T_{AA} angegeben. Ist bei einem Verbindungswechsel auch ein Scan der Kanäle notwendig, so wird die Zeitdauer mit $T_{Scan+AA}$ angegeben. Aus der Sicht des Autors machen Vergleiche der einzelnen Phasen zwar Sinn um die Performance von Teilabläufen gegenüberstellen zu können, jedoch ist dies aus Sicht eines Roaming-Vorgangs nicht ausreichend.

Bis zu diesem Zeitpunkt wurde Roaming vorrangig aus der Perspektive des Data Link Layers betrachtet. Am anderen Ende des Spektrums könnte Roaming auch aus Applikationssicht betrachtet werden. Das Ziel ist ja ein nahtloser Wechsel der Verbindung, der für eine Applikation nicht merkbar abläuft. Dies bedeutet wiederum, dass eine Messung der Roaming-Performance auch mit Bezug auf eine Applikation durchgeführt werden könnte. Es wäre beispielsweise eine Messung der Dauer der Verbindungsunterbrechung aus Sicht einer Applikation möglich. Das heißt, es könnte ein Stream von Daten übertragen werden und aufgezeichnet werden, wie lange beim Roaming die Datenübertragung für die Applikation unterbrochen wird. Diese Messung hätte jedenfalls seine Berechtigung, da ja das Ziel des Roaming sein soll, dass eine mögliche Datenübertragung dadurch nicht beeinflusst wird. Jedoch bringt die Ansicht auch weitere Probleme mit sich. Einerseits muss eine Applikation, mit der die Messung durchgeführt wird, definiert werden. Dies würde jedoch wieder eine Einschränkung der Bedeutung des Messergebnisses mit sich bringen. Die Messung wäre nicht allgemein gültig, sondern nur im Kontext der einen Applikation. Diese Messvariante ist in Abbildung 4.1 mit der Zeitdauer T_{Data} angegeben.

Ebenfalls problematisch bei dieser Betrachtung ist, dass das Senden von Daten von vielen Parametern des Betriebssystems abhängt. Das bedeutet, dass Verzögerungen, die dadurch entstehen, auch in das Messergebnis einfließen. Man denke an Datenpakete, die aufgrund der System- oder Prozessorauslastung verzögert übertragen werden. Dies hätte eine Unschärfe der Messung zur Folge, bei der die Gründe für etwaige Verzögerungen von Datenpaketen nicht festgestellt werden können.

Ebenfalls darf die Betrachtung des nahtlosen Übergangs nicht außer Acht gelassen werden. Eine Verbindungsumschaltung zählt nur als solche, wenn auch die Verbindung zum Netzwerk dadurch nicht unterbrochen wird. Das heißt, dass die Umschaltung für höhere Schichten unbemerkbar vonstatten geht. Ein Abbruch der Verbindung und ein Neuaufbau sind demnach kein Roaming an sich, da alle Applikationen die über diese Verbindung laufen, unweigerlich von diesen Vorgängen Notiz nehmen, da ihre Sessions ebenfalls abreißen werden. Einzig denkbare Lösung in diesem Zusammenhang wäre, dass sich eine Applikation selbst um die Umschaltung der Sessions kümmert. Dies würde jedoch wieder die Einschränkung auf eine bestimmte Applikation bedeuten. Die allgemeine Bedeutung der Messung wäre nicht mehr gegeben.

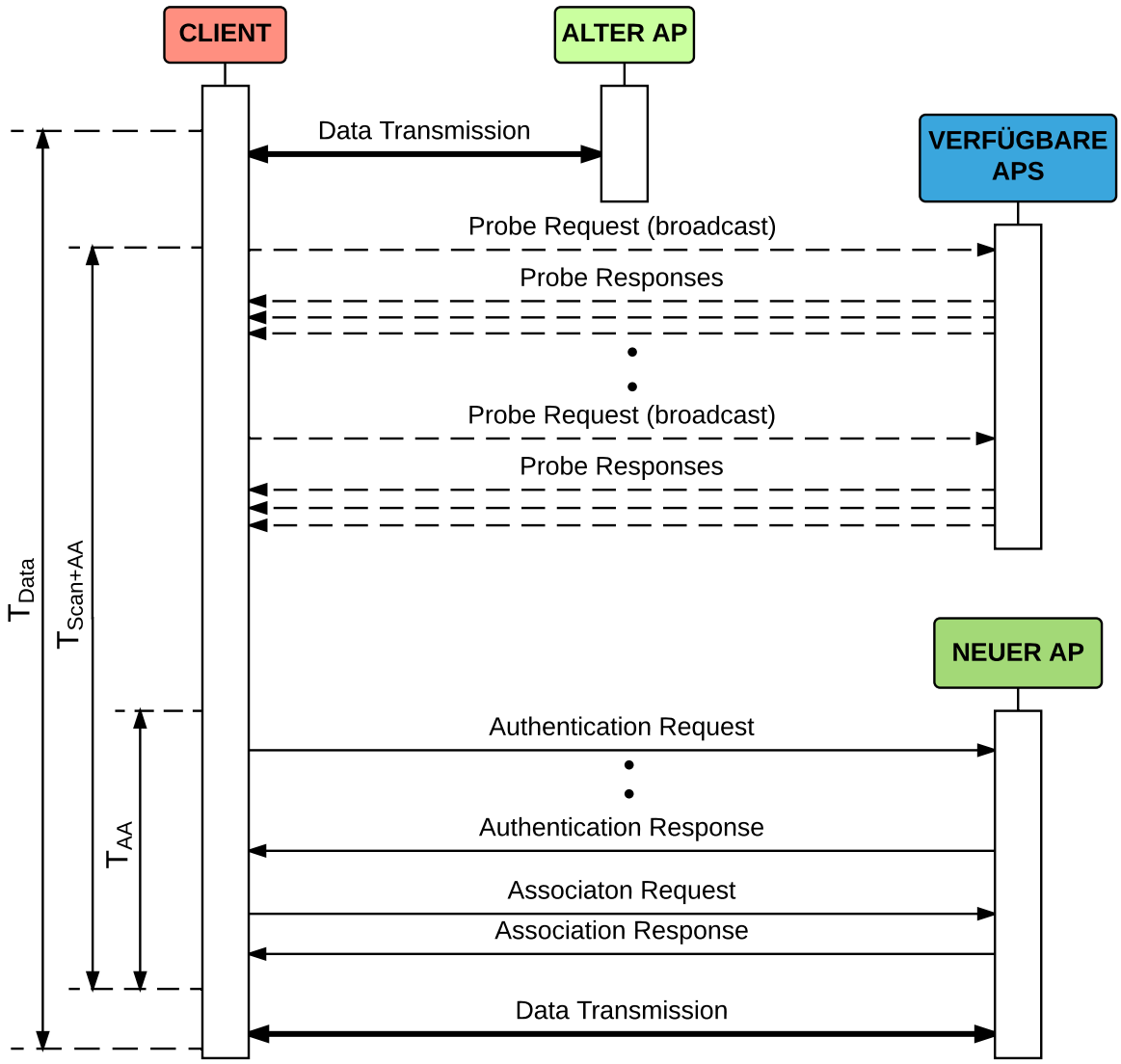

Abbildung 4.1: Übersicht über die möglichen Messvarianten eines Roaming-Vorgangs

Das bedeutet, es gibt grundlegend drei Anforderungen an die Messmethode. Ein Scan-Prozess, falls einer stattfindet, darf nicht außer Acht gelassen werden. Unschärfen die durch das Betriebssystem entstehen könnten, sind zu vermeiden. Die Betrachtung beschränkt sich auf eine wirkliche Verbindungsumschaltung, die nahtlos und unsichtbar für höhere Schichten abläuft. Mit den zuvor definierten Anforderungen an die Messung wird folgende Variante gewählt. Als Messbeginn wird jener Zeitpunkt gewählt, an dem die letzte Nachricht zum AP gesandt wurde, mit dem eine aufrechte Verbindung bestand. Die Verbindungsumschaltung ist abgeschlossen, sobald der Client am neuen AP die Authentifizierung abgeschlossen hat und eine Reassoziation stattfand. Diese ist bei der Verwendung von 802.11r, sobald der Reassociation Response vom neuen AP an den Client übermittelt wurde. Diese Messvariante entspricht somit genau der Dauer, in der zu keinem AP eine aufrechte Verbindung zum Übertragen von Daten zur Verfügung stand.

4.2 Messung der entwickelten Methode

Zu Beginn soll die Funktion des Roaming-Ablaufs noch einmal anhand einer Aufzeichnung des Nachrichtenverkehrs während eines Umschaltvorgangs beschrieben werden. Die Aufzeichnung erfolgte mit dem Programm Wireshark. Als Quelle dienten die Capture Interfaces der beiden APs, mit denen eine Live-Aufzeichnung des gesamten Nachrichtenverlaufs möglich war. Der Testaufbau fand in einer Laborumgebung statt. Die APs wurden nebeneinander aufgebaut und über eine zusätzlichen Router verbunden. Die Dämpfung von Verbindungen erfolgte durch die Abschirmung des jeweiligen APs. Dies bot die Möglichkeit unterschiedliche Empfangssituationen zu simulieren und so Roaming-Vorgänge auszulösen.

4.2.1 Analyse der Funktion

Die Aufzeichnung eines vollständigen Roaming-Ablaufs ist in Abbildung 4.2 zu sehen. Die Umschaltung der Verbindung erfolgt mit der in Kapitel 3.1 beschriebenen Methode, die auf der Überwachung der Verbindungsqualität beruht. Dabei wird sowohl die aktuelle Verbindung als auch jene zu benachbarten APs betrachtet. Unter der Verwendung des Standards IEEE 802.11k und des darin enthaltenen Neighbor Report Requests informiert sich der Client regelmäßig beim verbundenen AP über die Existenz von benachbarten APs, die als Roaming-Ziel in Frage kommen. Wird dabei ein AP gefunden, der eine bessere Verbindung zum Netzwerk bietet, so wird ein Roaming-Vorgang initiiert. Wie bereits erwähnt, wir als Maß für die Verbindungsqualität der RSSI-Wert herangezogen. Darüber hinaus müssen für ein Roaming noch weitere Voraussetzungen erfüllt sein. Beispielsweise muss auch der RSSI-Wert der aktuellen Verbindung unter einem definierten Grenzwert liegen. Die bedeutet, dass die aktuelle Verbindung unter einer definierten Qualität sein muss, damit ein Roaming-Vorgang ausgelöst wird.

Eine detaillierte Beschreibung und Analyse des aufgezeichneten Roaming-Ablaufs erfolgt nun anhand der Grafik 4.2. Am Roaming-Vorgang sind zwei APs und ein Client beteiligt. Der erste AP wird identifiziert mit der MAC-Adresse f0:9c:e9:5a:3e:d9, der zweite AP mit der MAC-Adresse f0:9c:e9:5a:66:d9. In der Abbildung sind diese jeweils in der Spalte Source oder Destination mit der Bezeichnung Aerohive_5a:66:d9 und Aerohive_5a:3e:d9 zu sehen. Der Client hat die MAC-Adresse 24:77:03:c8:00:e4. Zur Übersichtlichkeit wird darüber hinaus die Kommunikation vom Client mit je einem AP farblich hinterlegt. Nachrichten, die zwischen dem Client und dem ersten AP (MAC: f0:9c:e9:5a:3e:d9) ausgetauscht werden, sind hellblau dargestellt. Jene die zwischen dem Client und dem zweiten AP (MAC: f0:9c:e9:5a:66:d9) übermittelt werden, sind gelb markiert.

Wie zuvor beschrieben, werden zur Ermittlung von möglichen Roaming-Zielen sogenannte Neighbor Report Request an den AP, zu dem aktuell eine Verbindung besteht, gesendet. Die Aussendung von Neighbor Report Request passiert dabei in einem Abstand von 10 s. Diese sind in der Grafik als sogenannte Action-Nachrichten zu sehen. Beispielsweise handelt es sich bei den Nachrichten mit den Nummer 2809, 5758 und 8407 um derartige Requests. Einem Neighbor Report Requests folgt jeweils ein Neighbor Report Response. Diese sind die Antworten, die vom AP zurückgeschickt werden (siehe Nachrichten 2811, 5760 und 8409). Eine dieser Antworten vom ersten AP ist in 4.3 detailliert zu sehen. In der blau markierten Zeile ist dabei der Tag Neighbor Report ersichtlich. Darin wird der zweite AP (MAC: f0:9c:e9:5a:66:d9) als möglicher Nachbar angegeben. Die MAC-Adresse des zweiten APs ist in der Zeile BSSID zu sehen. Ebenfalls wird im Neighbor Report der Kanal, auf dem der benachbarte AP sendet, angegeben. Diese Information erleichtert dem Client den Scan-Prozess, da nicht das komplette WLAN Frequenzband

nach möglichen Roaming-Zielen durchsucht werden muss. Im gezeigten Neighbor Report Response wird der Kanal 9 als Sendekanal des zweiten APs mitgeteilt. Darüber hinaus werden in der Zeile BSSID Information weitere Details zum benachbarten AP übermittelt. Dazu gehören die Erreichbarkeit, vorhandene Security-Funktionen, die Mobility-Domain und weitere Eigenschaften des APs.

