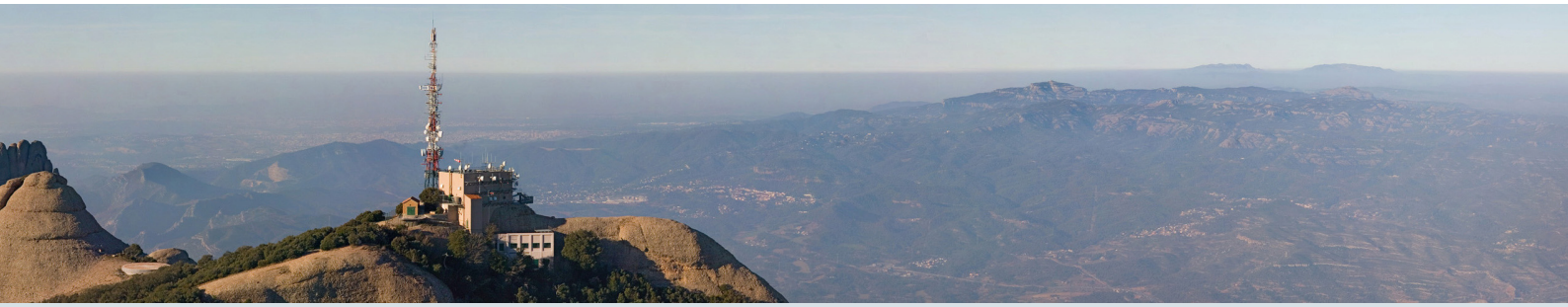
ulm university universität ulm



Diplomarbeit

Sicherheit der PHY- und MAC-Schicht in 802.11-Netzwerken

Bastian Könings

Universität Ulm
Institut für Medieninformatik

vorgelegt am 13. Januar 2009

Gutachter

Prof. Dr. Michael Weber
Dr. Frank Kargl

ulm university universität
uulm

Diplomarbeit
im Studiengang Medieninformatik

Sicherheit der PHY- und MAC-Schicht in 802.11-Netzwerken

Bastian Könings

Universität Ulm

Fakultät für Ingenieurwissenschaften und Informatik

Institut für Medieninformatik

vorgelegt am
13. Januar 2009

Gutachter
Prof. Dr. Michael Weber
Dr. Frank Kargl

Fassung vom 16. Januar 2009

Titelbild: *Montserrat-Gebirge in Katalonien* von David Iliff
http://en.wikipedia.org/wiki/User:Diliff

Inhaltsverzeichnis

Abkürzungsverzeichnis

A	Authenticator
AA	Authenticator Address
AC	Access Category
ACK	Acknowledgment
ADDBA	Add Block Acknowledgment
AES	Advanced Encryption Standard
AID	Association Identifier
AIFS	Arbitration Interframe Space
AIFSN	Arbitration Interframe Space Number
ANonce	Authenticator Nonce
AODV	Ad-hoc On-demand Distance Vector
AP	Access Point
ARF	Authentication Request Flood
ARP	Address Resolution Protocol
ASRF	Association Request Flood
ATIM	Announcement Traffic Indication Message
BPSK	Binary Phase Shift Keying
BSS	Basic Service Set
BSSID	Basic Service Set Identification
CA	Collision Avoidance
CAA	Colluding Adversaries Attack
CAP	Controlled Access Phase
CARP	Commodity Atheros Research Platform
CBC-MAC	Cipher-Block Chaining Message Authentication Code
CCA	Clear Channel Assessment
CCM	Counter Mode with CBC-MAC
CCMP	Counter Mode with CBC-MAC Protocol
CF	Contention Free
CFP	Contention Free Period
CP	Contention Period
CRC	Cyclic Redundancy Check
CS	Carrier Sense
CSMA	Carrier Sense Multiple Access
CTR	Counter Mode
CTS	Clear To Send
CVE	Common Vulnerabilities and Exposures
CW	Contention Window
DA	Destination Address
DCF	Distributed Coordination Function
DELBA	Delete Block Acknowledgment