No.	Time	Source	Destination	Info	Protocol
2809	*REF*	IntelCor_c8:00:e4	Aerohive_5a:3e:d9	Action, SN=1931, FN=0, …	802.11
2811	0.001003	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Action, SN=275, FN=0, F…	802.11
2881	0.240477	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Probe Response, SN=276,…	802.11
2885	0.251777	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Probe Response, SN=277,…	802.11
2889	0.260676	IntelCor_c8:00:e4	Aerohive_5a:3e:d9	QoS Data, SN=2, FN=0, F…	802.11
2934	0.421749	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Probe Response, SN=278,…	802.11
3012	0.633961	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Probe Response, SN=2278…	802.11
3014	0.634926	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Probe Response, SN=2279…	802.11
3044	0.716278	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Probe Response, SN=2281…	802.11
3208	1.261421	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Probe Response, SN=2287…	802.11
5758	10.004542	IntelCor_c8:00:e4	Aerohive_5a:3e:d9	Action, SN=2009, FN=0, …	802.11
5760	10.004811	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Action, SN=279, FN=0, F…	802.11
5772	10.028639	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Probe Response, SN=280,…	802.11
5838	10.287920	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Probe Response, SN=281,…	802.11
5866	10.398803	IntelCor_c8:00:e4	Aerohive_5a:3e:d9	QoS Data, SN=30, FN=0, …	802.11
5881	10.446804	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Probe Response, SN=282,…	802.11
5898	10.500992	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Probe Response, SN=283,…	802.11
5961	10.694139	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Probe Response, SN=2390…	802.11
5963	10.694813	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Probe Response, SN=2392…	802.11
6028	10.897802	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Probe Response, SN=2394…	802.11
8407	20.007654	IntelCor_c8:00:e4	Aerohive_5a:3e:d9	Action, SN=2091, FN=0, …	802.11
8409	20.007944	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Action, SN=284, FN=0, F…	802.11
8497	20.263707	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Probe Response, SN=285,…	802.11
8509	20.286298	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Probe Response, SN=286,…	802.11
8519	20.311024	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Probe Response, SN=287,…	802.11
8565	20.472553	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Probe Response, SN=288,…	802.11
8585	20.559330	IntelCor_c8:00:e4	Aerohive_5a:3e:d9	QoS Data, SN=33, FN=0, …	802.11
8676	20.893434	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Probe Response, SN=2502…	802.11
8706	20.995246	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Probe Response, SN=2503…	802.11
8719	21.030733	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Probe Response, SN=2505…	802.11
10897	30.010563	IntelCor_c8:00:e4	Aerohive_5a:3e:d9	Action, SN=2165, FN=0, …	802.11
10899	30.010827	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Action, SN=289, FN=0, F…	802.11
10936	30.200720	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Probe Response, SN=290,…	802.11
10955	30.245211	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Probe Response, SN=291,…	802.11
11010	30.425245	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Probe Response, SN=292,…	802.11
11095	30.679767	IntelCor_c8:00:e4	Aerohive_5a:3e:d9	QoS Data, SN=3, FN=0, F…	802.11
11112	30.747718	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Probe Response, SN=2624…	802.11
11118	30.774988	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Probe Response, SN=2625…	802.11
11127	30.790047	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Probe Response, SN=2627…	802.11
13415	40.010745	IntelCor_c8:00:e4	Aerohive_5a:3e:d9	Action, SN=2239, FN=0, …	802.11
13417	40.011032	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Action, SN=293, FN=0, F…	802.11
13622	40.819895	IntelCor_c8:00:e4	Aerohive_5a:3e:d9	QoS Data, SN=34, FN=0, …	802.11
14758	45.495825	IntelCor_c8:00:e4	Aerohive_5a:3e:d9	Action, SN=2240, FN=0, …	802.11
14760	45.496113	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Action, SN=294, FN=0, F…	802.11
14781	45.567370	IntelCor_c8:00:e4	Aerohive_5a:66:d9	Authentication, SN=2241…	802.11
14784	45.575015	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Authentication, SN=256,…	802.11
14785	45.582181	IntelCor_c8:00:e4	Aerohive_5a:66:d9	Reassociation Request, …	802.11
14788	45.587678	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Reassociation Response,…	802.11
14790	45.604801	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Action, SN=259, FN=0, F…	802.11
14791	45.605108	IntelCor_c8:00:e4	Aerohive_5a:66:d9	Action, SN=2243, FN=0, …	802.11
14793	45.605386	Aerohive_5a:66:d9	IntelCor_c8:00:e4	QoS Data, SN=0, FN=0, F…	802.11
17305	55.503039	IntelCor_c8:00:e4	Aerohive_5a:66:d9	Action, SN=2248, FN=0, …	802.11
17307	55.503325	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Action, SN=260, FN=0, F…	802.11
17377	55.787810	IntelCor_c8:00:e4	Aerohive_5a:66:d9	QoS Data, SN=0, FN=0, F…	802.11
17425	55.955396	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Probe Response, SN=1973…	802.11
17437	55.985007	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Probe Response, SN=1974…	802.11
17495	56.183391	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Probe Response, SN=261,…	802.11
17497	56.184126	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Probe Response, SN=262,…	802.11
17567	56.447365	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Probe Response, SN=263,…	802.11

Abbildung 4.2: Aufzeichnung eines Roaming-Vorgangs mit der entwickelten Methode

Nach dem Empfang eines Neighbor Report Response hat der Client nun die Information, dass auf Kanal 9 ein AP ist, der als Roaming-Ziel in Frage kommen würde. Die gewonnene Information

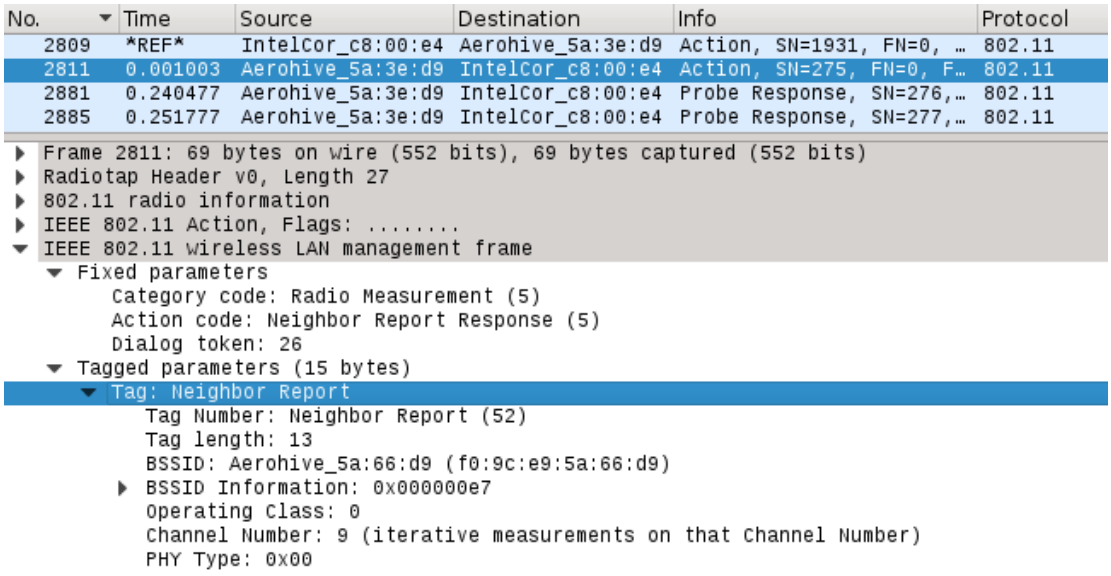

Abbildung 4.3: Beispiel eines Neighbor Report Response aus der Roaming-Messung

nutzt der Client und zeichnet die Probe Responses des zweiten APs auf Kanal 9 auf, um herausfinden, wie gut die Verbindungsqualität zu diesem AP ist. Zur Messung der Verbindungsqualität wird der RSSI-Wert herangezogen. Bei den Nachrichten mit den Nummern 3012, 3014, 3044 und 3208 in Abbildung 4.2 handelt es sich beispielsweise um derartige Probe Response Nachrichten. Die RSSI-Werte sind in der Grafik nicht zu sehen, diese sind im Graph 4.4 dargestellt. Die grüne Linie gibt dabei den RSSI-Wert zum ersten AP, die blaue Linie jenen zum zweiten AP an. Die Aussendung der Neighbor Report Request erfolgt in einem zeitlichen Abstand von 10 Sekunden (orange Markierungen). Wie im Graph ersichtlich, erfolgt die Aufzeichnung der RSSI-Werte ebenfalls alle 10 Sekunden. In der Abbildung ist weiter zu sehen, dass die Qualität der Verbindung zum ersten AP mit fortlaufender Zeit abnimmt. Jene des zweiten AP hingegen steigt mit fortschreitender Dauer. Bei 30 s sind die RSSI-Werte annähernd gleich, ehe dann der RSSI des zweiten APs den des ersten übersteigt. Wie in der Aufzeichnung in der Spalte Time zu sehen ist (Abbildung 4.2), erfolgt die Verbindungsumschaltung schließlich bei einer Zeit von 40 s bzw. bei der Nachricht mit der Nummer 14781 (rote Markierung). Die Authentifizierung und Reassoziierung (hellgrün und rot hinterlegt) dauern 20 ms. Die Dauer zwischen der letzten Nachricht, die zwischen dem Client und dem ersten AP übertragen wurde (Nachricht Nummer 14760) und dem Reassociation Response vom zweiten AP (Nachricht Nummer 14788) beträgt 92 ms. Dies ist die Dauer des Roamings, wie sie in Kapitel 4.1 definiert wurde. Sprich jene Dauer die zwischen der letzten Nachricht mit dem ursprünglichen und der erfolgreich abgeschlossenen Reassoziierung vergeht. Es sei angemerkt, dass diese Messung ohne Datenübertragung stattfand. Das bedeutet, dass hier auch nicht festgestellt werden kann, wie lange die Unterbrechung für eine Applikation wäre. Der Aufzeichnung ist nur die Dauer der Verbindungsumschaltung zu entnehmen.

4.2.2 Analyse mit laufender Datenverbindung

Der Verlauf eines Roaming-Vorgangs mit laufender Datenübertragung ist in Abbildung 4.5 zu sehen. Zur Datenübermittlung wurde das Protokoll UDP (User Datagram Protocol) gewählt.

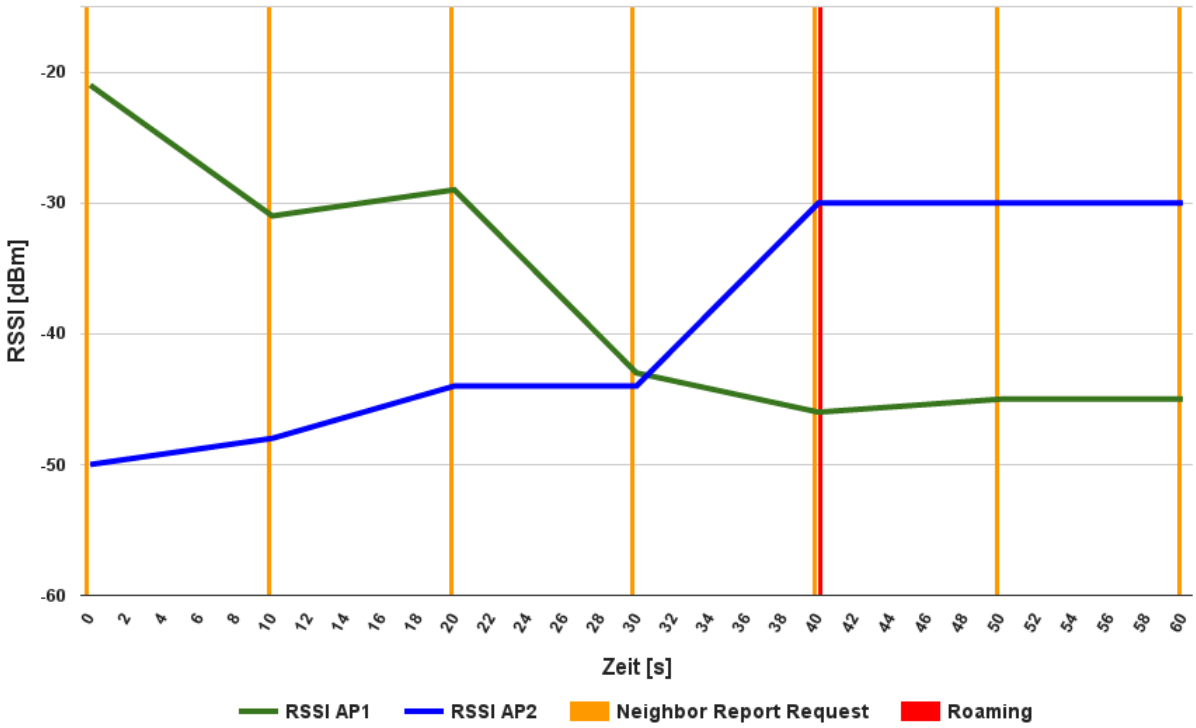

Abbildung 4.4: Verlauf der Verbindungsqualität zu den beiden APs während der Aufzeichnung einer Roaming-Abfolge

Der zu übertragende Daten-Stream wurde dafür mit der Applikation iPerf erzeugt [3]. IPerf ist ein Tool zur Analyse von erreichbaren Bandbreiten bei der Übertragung von UDP, TCP (Transmission Control Protocol) und SCTP Datenströmen (Stream Control Transmission Protocol). Die Applikation besteht dazu aus einem Server und einem Client. Der Server wartet auf einem bestimmten Port auf eingehende Daten vom Client und analysiert dann den Durchsatz des Netzwerkes. Der Client mit der MAC-Adresse 24:77:03:c8:00:e4 ist dabei die Datenquelle. Als Datensenke diente ein zweiter Rechner im Netzwerk mit der MAC-Adresse c0:3f:d5:69:18:9a. Der Client wird im Screenshot als IntelCor_c8:00:e4, der Server mit der Bezeichnung Elitegro_69:18:9a angezeigt. In der Abbildung 4.6 werden die übermittelten Datenpakete in der Spalte Info als "QoS Data"bezeichnet.

In grün und rot sind wieder die Authentifizierungs- und die Reassoziierungsnachrichten markiert. Das Roaming fand während einer laufenden Datenübertragung statt. Das letzte Datenpaket, das vor dem Roaming übertragen wurde, ist jenes mit der Nr. 27383. Dieses Datenpaket wurde auch als Zeitreferenz herangezogen. Des Weiteren ist zu sehen, dass 66 ms nach dem letzten Datenpaket die erste Authentifizierungsnachricht aufgezeichnet wurde. Der Roaming-Vorgang ist schließlich nach weiteren 13 ms abgeschlossen. Zu diesem Zeitpunkt wurde die Reassoziierungsnachricht gesendet. Direkt danach ist das erste Datenpaket zu sehen, das über den zweiten AP geschickt wurde. Dieses hat die Nr. 27425. Die Übertragung des ersten Datenpakets über den zweiten AP fand 96 ms nach dem letzten Datenpaket zum ersten AP statt. Die Verzögerung ist damit knapp unter der geforderten Dauer von 100 ms.

Interessant zu beobachten ist, dass der eigentliche Umschaltvorgang dabei nur 13 ms einnimmt,

No.	Time	Source	Destination	Info	Protocol
27381	89.044380	IntelCor_c8:00:e4	Aerohive_5a:3e:d9	Action, SN=1807, FN=0, …	802.11
27383	*REF*	IntelCor_c8:00:e4	Elitegro_69:18:9a	QoS Data, SN=1449, FN=0…	802.11
27385	0.000431	Aerohive_5a:3e:d9	IntelCor_c8:00:e4	Action, SN=309, FN=0, F…	802.11
27418	0.066520	IntelCor_c8:00:e4	Aerohive_5a:66:d9	Authentication, SN=1808…	802.11
27421	0.070681	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Authentication, SN=256,…	802.11
27422	0.077430	IntelCor_c8:00:e4	Aerohive_5a:66:d9	Reassociation Request, …	802.11
27424	0.079953	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Reassociation Response,…	802.11
27425	0.095982	IntelCor_c8:00:e4	Elitegro_69:18:9a	QoS Data, SN=0, FN=0, F…	802.11
27429	0.113906	IntelCor_c8:00:e4	Elitegro_69:18:9a	QoS Data, SN=1, FN=0, F…	802.11
27431	0.118334	Aerohive_5a:66:d9	IntelCor_c8:00:e4	Action, SN=259, FN=0, F…	802.11

Abbildung 4.5: Aufzeichnung eines Roaming-Vorgangs mit laufendem Datenverkehr

jedoch die Datenübertragung für annähernd 100 ms ausgesetzt wurde. Auf diese zusätzliche Verzögerung wurde teilweise schon im Kapitel 4.1 eingegangen, doch es sollen auch an dieser Stelle noch einmal die Gründe dafür aufgezeigt werden. Vorweg soll darauf hingewiesen werden, dass nicht bestimmt werden kann, wo die Verzögerungen entstanden. Einerseits könnte dies durch die Applikation iPerf passiert sein. Andererseits können weitere Verzögerungen durch das Betriebssystem oder durch andere Softwareteile verursacht worden sein. Außerdem ist der Server über einen weiteren Router in das System eingebunden. Auch diese Laufzeiten addieren sich in die Messung. Es entsteht somit eine große Ungenauigkeit in der Messung.

Gleichzeitig mit der Aufzeichnung der Datennachrichten an den APs fand auch am iPerf-Server die Aufzeichnung des UDP-Pakete statt. Serverseitig konnte keine Verzögerung der UDP-Nachrichten zum Zeitpunkt des Roamings festgestellt werden. Auch die Anzahl der empfangenen UDP-Pakte am Server bricht während dem Roaming-Vorgang nicht ein. Dies ist in Abbildung 4.6 zu sehen. Dazu wurden die übertragenen Pakete über der Zeit aufgezeichnet. Wie in der Grafik zu sehen ist, startet die Datenübertragung in der Umgebung von 32 s. Die Paketrate pendelt sich dann bei einer Rate von etwa 90 Pakete pro Sekunde ein. Das Roaming passiert bei einer Zeit von 39 s. Dies ist in der Abbildung mit einem schwarzen Punkt markiert. An dieser Stelle ist keine Abnahme der durchschnittlich empfangenen UDP-Pakete festzustellen.

Um eine genauere Aussage über die Anzahl an verlorenen UDP-Paketen machen zu können, wurde darüber hinaus noch eine Client-Server-Testapplikation entwickelt. Diese erzeugt am Client im Millisekundentakt UDP-Pakete und sendet diese an den Server. Die UDP-Nachrichten wurden mit einer Sequenznummer versehen um am Server eine Überprüfung auf fehlende Datenpakete machen zu können. Die Messung des Paketverlusts ergab, dass sich diese an die Dauer des Roamings anglichen. Dauerte ein Roaming-Vorgang 100 ms, so waren die Paketverluste auch in der Größenordnung von 100 Paketen.

4.2.3 Messung der Roaming-Dauer

In den vorangegangen Unterkapiteln erfolgte die Analyse der Roaming-Funktion, sowie eine Analyse des Roaming bei laufender Datenübertragung. Weiterführend wurde eine Messreihe durchgeführt, um die mittlere Dauer des Roaming-Vorgangs zu ermitteln. Dazu ist der Grafik 4.7 die relative Häufigkeitsdichte einer Messreihe der entwickelten Roaming-Methode zu entnehmen. Die Dauer des Roaming-Vorgangs beläuft sich auf 91 ms. Dies liegt knapp unter den geforderten 100 ms. Die Messung erfolgte wieder anhand, der in Kapitel 4.1 festgelegten Methode. Das

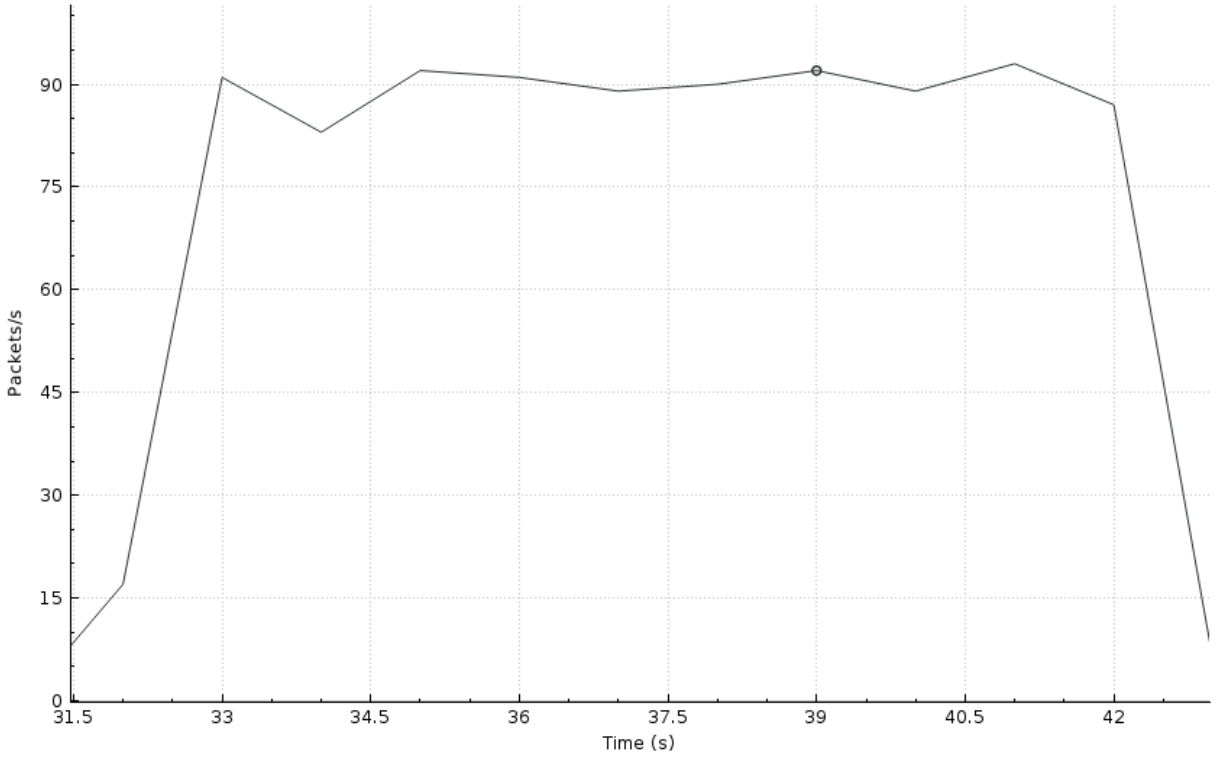

Abbildung 4.6: Aufzeichnung des UDP-Datenverkehrs während dem Roaming-Vorgang

bedeutet, es wurde die Zeit zwischen der letzten Nachricht zum ersten AP bis zur erfolgreich durchgeführten Wiederherstellung der Verbindung am zweiten AP, gemessen.

Des Weiteren wurde in Abbildung 4.8 eine Abfolge von 25 Roaming-Vorgängen und deren Dauer aufgezeichnet. Auffallend dabei ist, dass der erste Vorgang über 2.8 s gedauert hat. Es ist klarzustellen, dass es sich dabei um keinen Roaming-Vorgang sondern um den erstmalige Verbindungsaufbau zum Netzwerk gehandelt hat. Dabei besteht keine Möglichkeit den Prozess zu verkürzen. Das heißt, es musste vorab ein Scan der Kanäle, gefolgt vom kompletten Authentifizierungsverfahren durchlaufen werden. Bei allen weiteren Vorgängen handelt es sich um richtige Roaming-Abläufe. Wie im Diagramm zu sehen, dauerten diese in der Größenordnung von 100 ms.

In diesem Kapitel wurde die Funktion und Performance der entwickelten Roaming-Methode analysiert und gemessen. In weiterer Folge soll das implementierte Verfahren nun mit herkömmlichen Methoden verglichen werden.

4.2.4 Messung in realitätsnahem Aufbau

Neben den Messungen in einer Laborumgebung sollte auch eine Überprüfung in einem realitätsnahen Aufbau erfolgen. Dazu wurden die beiden APs in zwei separaten, nicht angrenzenden Räumen aufgestellt. Die Standorte der APs lagen etwa 20 m voneinander entfernt, getrennt durch mehrere Bürowände. Die Lokationen der APs waren dabei so gewählt, dass ein Client in der direkten Umgebung des ersten APs keine Verbindung zum zweiten AP aufbauen konnte.