DFS	Dynamic Frequency Selection
DIFS	Distributed (Coordination Function) Interframe Space
DLS	Direct Link Setup
DoS	Denial of Service
DS	Distribution System
DSSS	Direct Sequence Spread Spectrum
EAP	Extensible Authentication Protocol
EAPOL	Extensible Authentication Protocol over LANs
EDCA	Enhanced Distributed Channel Access
EIFS	Extended Interframe Space
EN	Euro Norm
ESS	Extended Service Set
ETSI	European Telecommunications Standards Institute
EU	European Union
FCS	Frame Check Sequence
FDMA	Frequency Division Multiple Access
FHSS	Frequency Hopping Spread Spectrum
FPGA	Field Programmable Gate Array
GSM	Global System for Mobile Communications
GTK	Group Temporal Key
HCCA	HCF Controlled Channel Access
HC	Hybrid Coordinator
HCF	Hybrid Coordination Function
HEC	Header Error Check
IBSS	Independent BSS
ICMP	Internet Control Message Protocol
ICV	Integrity Check Value
IDS	Intrusion Detection System
IE	Information Element
IEEE	Institute of Electrical and Electronics Engineers
IETF	The Internet Engineering Task Force
IFS	Interframe Space
ISM	Industrial, Scientific and Medical
ISO	International Organization for Standardization
IV	Initialization Vector
LAN	Local Area Network
LLC	Logical Link Control
MAC	Medium Access Control
MANET	Mobile Ad Hoc Network
MIC	Message Integrity Code
MIMO	Multiple Input Multiple Output
MitM	Man in the Middle
MLME	MAC Sublayer Management Entity
MoKB	Month of Kernel Bugs
MPDU	MAC Protocol Data Unit
MSDU	MAC service data unit
MSK	Master Session Key
NAV	Network Allocation Vector

NIC	Network Interface Card
OFDM	Orthogonal Frequency Division Multiplexing
OLPC	One Laptop Per Child
OLSR	Optimized Link State Routing
OSI	Open Systems Interconnection
PC	Point Coordinator
PCF	Point Coordination Function
PDA	Personal Digital Assistant
PDR	Packet Delivery Ratio
PHY	Physical Layer
PIFS	Point (Coordination Function) Interframe Space
PIM	Pseudo-IBSS Mode
PLCP	Physical Layer Convergence Procedure
PLME	Physical Layer Management Entity
PMD	Physical Medium Dependent
PMK	Pairwise Master Key
PPDU	PLCP Protocol Data Unit
PPM	Pulse Position Modulation
PRF	Probe Request Flood
PS	Power Save (Mode)
PSK	Preshared Key
PSR	Packet Send Ratio
PTK	Pairwise Transient Key
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RA	Receiving Station Address
RADIUS	Remote Authentication Dial-In User Service
RF	Radio Frequency
RSN	Robust Security Network
RSNA	Robust Security Network Association
RTS	Request To Send
SA	Source Address
SAP	Service Access Point
SDM	Spatial Division Multiplexing
SDR	Software Defined Radio
SIFS	Short Interframe Space
SFD	Start Frame Delimiter
SLRC	Station Long Retry Count
SME	Station Management Entity
SN	Sequence Number
SPNonce	Supplicant Nonce
SP	Supplicant
SPA	Supplicant Address
SSID	Service Set Identifier
SSRC	Station Short Retry Count
STA	Station
STBC	Space Time Block Coding

TA	Transmitting Station Address
TBTT	Target Beacon Transmission Time
TCP	Transmission Control Protocol
TID	Traffic Identifier
TIM	Traffic Indication Map
TKIP	Temporal Key Integrity Protocol
TLS	Transport Layer Security
TPC	Transmit Power Control
TSC	TKIP Sequence Counter
TSF	Timing Synchronization Function
TSPEC	Traffic Specification
TU	Time Unit
TXOP	Transmission Opportunity
UDP	User Datagram Protocol
USB	Universal Serial Bus
USRP	Universal Software Radio Peripheral
VANET	Vehicular Ad Hoc Network
VAP	Virtual Access Point
VoWLAN	Voice over Wireless LAN
WAVE	Wireless Access in Vehicular Environments
WBSS	WAVE BSS
WDS	Wireless Distribution System
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network
WMN	Wireless Mesh Network
WPA	Wi-Fi Protected Access
WVE	Wireless Vulnerabilities & Exploits
XOR	Exclusive Or