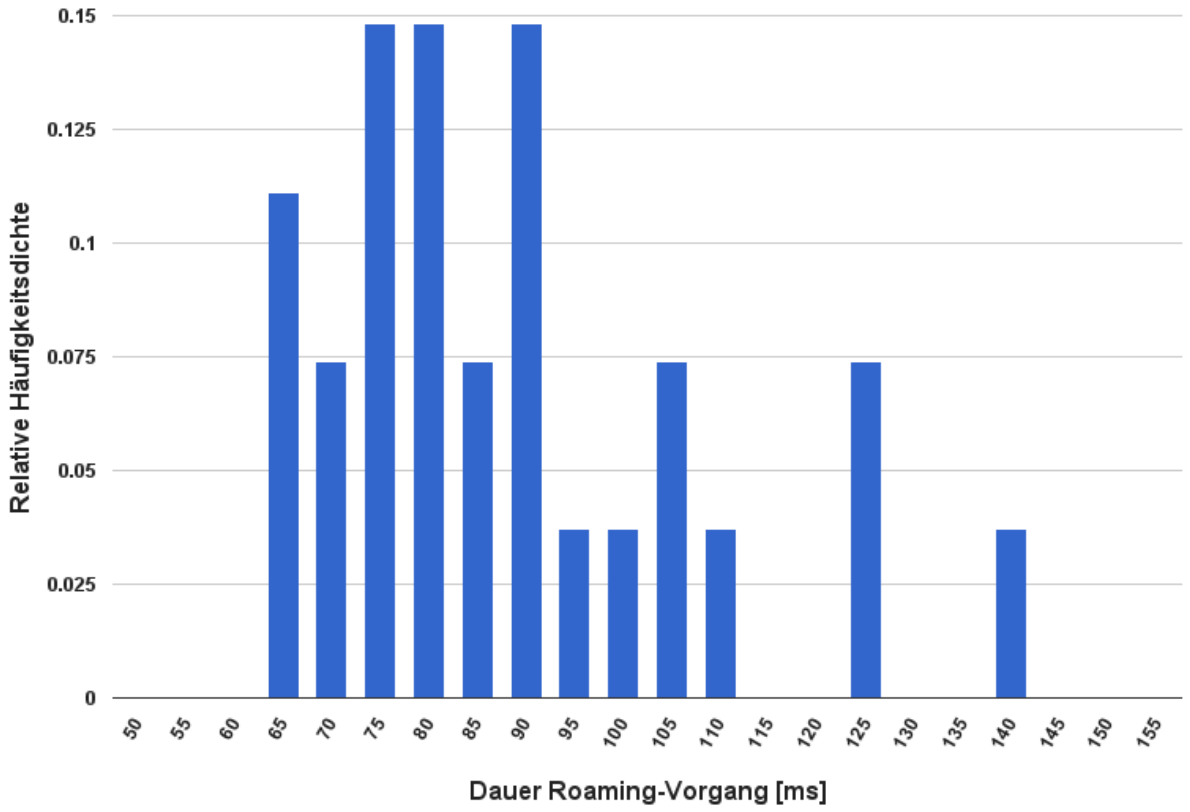

Abbildung 4.7: Relative Häufigkeitsdichte der Dauer des Roaming-Vorgangs mit der entwickelten Methode

Die Messung erfolgt so, dass im Nahfeld des ersten APs eine Verbindung zu selbigem hergestellt wurde. Danach wurde der Client in Richtung des zweiten APs bewegt. Erfolgte die Bewegung langsam, so konnte sich der Client mittels Neighbor Report Request beim ersten AP über mögliche Nachbarn informieren. Sobald man sich in den Empfangsbereich des zweiten APs bewegte, verglich das Client-System die RSSIs des ersten und zweiten APs. Passierte es in weiterer Folge, dass jener RSSI des zweiten APs zunahm und der des ersten APs sank, wurde erfolgreich ein Roaming-Vorgang ausgelöst und durchgeführt. Die Verbindung wurde somit zum zweiten AP umgeschaltet.

Zu Problemen beim Roaming kam es allerdings, sobald die Gehgeschwindigkeit erhöht wurde. Grund dafür war, wie auch schon in Kapitel 3.1.5 beschrieben wurde, dass die Frequenz der Aussendung von Neighbor Report Requests zu gering war, um der Dynamik des Netzwerkes folgen zu können. Dies bedeutet, dass es passierte, dass die Verbindung abrupt einbrach, die Überwachung dies jedoch nicht registrierte, da nur alle 10 s ein Neighbor Report Request gesendet wurde. Die Erhebung des RSSIs zum anderen AP erfolgte sobald der Neighbor Report Response empfangen wurde. Zu diesem Zeitpunkt war jedoch die Verbindung zum ersten AP bereits verloren. Die selbe Problematik ist noch einmal in Abbildung 4.9 zu sehen. Dabei ist der Verlauf des RSSI-Wertes zum AP1 in blau, jener zum AP2 in grün dargestellt. In orange ist die Aussendung der Neighbor Report Requests zu sehen. Diese erfolgen, wie in der Abbildung zu sehen, im Takt von 10 s. Der rote Balken kennzeichnet jenen Zeitpunkt, an dem die Verbindung zum AP1 verloren wurde, weil zuvor der RSSI des AP1 stark eingebrochen war. Die Abbildung zeigt also, dass die Anfragen

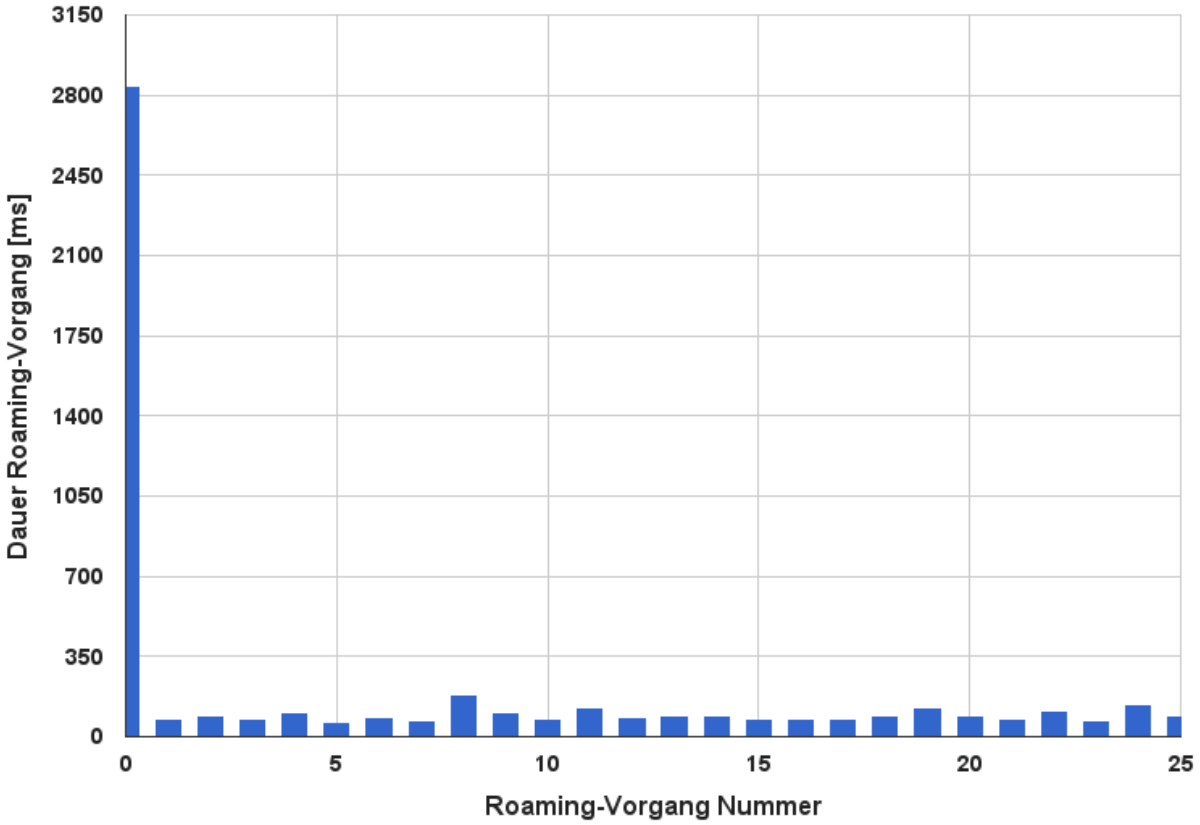

Abbildung 4.8: Dauer von mehreren Roaming-Vorgängen

bezüglich Nachbarinformationen zu selten erfolgen. Der RSSI der Verbindung brach zu schnell ein und es konnte kein Roaming-Vorgang ausgelöst werden, da die Messung der RSSI-Werte ebenfalls nur alle 10 s passiert. Dies bedeutet also, dass die Frequenz der Aussendung von Neighbor Report Request dementsprechend erhöht werden muss.

Im Testaufbau erfolgte somit eine Korrektur des Überwachungsprozesses in dem Sinne, dass der Takt der Aussendung der Neighbor Report Requests von 10 s auf 5 s verringert wurde. Dies brachte eine Verbesserung des Roaming-Ablaufs. Die Verbindung wurde, auch bei schnellerer Gangweise meistens umgeschaltet. Jedoch kam es trotzdem manchmal zum zuvor beschriebenen Problem. Somit stellte sich in weiterer Folge die Frage, wie weit der Takt zur Aussendung der Neighbor Report Request erniedrigt werden kann und was die daraus folgenden Konsequenzen sind.

Das Problem, das mit der Erhöhung der Nachbar-Abfrage einher geht, ist jenes, dass immer wenn ein Neighor Report Response empfangen wurde ein Scan jener Kanäle notwendig ist, auf denen angegeben wurde, dass benachbarte APs sind. Dies bedeutet wiederum, dass in der Zeit, in der der Scan durchgeführt wird, keine Nutzdaten übertragen werden können. Darüber hinaus muss vor dem Scan der Neighbor Report Request gesendet und auf den Neighbor Report Response gewartet werden. Auch dies verringert die Zeit, die für Nutzdatenübertragung zur Verfügung steht.

Um einen Zusammenhang zwischen Nutzdaten und Beeinflussung durch den Neighbor Report Request und den Scan herstellen zu können, muss zuvor ermittelt werden, wie lange sowohl die

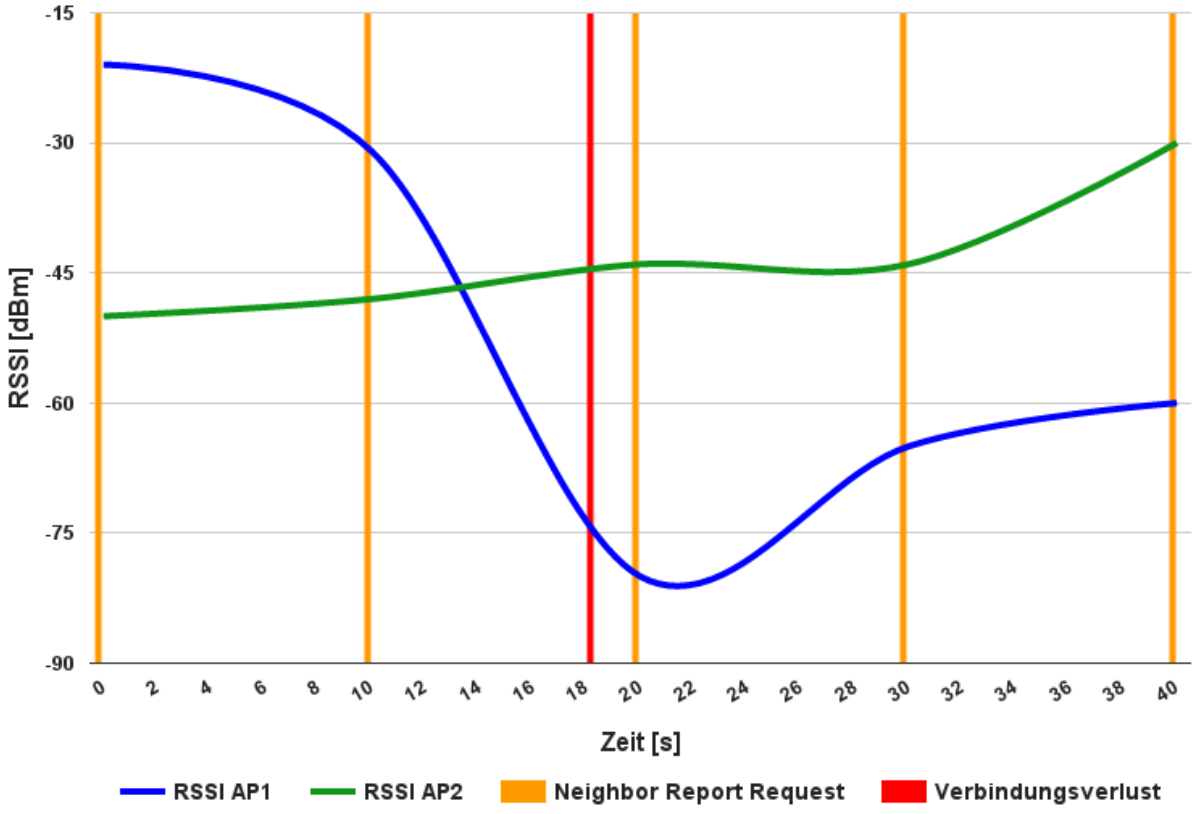

Abbildung 4.9: Problematik bei zu niedrig getakteter Aussendung von Neighbor Report Requests

Übertragung eines Neighbor Reports als auch eines Scans dauert. Messungen ergaben, dass die Sendung eines Neighbor Reports in wenigen Millisekunden abgeschlossen ist. Es wird also angenommen, dass dieser Nachrichtenaustausch in 10 ms erledigt ist. Für den Scan wird in [MDK07] angegeben, dass ein aktiver Scan theoretisch nur einige Millisekunden dauert, praktisch jedoch teilweise sogar bis zu einer Sekunde in Anspruch nehmen kann. Die im Zuge des Funktionstests durchgeführten Messungen des Roamings ergaben, dass die Scan-Dauer in der Nähe von 100 ms liegt. In Summe wird also eine Dauer von 110 ms für weitere Überlegungen verwendet. In Abbildung 4.10 ist dazu die Auswirkung der Erhöhung der Frequenz der Aussendung der Neighbor Report Request zu sehen. Die blaue Linie stellt dabei die Verhältnisse bei einem einzelnen, benachbarten AP dar. Dies bedeutet, dass beispielsweise alle 10 s eine Dauer von 110 ms für die Sendung des Neighbor Report Request und für den Scan zur Ermittlung des RSSI-Wertes nötig ist. Dies wiederum heißt, dass in diesem Fall 1.1 % der Zeit nicht zur Übertragung von Nutzdaten verwendet werden kann. Umso kleiner nun der Takt zur Aussendung der Neighbor Report Requests gewählt wird, desto weniger Zeit steht zur Nutzdatenübertragung zur Verfügung. Würden die Neighbor Request und der Scan alle 110 ms durchgeführt werden, wäre also keine Zeit mehr für Nutzdatenübertragung übrig. Des Weiteren werden im Diagramm 4.10 die Verhältnisse bei mehreren benachbarten APs dargestellt. Dies erhöht klarerweise die Scan-Dauer, da auf jedem Kanal, auf dem ein benachbarter AP verfügbar ist, gescannt werden muss. Neighbor Report Request muss hingegen nur ein einzelner ausgesendet werden.

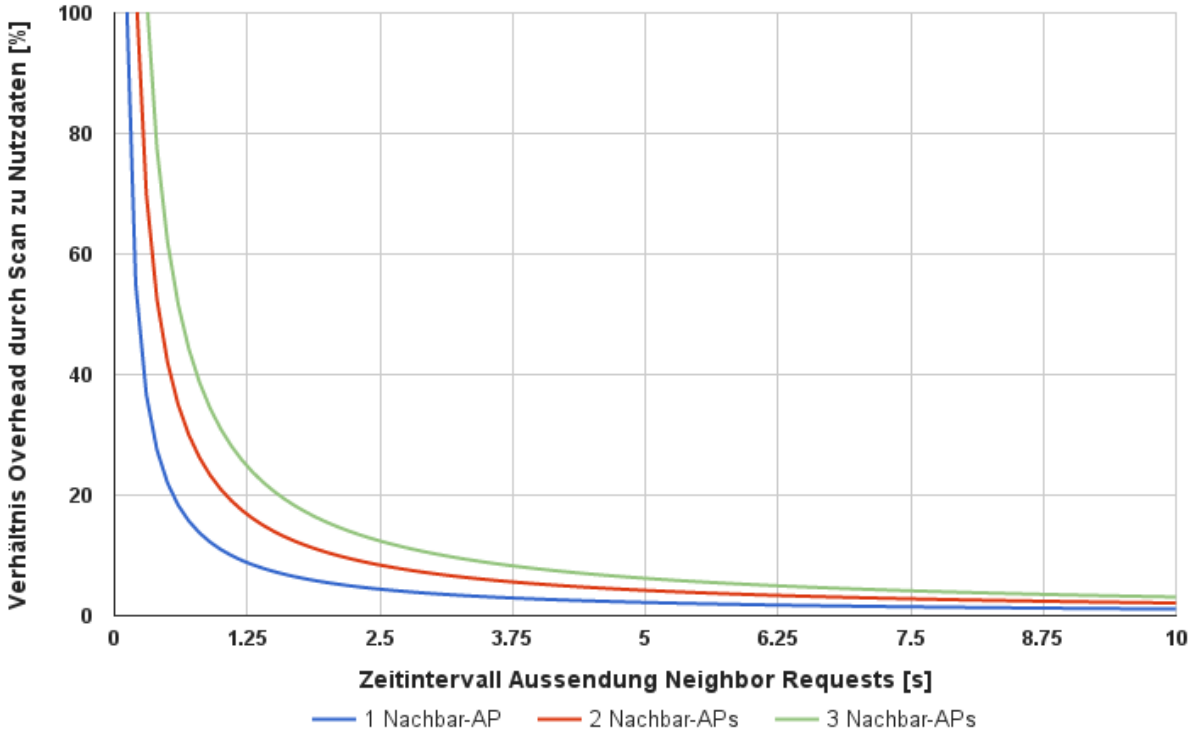

Abbildung 4.10: Verhältnis Dauer des Scans zu Nutzdatenübertragung

4.3 Vergleichsmessung mit bekannten Verfahren

Um die Performance des Roaming-Verfahrens einschätzen zu können, wurden Messungen mit verschiedenen, anderen WLAN-Konfigurationen durchgeführt. Vorweg ist dabei anzumerken, dass alle folgenden Vergleichsmessungen von der Problematik betroffen sind, dass diese die Verbindungsqualität der aufrechten Verbindung zu einem AP nicht beobachten. Das bedeutet, dass so lange mit einem bestimmten AP kommuniziert wird, bis die Verbindung abbricht. Die hat zur Folge, dass ein Roaming, wie es von der Lösung in dieser Arbeit gefordert wurde, gar nicht möglich ist. Vielmehr handelt es sich bei den folgenden Verfahren um einen Abbruch der Verbindung und einem erneuten Verbindungsaufbau. Dies kann eigentlich nicht als Roaming betrachtet werden, da dies vor allem für höhere Schichten nicht transparent abläuft. Für höhere Schichten ergibt sich generell das Problem, dass bei dieser Art und Weise der Trennung und erneuten Verbindung jeder aufgebaute Kanal, zum Beispiel ein TCP-Stream, abgebrochen wird. Schon hier bringt die entwickelte Methode also einen klaren Vorteil. Nichts desto trotz werden in der Folge drei Verfahren vorgestellt und Vergleiche zur entwickelten Variante getätigt.

Zu Beginn wird in der Grafik 4.11 ein Verbindungswechsel in einem ungesicherten Netzwerk dargestellt. Der Client mit der MAC-Adresse 24:77:03:c8:00:e4 war zu einem Netzwerk mit der Kennung f0:9c:e9:5a:3e:da verbunden. Diese Verbindung wurde unterbrochen. Dies ist ersichtlich an der Disassociate-Nachricht (Nachricht Nr. 1). Der Client beginnt daraufhin mit dem Scan der Kanäle um nach alternativen Netzwerken zu suchen. Dies ist an den Probe Response Nachrichten (Nr. 2-5) ersichtlich. Danach führt der Client die Authentifizierung und Assoziierung zum neuen Netzwerk durch. Alleine der Scan-Prozess dauert, wie in der Grafik in der Spalte Time zu sehen

ist, 3.04 s. Die Authentifizierung und Assoziierung läuft, da es sich um ein ungesichertes Netzwerk handelt, in 17 ms ab. In Summe jedoch vergehen zwischen dem Verbindungsabbruch, also der Disassociate-Nachricht, und dem erneuten Verbindungsaufbau 3.05 s. Ein Test mit laufender Datenübertragung war aus den oben beschriebenen Gründen des Verbindungsabbruchs nicht möglich.

No.	Time	Source	Destination	Info	Protocol
1	0.000000	Aerohive_5a:3e:da	IntelCor_c8:00:e4	Disassociate, SN=259,…	802.11
2	0.282636	Aerohive_5a:66:da	IntelCor_c8:00:e4	Probe Response, SN=26…	802.11
3	0.413675	Aerohive_5a:66:da	IntelCor_c8:00:e4	Probe Response, SN=26…	802.11
4	0.424058	Aerohive_5a:66:da	IntelCor_c8:00:e4	Probe Response, SN=26…	802.11
5	0.511662	Aerohive_5a:66:da	IntelCor_c8:00:e4	Probe Response, SN=26…	802.11
6	3.036315	IntelCor_c8:00:e4	Aerohive_5a:66:da	Authentication, SN=84…	802.11
7	3.047480	Aerohive_5a:66:da	IntelCor_c8:00:e4	Authentication, SN=25…	802.11
8	3.051218	IntelCor_c8:00:e4	Aerohive_5a:66:da	Association Request, …	802.11
9	3.053209	Aerohive_5a:66:da	IntelCor_c8:00:e4	Association Response,…	802.11
10	3.060643	IntelCor_c8:00:e4	Aerohive_5a:66:da	Action, SN=842, FN=0,…	802.11
11	3.060657	Aerohive_5a:66:da	IntelCor_c8:00:e4	Action, SN=258, FN=0,…	802.11

Abbildung 4.11: Verbindungswechsel in einem ungesicherten Netzwerk

In Abbildung 4.12 ist die relative Häufigkeitsdichte einer Messreihe der Dauer des Roaming-Vorgangs in einem ungesicherten Netzwerk zu sehen. Gemessen wurde, wie schon zuvor, von der Disassociate-Nachricht bis zur erneuten Assoziierung. Im Mittel ergab die Messung eine Dauer von 4.38 s. Wie ebenfalls zuvor schon erwähnt wurde, ist der Großteil der Dauer dem Scan-Prozess zuzuordnen.

Ein ähnliches Bild ergibt sich bei der Aufzeichnung der Verbindungsumschaltung unter der Verwendung des Standards WPA2-Pre-Shared Key. Dort ist der Master Key vorab, sowohl dem Client als auch dem Authenticator, bekannt (Details dazu siehe im Abschnitt 2.2.1). Es verwenden darüber hinaus alle Clients im Netzwerk den selben Master Key, was zu in Kapitel 2.2.1 beschriebenen Sicherheitsproblemen führt.

Die Aufzeichnung einer Verbindungsumschaltung ist in Abbildung 4.13 zu sehen. Wie schon beim ungesicherten Netzwerk zuvor, findet auch bei der WPA2 Pre-Shared Key Variante eine Unterbrechung der Verbindung statt (Disassociate-Nachricht). Danach muss wiederum ein neuer Scan-Prozess gestartet werden, um alternative Netzwerke zu finden. Die Suche beläuft sich dabei auf eine Dauer von 2.84 s. Die Authentifizierung und Assoziierung zum neuen Netzwerk dauert weitere 39 ms. In Summe kommt es zu einer Verbindungsunterbrechung von gesamt 2.87 s. Wie ebenfalls schon angemerkt, handelt es sich auch hier nur um eine Trennung und einen Neuaufbau einer Verbindung. Dies passiert nicht transparent für höhere Schichten. Jegliche Verbindungen, die dort bestanden, wurden dabei unterbrochen.

Die relative Häufigkeitsdichte einer Messreihe der Dauer des Roaming-Vorgangs eines mit WPA2-PSK gesicherten Netzwerkes ist im Diagramm 4.14) ersichtlich. Die gemessenen Zeiten beinhalten ebenfalls den Scan-Prozess. Im Durchschnitt betrug die Dauer des Roaming-Vorgangs 4.41 s. Erneut nimmt die Suche nach einem neuen AP und der damit verbundene Scan der Kanäle einen Großteil der Gesamtdauer ein. Dies ist auch der Grund, warum die Dauer ähnlich jener eines ungesicherten Netzwerkes ist.

Die letzte Vergleichsmessung verwendet als Authentifizierungsmethode WPA2-Enterprise. Wie schon bei den Messungen zuvor, findet auch in diesem Fall kein richtiges Roaming statt, sondern es