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Kabellose lokale Netzwerke basierend auf dem IEEE 802.11-Standard haben in den letzten Jahren enorm an Popularität gewonnen. Derartige Netzwerke, auch bekannt als WLANs, sind heute sowohl in privaten, gewerblichen als auch öffentlichen Bereichen gleichermaßen vertreten und bieten eine kostengünstige, flexible und vor allem leicht zu integrierende Möglichkeit der kabellosen Datenkommunikation. Ein sehr beliebter und weit verbreiteter Einsatzbereich von WLANs sind öffentliche Zugangspunkte zum Internet, sogenannte Hotspots, die meist gegen Bezahlung für jedermann zugänglich sind. Unternehmen wie FON[1] oder WiFiTastic[2] haben sich zum Ziel gesetzt private WLANs in öffentliche Hotspots umzufunktionieren, um somit die weltweite Abdeckung zu erhöhen [69]. Innerhalb der Yamanote Zug-Ringstrecke, eine der meist genutzten Pendlerstrecken Tokios, konnte FON bisher schon eine Abdeckung von 80 Prozent erreichen. Laut einer Studie des Marktforschungsunternehmens ABI Research [2] soll die Anzahl öffentlicher WLAN-Hotspots weltweit bis zum Ende des Jahres 2008 um 40 Prozent im Vergleich zum Vorjahr steigen.

Die Studie prognostiziert weiterhin für das Jahr 2011 einen jährlichen Verkauf von 250 Millionen Geräten aus dem Bereich der Unterhaltungselektronik und über 360 Millionen sonstiger mobiler Geräte, die mit WLAN-Komponenten ausgestattet sind. Dies zeigt, dass WLAN nicht mehr nur für Laptops oder PDAs interessant ist, sondern auch viele neue Anwendungsgebiete im Bereich Multimedia oder Telekommunikation entstehen lässt, die über das einfache Surfen im Internet hinausgehen. So existieren bereits Kameras, Fernseher, MP3- und DVD-Spieler mit integrierter WLAN-Schnittstelle. Spezielle Telefone ermöglichen per *Voice over Wireless LAN* (VoWLAN) das kostenlose Telefonieren über das Internet. Auch im Mobilfunkbereich existieren derzeit schon mehr als 100 Dual-Mode Mobiltelefone, die neben herkömmlichen Mobilfunktechniken wie GSM auch WLAN-Verbindungen unterstützen.

Ein weiteres wichtiges Einsatzgebiet, das besonders für zukünftige Anwendungsfelder immer interessanter wird, sind mobile Ad-hoc-Netze (MANETs) und kabellose Mesh-Netze (WMNs). Viele der aktuellen Forschungsprojekte, die sich mit Inter-Fahrzeugkommunikation und somit einer Spezialform von MANETs beschäftigen, setzen ebenfalls auf WLAN als Übertragungstechnik [128, 66, 90]. Mesh-Netze basierend auf WLAN sind besonders beliebt, um in schlecht erreichbaren Gebieten mit mangelnder Infrastruktur eine Zugangsmöglichkeit zum Internet zu schaffen. Eines der bekanntesten Projekte, bei dem Mesh-Netze dieser Art zum Einsatz kommen, ist das OLPC[3] Projekt. Aber auch viele andere Projekte basieren auf Mesh-Netzen, wie beispielsweise die nichtkommerzielle Initiative Freifunk[4]. Freifunk verfolgt das Ziel, in Deutschland ein möglichst großflächiges Mesh-Netz für den kostenlosen Internetzugang zu etablieren. In Berlin stehen Mitgliedern der Freifunk-Community beispielsweise schon mehr als 500 Zugangspunkte zur Verfügung.

[1] http://www.fon.com
[2] http://www.wifitastic.com
[3] http://laptop.org
[4] http://www.freifunk.net

Bei der zunehmenden Verbreitung und Bedeutung von kabellosen lokalen Netzwerken, insbesondere in kritischen Anwendungsbereichen, spielt die Sicherheit eine immer wichtigere Rolle. Diese zu gewährleisten stellt aber speziell bei kabellosen Netzwerken eine große Herausforderung dar. Die Sicherheitsaspekte Vertraulichkeit, Integrität und Authentizität können unter Verwendung bereits existierender Sicherheitsmechanismen meist angemessen garantiert werden (siehe Abschnitt 2.6). Wesentlich schwerer ist hingegen die Verfügbarkeit eines kabellosen Netzes zu garantieren. Da kabellose Netze über keine physische Abschirmung nach außen verfügen, wie es etwa bei kabelgebundenen Netzen der Fall ist, kann jeder, der sich in Reichweite des kabellosen Netzes befindet, darauf zugreifen. Jegliche Kommunikation kann somit von potentiellen Angreifern mitverfolgt werden. Auch wenn der Großteil der dabei übertragenen Daten durch Verschlüsselungsverfahren zunächst nicht eingesehen werden kann, hat ein Angreifer dennoch die Möglichkeit, verschiedene Angriffe durchzuführen. Ein möglicher Angriff ist die Kommunikation mit einem Funksignal zu stören oder gänzlich zu unterbrechen. Komplexere Angriffe nutzen Schwachstellen der 802.11-Mechanismen für die Zugriffskontrolle aus, um ähnliche Ziele oder auch eigene Vorteile wie einen höheren Datendurchsatz zu erreichen.