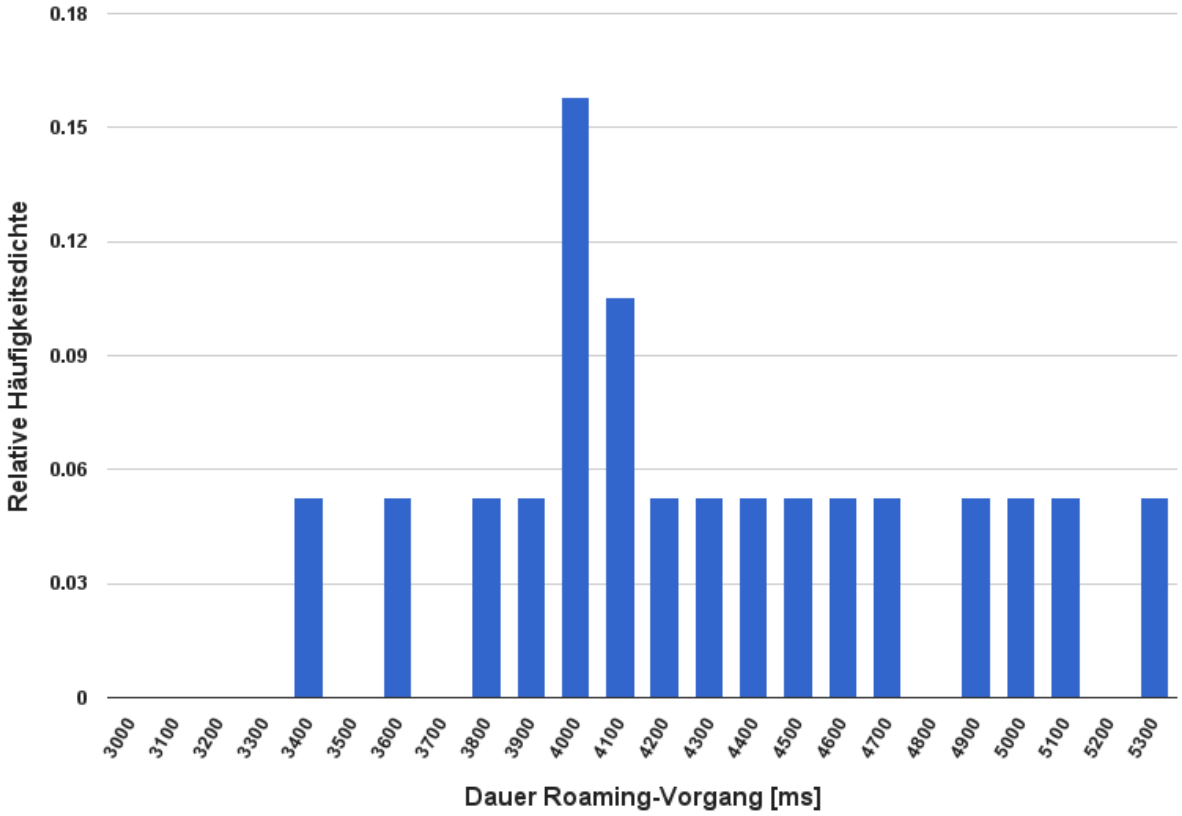

Abbildung 4.12: Relative Häufigkeitsdichte der Dauer eines Roaming-Vorgangs in einem ungesicherten Netzwerk

erfolgt ein Verbindungsabbruch gefolgt von einem erneuten Verbindungsaufbau zu einem anderen Netzwerk. Zu sehen ist dies in der Grafik 4.15. Die Disassociate-Nachricht (Nr. 1) zeigt erneut den Verbindungsabbruch. Darauf folgt ein Scan-Prozess, der 2.92 s dauert. Im Gegensatz zum zuvor gezeigten Verfahren, bietet der Authentifizierungsprozess bei WPA2-Enterprise einen starke Sicherung der Verbindung. Dieser Ablauf ist ersichtlich in den Nachrichten 11 bis 36. Dieses Verfahren dauert 2.45 s. Gesamt beläuft sich der Verbindungswechsel also auf eine Dauer von 5.38 s.

Die relative Häufigkeitsdichte der Messreihe der Roaming-Dauer ist der Abbildung 4.16 zu entnehmen. Auch hier kommt es aufgrund des langen Scan-Prozesses zu einer durchschnittlichen Roaming-Dauer von 5.34 s.

Der Vergleich der Dauer der unterschiedlichen Roaming-Methoden ist noch einmal im Diagramm 4.17 dargestellt. Auch hier ist ganz klar zu erkennen, dass es für eine gute Roaming-Performance unumgänglich ist den zeitaufwendigen Scan-Prozess zu vermeiden. Die erreichbaren Umschaltzeiten können durch den Einsatz der Protokolle IEEE 802.11k und IEEE 802.11r um Größenordnungen verringert werden.

Zusammenfassend lässt sich sagen, dass das grundlegende Problem bei den herkömmlichen Varianten des Verbindungswechsels im Verlust der Verbindung liegt. Wie in den Abbildungen 4.11, 4.13 und 4.15 ist dies durch die Disassociate-Nachricht erkennbar. Der Client verliert dadurch

No.	Time	Source	Destination	Info	Protocol
1	0.000000	Aerohive_5a:3e:d7	IntelCor_c8:00:e4	Disassociate, SN=259,…	802.11
2	0.205686	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Probe Response, SN=90…	802.11
3	0.350166	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Probe Response, SN=90…	802.11
4	0.350767	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Probe Response, SN=90…	802.11
5	0.455657	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Probe Response, SN=91…	802.11
6	0.457813	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Probe Response, SN=91…	802.11
7	2.836782	IntelCor_c8:00:e4	Aerohive_5a:66:d7	Authentication, SN=10…	802.11
8	2.837449	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Authentication, SN=25…	802.11
9	2.866883	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Key (Message 1 of 4)	EAPOL
10	2.857420	IntelCor_c8:00:e4	Aerohive_5a:66:d7	Association Request, …	802.11
11	2.857782	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Association Response,…	802.11
12	2.863909	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Key (Message 1 of 4)	EAPOL
13	2.869310	IntelCor_c8:00:e4	Aerohive_5a:66:d7	Key (Message 2 of 4)	EAPOL
14	2.875064	Aerohive_5a:66:d7	IntelCor_c8:00:e4	Key (Message 3 of 4)	EAPOL
15	2.875559	IntelCor_c8:00:e4	Aerohive_5a:66:d7	Key (Message 4 of 4)	EAPOL
16	2.881980	IntelCor_c8:00:e4	Aerohive_5a:66:d7	Action, SN=1080, FN=0…	802.11

Abbildung 4.13: Verbindungswechsel in einem mit WPA2-Personal gesicherten Netzwerk

die Anbindung an das Netzwerk und es können vom AP keine Nachrichten mehr an den Client zugestellt werden. Verbindungen, die auf darüber liegenden Schichten aufgebaut wurden, gehen verloren. Eine Datenübertragung während einem Roaming-Prozess ist daher nicht möglich.

Die zweite Thematik, die bei den herkömmlichen Varianten auftritt, ist die Dauer der AP-Suche. Sobald die Verbindung zum Netzwerk unterbrochen wurde, beginnt der Client die Kanäle zu scannen. Dies dauerte in den Aufzeichnungen mindestens 2.5 s. Dies ist in jedem Fall eine Dauer, die den Einsatz in zeitkritischen Anwendungen verhindert. Wie in Kapitel 1.2 erwähnt wurde, ist für eine Telefonat in IP-Netzwerken die maximale Verzögerung von Datenpaketen von Enduser zu Enduser 150 ms. Diese Anforderung kann somit mit herkömmlichen Varianten nicht gelöst werden.

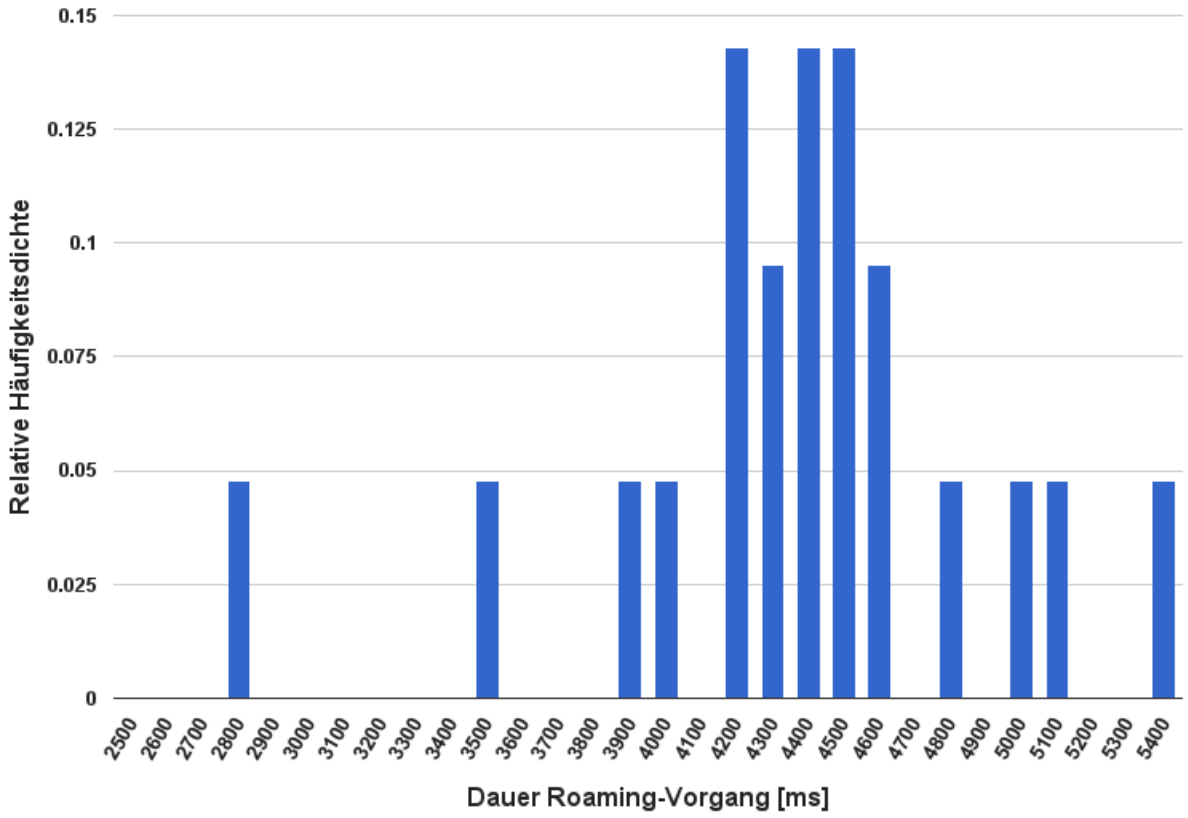

Abbildung 4.14: Relative Häufigkeitsdichte der Dauer eines Roaming-Vorgangs in einem mit WPA2-PSK gesicherten Netzwerk

No.	Time	Source	Destination	Info	Protocol
1	0.000000	Aerohive_5a:3e:d8	IntelCor_c8:00:e4	Disassociate, SN=259,…	802.11
2	0.402797	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Probe Response, SN=30…	802.11
3	0.491938	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Probe Response, SN=30…	802.11
4	2.922270	IntelCor_c8:00:e4	Aerohive_5a:66:d8	Authentication, SN=36…	802.11
5	2.920726	IntelCor_c8:00:e4	Aerohive_5a:66:d8	Authentication, SN=36…	802.11
6	2.921706	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Authentication, SN=25…	802.11
7	3.028517	IntelCor_c8:00:e4	Aerohive_5a:66:d8	Authentication, SN=36…	802.11
8	3.028798	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Authentication, SN=25…	802.11
9	3.093775	IntelCor_c8:00:e4	Aerohive_5a:66:d8	Association Request, …	802.11
10	3.096717	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Association Response,…	802.11
11	3.099830	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Request, Identity	EAP
12	5.110755	IntelCor_c8:00:e4	Aerohive_5a:66:d8	Start	EAPOL
13	5.115170	IntelCor_c8:00:e4	Aerohive_5a:66:d8	Response, Identity	EAP
14	5.128220	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Request, Protected EA…	EAP
15	5.132924	IntelCor_c8:00:e4	Aerohive_5a:66:d8	Client Hello	TLSv1
16	5.243218	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Server Hello, Certifi…	TLSv1
17	5.245640	IntelCor_c8:00:e4	Aerohive_5a:66:d8	Response, Protected E…	EAP
18	5.261075	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Server Hello, Certifi…	TLSv1
19	5.265493	IntelCor_c8:00:e4	Aerohive_5a:66:d8	Client Key Exchange, …	TLSv1
20	5.302993	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Change Cipher Spec, E…	TLSv1
21	5.303653	IntelCor_c8:00:e4	Aerohive_5a:66:d8	Response, Protected E…	EAP
22	5.311404	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Application Data	TLSv1
23	5.311917	IntelCor_c8:00:e4	Aerohive_5a:66:d8	Application Data, App…	TLSv1
24	5.324395	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Application Data	TLSv1
25	5.326083	IntelCor_c8:00:e4	Aerohive_5a:66:d8	Application Data, App…	TLSv1
26	5.325853	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Application Data	TLSv1
27	5.342560	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Application Data	TLSv1
28	5.343033	IntelCor_c8:00:e4	Aerohive_5a:66:d8	Application Data, App…	TLSv1
29	5.353880	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Application Data	TLSv1
30	5.356494	IntelCor_c8:00:e4	Aerohive_5a:66:d8	Application Data, App…	TLSv1
31	5.366572	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Success	EAP
32	5.368580	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Key (Message 1 of 4)	EAPOL
33	5.370502	IntelCor_c8:00:e4	Aerohive_5a:66:d8	Key (Message 2 of 4)	EAPOL
34	5.375048	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Key (Message 3 of 4)	EAPOL
35	5.377421	IntelCor_c8:00:e4	Aerohive_5a:66:d8	Key (Message 4 of 4)	EAPOL
36	5.377692	IntelCor_c8:00:e4	Aerohive_5a:66:d8	Key (Message 4 of 4)	EAPOL
37	5.389740	Aerohive_5a:66:d8	IntelCor_c8:00:e4	Action, SN=258, FN=0,…	802.11

Abbildung 4.15: Verbindungswechsel in einem mit WPA2-Enterprise gesicherten Netzwerk

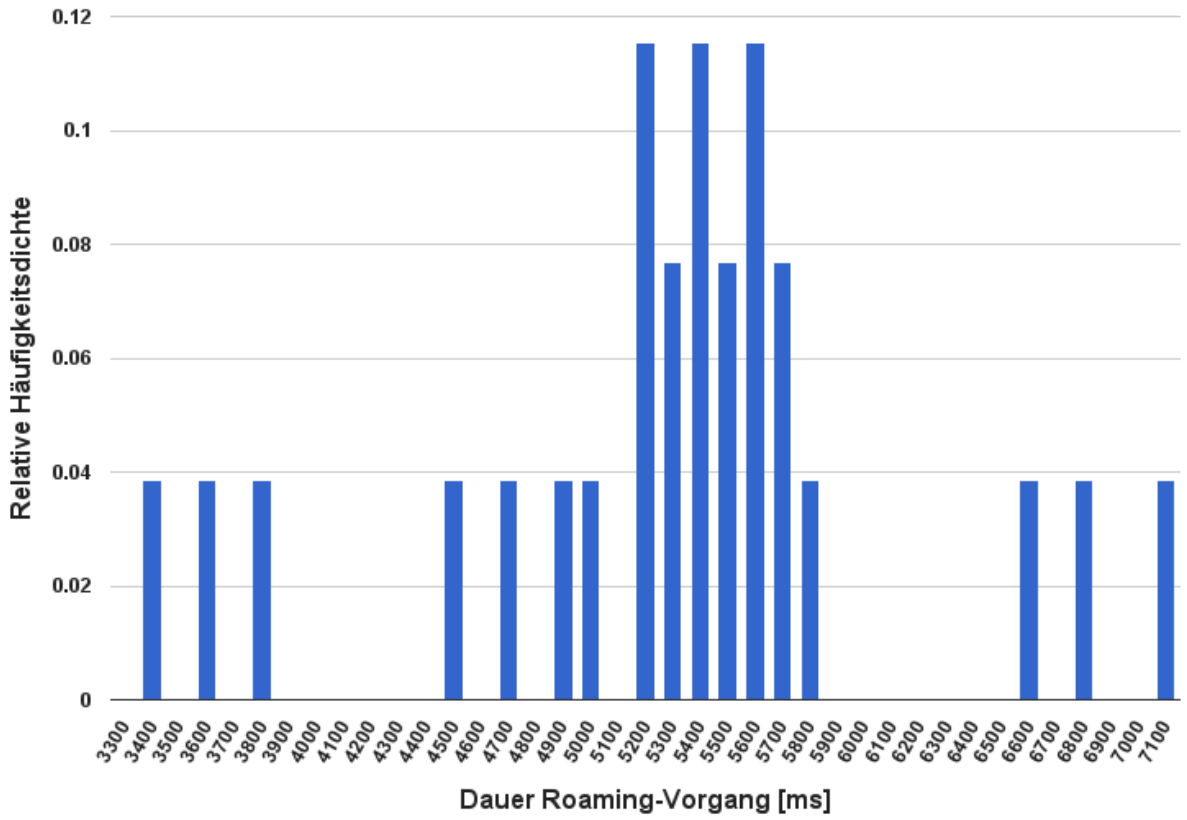

Abbildung 4.16: Relative Häufigkeitsdichte der Dauer eines Roaming-Vorgangs in einem mit WPA2-Enterprise gesicherten Netzwerk

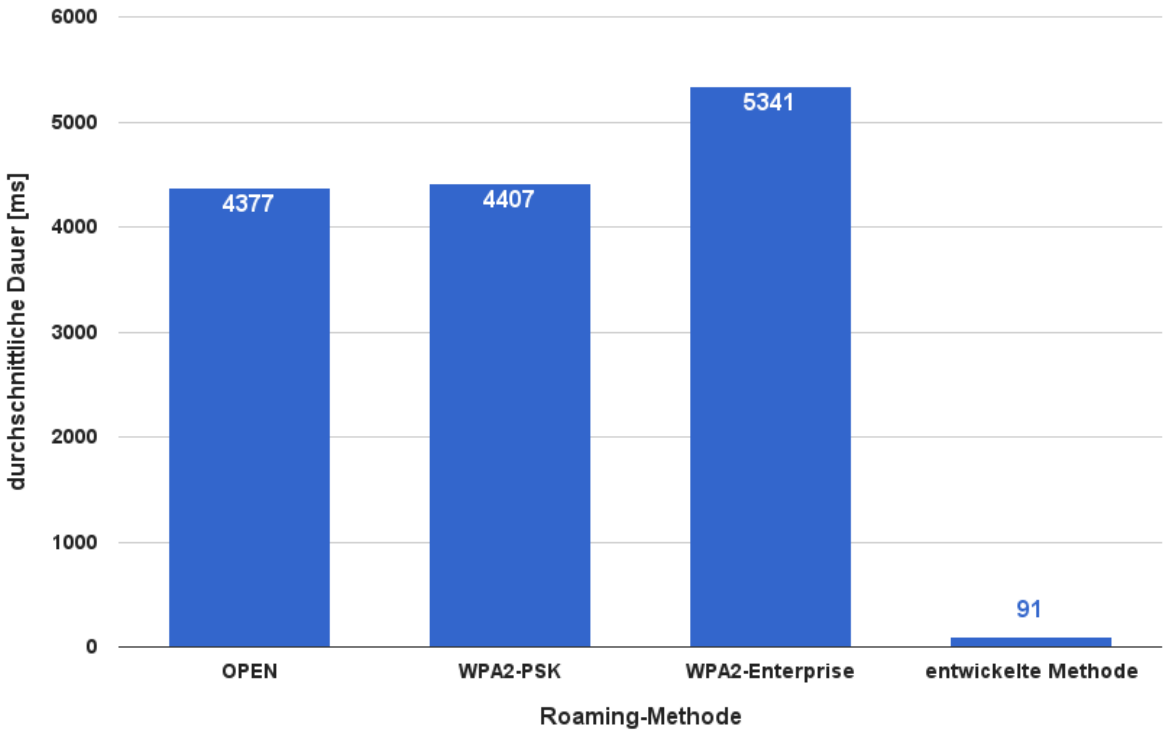

Abbildung 4.17: Vergleich der mittleren Roaming-Dauer unter der Verwendung unterschiedlicher Roaming-Methoden

5 Zusammenfassung und Ausblick

Ziel der Arbeit war der Vergleich von verschiedenen Roaming-Varianten in einem IEEE 802.11 WLAN-Netzwerk. Es sollten dazu bestehende Lösungen verglichen und ein Verfahren gefunden werden, das eine Verbindungsumschaltung in unter 100 ms ermöglicht. Eine weitere Anforderung an das Roaming war die Transparenz hinsichtlich höherer Schichten und Applikationen. Diese dürfen von einem Umschaltvorgang von einem zu einem anderen AP keine Beeinflussung und keine Unterbrechung der Verbindung erfahren. Laufende Datenübertragungen müssen auch nach einem Roaming-Ablauf weiterhin aufrecht bleiben.

Die Suche nach vorhandenen Lösungen ergab größtenteils theoretische Modelle, die auf unterschiedliche Weise das Roaming-Problem in den Griff zu bekommen versuchten. Teilweise wären auch Änderungen seitens der Netzwerkinfrastruktur nötig gewesen. Da es es sich dabei aber meist um nicht standardisierte Protokolle handelte, kamen diese Lösungen für einen weitläufigen Einsatz nicht in Frage.

Die detaillierte Analyse der Abläufe beim Roaming zeigte, dass der Umschaltvorgang grundsätzlich in zwei Phasen geteilt werden kann. Die Optimierung dieser beiden Phasen kann dann weitestgehend getrennt voneinander erfolgen. Dabei ist die Entdeckungsphase jene Phase, in der ein Client einerseits den Roaming-Vorgang auslöst und andererseits die Funkkanäle nach möglichen Roaming-APs scannt. Der zweite Abschnitt ist jener der Authentifizierungsphase. In dieser findet infrastrukturseitig die Überprüfung statt, ob einem Client der Zugang zum Netzwerk gewährt oder verweigert wird. Dieser Abschnitt kann je nach verwendeter Authentifizierungsmethode ebenfalls erhebliche Zeit in Anspruch nehmen.

Als grundlegendes Problem ist jedoch festzuhalten, dass es zwar Protokolle gibt, die bei einem schnelleren Roaming-Vorgang unterstützen können, jedoch dass die endgültige Entscheidung, ob eine Verbindungsumschaltung passieren soll, in keinem Standard festgelegt ist. Dies führt zur Überlegung aufgrund welcher Informationen eine Entscheidung erfolgen kann. Eine Optimierung kann dabei in vielerlei Hinsicht passieren.

Um ein Roaming mit den zuvor vorgestellten Anforderungen zu ermöglichen, wurde deshalb ein Verfahren entwickelt, welches auf der Überwachung der Verbindungsqualität und der regelmäßigen Aggregation von Informationen über mögliche Roaming-Ziele basiert. Als Maß für die Verbindungsqualität wurde der RSSI-Wert herangezogen. Für das Sammeln von Informationen über benachbarte APs kam der Standard IEEE 802.11k zur Anwendung. Konkret kam der in diesem Standard enthaltene Neighbor Report Request, also die Anfrage von einem Client an einen AP, ob dieser Informationen über benachbarte APs hat, zum Einsatz. Das Protokoll IEEE

802.11k bietet somit dem Client die Möglichkeit, sich auf ein mögliches Roaming vorzubereiten. Der besondere Vorteil dabei ist, dass im optimalen Fall ein lange dauernder Scan-Prozess vermieden werden kann. Einzig die Verbreitung von IEEE 802.11k fähiger Hardware lässt vor allem in Linux-Umgebungen noch zu wünschen übrig. Auch die Notwendigkeit, dass der Standard in allen Teilen der Soft- und Hardware unterstützt werden muss, ist sicher ein Punkt, der momentan einem breitflächigen Einsatz im Weg steht. Der Standard würde grundsätzlich jedoch eine gute Grundlage für schnelles und effizientes Roaming in WLAN-Netzwerken bieten, jedoch wird dieser aktuell, mit Ausnahme von Geräten der Firma Apple, kaum genutzt.

Als zweites Protokoll für ein nahtloses Roaming wurde der Standard IEEE 802.11r eingesetzt. Dieser ermöglicht ein schnelleres Roaming durch eine Verkürzung des Authentifizierungsprozesses. Das Verfahren basiert auf einer Vorauthentifizierung und Verteilung der benötigten Schlüssel im Netzwerk. Auch dieses Protokoll muss sowohl client- als auch infrastrukturseitig unterstützt werden. Im Gegensatz zum Protokoll IEEE 802.11k wird jedoch der Standard 802.11r von vielen Netzwerkequipment-Herstellern zur Verfügung gestellt. Auch der Support durch das Betriebssystem ist in linuxbasierten Systemen durch die Applikation wpa_supplicant gegeben.

Um die Funktion und Performance der entwickelten Roaming-Methode messen und mit bestehenden Lösungen vergleichen zu können, wurde ein Testaufbau bestehend aus einem Client und zwei APs entwickelt. Die Implementierung des Roaming-Algorithmus erfolgte dabei am Client. Dazu wurden diverse Softwareteile rund um die Applikation wpa_supplicant auf einem Linux-Rechner geändert und erweitert. Die abschließenden Messungen ergaben, dass mit dem entwickelten Algorithmus eine Verbindungsumschaltung im Durchschnitt von 91 ms möglich ist. Ein besonderer Vorteil der implementieren Lösung ist, dass das Roaming dabei transparent für höhere Schichten abläuft. Das bedeutet, dass die Assoziation zu einem Netzwerk bei einer Verbindungsumschaltung zu keinem Zeitpunkt verloren geht. Darüber hinaus muss trotz einer guten Roaming-Performance nicht auf eine starke Sicherung der Netzwerkverbindung verzichtet werden. Als Authentifizierungsmethode wurde der Standard WPA2-Enterprise mit einem RADIUS-Server eingesetzt.

Der Vergleich zu bestehenden Lösungen zeigte generell, dass eine echte Verbindungsumschaltung, die transparent für höhere Schichten abläuft, bei herkömmlichen Netzwerkkonfigurationen nicht möglich ist. Lediglich eine Neuverbindung nach dem Abbruch einer Verbindung war möglich. Die Dauer für diese Abläufe belief sich in allen gemessenen Varianten auf mehr als 2.5 s, da in jedem Fall ein kompletter Scan der WLAN-Kanäle durchgeführt werden musste. Selbst die Verwendung eines ungesicherten Netzwerkes konnte hierfür keine bessere Performance erbringen. Nahtloses Roaming ist somit bei herkömmlichen Varianten nicht möglich.

Alles in allem wurde in dieser Arbeit gezeigt, dass selbst mit einem recht simplen Modell sehr gute Roaming-Zeiten erreichbar sind. Verbesserungspotential für die entwickelte Lösung ist jedoch an vielen Stellen vorhanden. Besonders jener Teil der Triggerung des Roamings bietet weitreichende Möglichkeiten. In der Realisierung wurden als Auslöser starre Grenzwerte definiert. Sobald diese überschritten wurden, erfolgte eine Verbindungsumschaltung. Diese starren Grenzen könnten durch die Verwendung von Variablen einen weiteren Performance-Schub bringen. So wäre es beispielsweise denkbar die RSSI-Schwellwerte der aktuellen Verbindung, ab der ein Roaming ausgelöst wird, variabel zu gestalten. Dies könnte je nach Qualität der vorhandenen Netzwerkinfrastruktur passieren. Beispielsweise könnte eine Korrektur der Schwelle nach oben erfolgen, wenn in der vorhandenen Funkumgebung mehrere mögliche APs mit guter Anbindung zu Verfügung stehen. Dadurch würde gewährleistet werden, dass der Client immer versucht zum optimalen AP zu roamen.

Auch die Frequenz mit der Neighbor Report Requests ausgesendet werden, könnte flexibel gestaltet werden. Beispielsweise könnte eine Kopplung an die Bewegungsgeschwindigkeit des Clients gemacht werden. Ein Client, der sich schnell bewegt, sollte die Frequenz dementsprechend erhöhen um der Dynamik der Funkumgebung folgen zu können. Im Gegensatz könnte das Intervall für die Aussendung von Neighbor Report Requests bei einem nicht oder nur langsam bewegten Client dementsprechend erhöht werden, ohne dass diesem dadurch Änderungen des Netzwerks entgehen würden. Dies würde darüber hinaus auch eine Einsparung von Nachrichten, die im laufenden Nachrichtenverkehr gesendet werden müssen, zur Folge haben.

Außerdem beschränkt sich die Roaming-Lösung auf den RSSI-Wert als Maß für die Verbindungsqualtität. Jedoch muss ein hoher RSSI nicht gleichbedeutend mit einer guten Anbindung an das Netzwerk sein. Ein AP, zu dem eine vermeintlich gute Verbindung, also ein hoher RSSI besteht, muss nicht gleichbedeutend mit einer optimalen Anbindung an das Netzwerk sein. Generell kann die Optimierung des Roamings anhand vieler Parameter erfolgen. Die Nutzung des RSSI ist nur eine mögliche Lösung. Eine zusätzliche Performance-Steigerung kann mit Sicherheit dadurch erreicht werden, wenn eine Optimierung der Verbindungsumschaltung in Bezug auf laufende Anwendungen passiert. So können je nach Anwendungsumfeld auch die Anforderungen durchwegs unterschiedlich sein. Beispielsweise könnte die Verbesserung des Roamings für den Betrieb in einem Büroumfeld eine andere sein, wie für den Einsatz zur Steuerung von Robotern.

Abschließend soll auch noch angemerkt werden, dass die implementierte Roaming-Methode nur eine Optimierung aus Sicht eines einzelnen Clients ermöglicht. Ob eine Roaming-Entscheidung die ein Client trifft, auch für das Netzwerk insgesamt positiv ist, kann aus Client-Sicht nicht beantwortet werden. Zu dessen Beurteilung fehlen dem Client die Möglichkeiten. Um also zum Beispiel eine Optimierung hinsichtlich einer effizienten Lastverteilung in Netzwerk erreichen zu können, muss das Netzwerk infrastrukturseitig in der Lage sein, dem Client solche Parameter zu übermitteln. Der Standard IEEE 802.11v (Wireless Network Management) geht in diese Richtung. Dazu wird unter anderem ebenfalls der in dieser Arbeit verwendete Standard IEEE 802.11k verwendet.

Literatur

[ABV+04] Aboba, B. ; Blunk, L. ; Vollbrecht, J. ; Carlson, J. ; Levkowetz, H: RFC 3748 - Extensible Authentication Protocol (EAP), 2004

[AH08] Ahmed, Hassan ; Hassanein, Hossam: A Performance Study of Roaming in Wireless Local Area Networks Based on IEEE 802.11r. In: *24th Biennial Symposium on Communications*, 2008. – ISBN 978–1–4244–1945–6, S. 256

[AKT08] Athanasiou, George ; Korakis, Thanasis ; Tassiulas, Leandros: Cooperative Handoff in Wireless Networks. In: *IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008.*, 2008. – ISBN 978–1–4244–2643–0, S. 1–6

[AN15] Aerohive Networks, Inc.: Datasheet Aerohive AP141 - 802.11n Dual Radio Access Point, 2015

[BHL06] Bittau, Andrea ; Handley, Mark ; Lackey, Joshua: The Final Nail in WEP's Coffin. In: *IEEE Symposium on Security and Privacy 2006*, 2006. – ISBN 0–7695–2574–1, S. 399

[BK12] Balažia, Ján ; Kotuliak, Ivan: Seamless Handover in 802.11 Networks. In: *Wireless and Mobile Networking Conference (WMNC), 2012 5th Joint IFIP*, 2012. – ISBN 978–1–4673–2993–4, S. 127

[CC15] Chen, Chia-Mei ; Chang, Tien-Ho: The Cryptanalysis of WPA and WPA2 in the Rule-Based Brute Force Attack, an Advanced and Efficient Method. In: *10th Asia Joint Conference on Information Security (AsiaJCIS)*, 2015, S. 38

[Cor09] Corporation, Intel: wpa_supplicant Developer Documentation, 2009

[Cor11] Corporation, Intel: Datasheet Intel Centrino Ultimate-N6300, 2011

[DS14] Dutta, Ashutosh ; Schulzrinne, Henning: *Mobility Protocols and Handover Optimization: Design, Evaluation and Application.* Wiley, 2014. – 167 S. – ISBN 978–0–470–74058–3

[GME11] Georgopoulos, Panagiotis ; McCarthy, Ben ; Edwards, Christopher: A collaborative AAA Architecture to enable secure real-World Network Mobility. In: *NETWORKING 2011, 10th International IFIP TC 6 Networking Conference*, 2011. – ISBN 978–3–642–20756–3, S. 216

[IT03] ITU-T: ITU-T Recommendation G.114: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS - International Telephone Connections and Circuits – General Recommendations on the Transmission Quality for an entire international Telephone Connection - One-Way Transmission Time, 2003

[LC06] Liao, Yong ; Cao, Lixin: Practical Schemes for Smooth MAC Layer Handoff in 802.11 Wireless Networks. In: *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks (WOWMOM '06)*, 2006. – ISBN 0–7695–2593–8, S. 191

[MDK07] Murray, David ; Dixon, Michael ; Koziniec, Terry: Scanning Delays in 802.11 Networks. In: *The 2007 International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST 2007)*, 2007

[MSJ+04] Mishra, A. ; Shin, Min H. ; Jr., N.L. P. ; Clancy, T.C. ; Arbaugh, W.A.: Proactive Key Distribution using Neighbor Graphs. In: *Wireless Communications, IEEE (Volume:11 , Issue: 1)*, 2004, S. 26–36

[MW12] Machań, P. ; Wozniak, J.: A Lightweight Algorithm for Fast IEEE 802.11 Handover. In: *Telecommunication Networks and Applications Conference (ATNAC), 2012 Australasian*, 2012. – ISBN 978–1–4673–4408–1, S. 1, 5

[Nag] von Nagy, Andrew: Whitepaper - Voice Enterprise Certification - Standardized Fast Secure Roaming. In: *Voice Enterprise Certification*, S. 5–7

[PCKC07a] Pack, S. ; Choi, Jaeyoung ; Kwon, Taekyoung ; Choi, Yanghee: Fast-Handoff Support in IEEE 802.11 Wireless Networks. In: *Communications Surveys and Tutorials, IEEE*, 2007, S. 3

[PCKC07b] Pack, Sangheon ; Choi, Jaeyoung ; Kwon, Taekyoung ; Choi, Yanghee: Fast-Handoff Support in IEEE 802.11 Wireless Networks. In: *IEEE Communications Surveys and Tutorials, vol.9, no.1*, 2007, S. 1–25

[PJKC05] Pack, Sangheon ; Jung, Hakyung ; Kwon, Taekyoung ; Choi, Yanghee: A selective Neighbor Caching Scheme for fast Handoff in IEEE 802.11 Wireless Networks. In: *IEEE International Conference on Communications, 2005 (Volume:5)*, 2005. – ISBN 0–7803–8938–7, S. 3599–3603

[RS05] Ramani, Ishwar ; Savage, Stefan: SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks. In: *24th Annual Joint Conference of the IEEE Computer and Communications Societies. INFOCOM 2005.*, 2005. – ISBN 0–7803–8968–9, S. 675–684

[SFRS04] Shin, Sangho ; Forte, Andrea G. ; Rawat, Anshuman S. ; Schulzrinne, Henning: Reducing MAC Layer Handoff Latency in IEEE 802.11 Wireless LANs. In: *MobiWac 04 Proceedings of the second international Workshop on Mobility Management and Wireless Access Protocols*, 2004. – ISBN 1–58113–920–9, S. 19–26

[SJB09] Sauter, Thilo ; Jasperneite, Jýrgen ; Bello, Lucia L.: Towards New Hybrid Networks for Industrial Automation. In: *IEEE Conference on Emerging Technologies and Factory Automation, 2009. ETFA 2009*, 2009, S. 1–8

[SMA04] Shin, Minho ; Mishra, Arunesh ; Arbaugh, William A.: Improving the Latency of 802.11 Hand-offs using Neighbor Graphs. In: *Processings of the ACM MobiSys Conference, Boston, USA*, 2004, S. 71

[Soc08a] Society, IEEE C.: IEEE Standard for Information technology - Local and metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Fast Basic Service Set (BSS) Transition. In: *IEEE Std 802.11r-2008 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008)*, 2008, S. 40, 44

[Soc08b] Society, IEEE C.: IEEE Standard for Information Technology - Local and metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 1: Radio Resource Measurement of Wireless LANs. In: *IEEE Std 802.11kTM-2008 (Amendment to IEEE Std 802.11TM-2007*, 2008, S. 4–6

[Soc12] Society, IEEE C.: 802.11-2012 - IEEE Standard for Information technology - Telecommunications and Information Exchange between Systems local and metropolitan Area Networks - Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. In: *IEEE Std 802.11-2012*, 2012, S. 1163, 1227, 1234, 1214

[Sys16] Systems, Cisco: Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016, S. 21 – 23

[TTHJ09] Tabassam, Ahmad A. ; Trsek, Henning ; Heiss, Stefan ; Jasperneite, Jürgen: Fast and Seamless Handover for Secure Mobile Industrial Applications with 802.11r. In: *IEEE 34th Conference on Local Computer Networks, LCN 2009*, 2009. – ISBN 978–1–4244–4488–5, S. 751, 754, 756

[Var12] Variscite: VAR-MX6CustomBoard Rev. 1.1 Datasheet, 2012, S. 9

[wif12] Wi-Fi CERTIFIED WPA2 Delivers Advanced Security to Homes, Enterprises and Mobile Devices. In: *The State of Wi-Fi® Security*, 2012, S. 6

[YMW08] Yu, Ching-Hwa ; M., Pan ; Wang, Sheng-De: Adaptive Neighbor Caching for Fast BSS Transition Using IEEE 802.11k Neighbor Report. In: *International Symposium on Parallel and Distributed Processing with Applications, 2008*, 2008. – ISBN 978–0–7695–3471–8, S. 353–360

[ZWB05] Zan, Lei ; Wang, Jidong ; Bao, Lichun: Personal AP Protocol for Mobility Management in IEEE 802.11 Systems. In: *The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services. MobiQuitous 2005.*, 2005. – ISBN 0–7695–2375–7, S. 420

Internet Referenzen

[1] Cisco. *802.11 WLAN Roaming and Fast-Secure Roaming on Cisco Unified Wireless Network*, November 2015. http://www.cisco.com/c/en/us/support/docs/wireless-mobility/wireless-lan-wlan/116493-technote-technology-00.html.

[2] Cisco Systems, Inc. *Cisco Radio Frequency Site Survey*, August 2016. http://www.cisco.com/c/en/us/support/docs/wireless-mobility/wireless-lan-wlan/68666-wireless-site-survey-faq.html#q26.

[3] ESnet, Lawrence Berkeley National Laboratory. *iPerf - The Network Bandwidth Measurement Tool*, August 2016. https://iperf.fr/.

[4] Linux Kernel. *Linux Wireless Kernel Documentation*, January 2015. https://wireless.wiki.kernel.org/en/users/drivers/ath10k/architecture.

[5] Revolution Wifi. *Revolution Wifi - Cisco Centralized Key Management*, November 2015. http://www.revolutionwifi.net/revolutionwifi/2012/02/wi-fi-roaming-analysis-part-2-roaming.html.

[6] Statistik Austria. *Beschäftigte in Unternehmen, die mit tragbaren Geräten mit mobilem Internetzugang ausgestattet wurden, 2014*, Oktober 2014. http://www.statistik.at/web_de/statistiken/energie_umwelt_innovation_mobilitaet/informationsgesellschaft/ikt-einsatz_in_unternehmen/022199.html.

[7] Wi-Fi Alliance. *Product Finder - Voice Enterprise Certification*, September 2016. http://www.wi-fi.org/product-finder-results.

[8] wpa_supplicant Documentation. *Documentation wpa_supplicant control Interface*, August 2016. https://w1.fi/wpa_supplicant/devel/ctrl_iface_page.html.