Die Verfügbarkeit eines kabellosen Netzes ist, abhängig von dessen Einsatzgebiet, eine der wichtigsten Sicherheitsanforderungen. Bei den heutigen, meist auf Infrastruktur basierenden privaten WLANs oder Hotspots bedeutet der Verlust der Verfügbarkeit konkret die Unterbrechung einer Internetverbindung, oder bei WLAN-fähigen Telefonen die Unterbrechung eines Telefongespräches. Geht man aber von kritischeren Anwendungsbereichen wie dem Gesundheitswesen aus, so kann der Verlust der Verfügbarkeit ernsthafte Gefahren darstellen. Schon heute hat sich der Einsatz von WLANs in deutschen Krankenhäusern fest etabliert [39]. Wird also hier beispielsweise die WLAN-Verbindung eines medizinischen Gerätes unterbrochen, sind fatale Konsequenzen denkbar. Auch bei mobilen Ad-hoc-Netzen und kabellosen Mesh-Netzen kann die Kompromittierung der Verfügbarkeit größere Auswirkungen haben. Da Netze dieser Art über keine feste Infrastruktur verfügen und somit die Daten von Knoten zu Knoten weitergereicht werden müssen, kann bei ungünstiger Knotentopologie der Ausfall eines einzelnen Knoten den Zusammenbruch des gesamten Netzes bedeuten. Bei dem konkreten Beispiel eines kabellosen Ad-hoc-Fahrzeugnetzes (VANET) bedeutet dies, dass Daten wie Verkehrsinformationen, aber auch kritische Warnmeldungen, nicht weitergereicht werden können. In dem Bereich der MANETs sind viele Anwendungen noch in der Entstehungsphase und daher ist eine grundlegende Bewertung des Sicherheitsaspekts Verfügbarkeit eine umso wichtigere Aufgabe und Herausforderung. Da die meisten bereits existierenden kabellosen Netze, aber auch viele der neu entstehenden Systeme, auf dem IEEE 802.11-Standard basieren werden, sind Angriffe, die Sicherheitslücken in diesem Standard ausnutzen, als hohe Bedrohung einzustufen.

Der Fokus dieser Diplomarbeit liegt daher auf der Sichtung existierender und der Aufdeckung neuer Angriffsmöglichkeiten gegen die Verfügbarkeit von WLANs, deren Umsetzbarkeit und Auswirkungen anhand realer Tests analysiert werden sollen. Kapitel 2 erläutert die nötigen Grundlagen und Mechanismen des 802.11-Standards, die insbesondere für die in Kapitel 3 diskutierten Angriffe gegen die Verfügbarkeit relevant sind. In Kapitel 4 wird die Umsetzung von drei ausgewählten Angriffen und der Aufbau der Testumgebung beschrieben, in der die jeweiligen Angriffe untersucht wurden. Die Ergebnisse der Tests und eine anschließende Bewertung werden in Kapitel 5 vorgestellt. Kapitel 6 fasst den Inhalt sowie die Ergebnisse dieser Arbeit zusammen und bietet einen kurzen Ausblick auf die zukünftige Entwicklung.

2 Der IEEE 802.11-Standard

Das folgende Kapitel gibt einen grundlegenden Überblick über die Einordnung, Architektur und Funktionsweise der PHY- und MAC-Schicht sowie existierende Sicherheitsmechanismen des aktuellen IEEE 802.11-Standards [60]. Insbesondere werden Erweiterungen des Standards vorgestellt, die für die in Kapitel 3 beschriebenen Angriffe gegen die Verfügbarkeit relevant sind.

2.1 Einordnung

Nachdem Anfang der 80er Jahre eine enorme Vielfalt an unterschiedlichen LANs in Bezug auf Übertragungstechnik, Bandbreite und Zugangsverfahren vorhanden war, wurde die Arbeitsgruppe 802 des *Institute of Electrical and Electronics Engineers* (IEEE) gegründet, die die verschiedenen Verfahren für lokale Netzwerke standardisieren sollte. Ein Großteil der 802.x Standards beschränkt sich auf die beiden unteren Schichten des ISO/OSI Referenzmodells [131], die Bitübertragungsschicht (*Physical Layer*) und die Sicherungsschicht (*Data Link Layer*). Abbildung 2.1 zeigt eine Übersicht der wichtigsten 802.x Standards und ihre Einordnung in das ISO/OSI Referenzmodell.

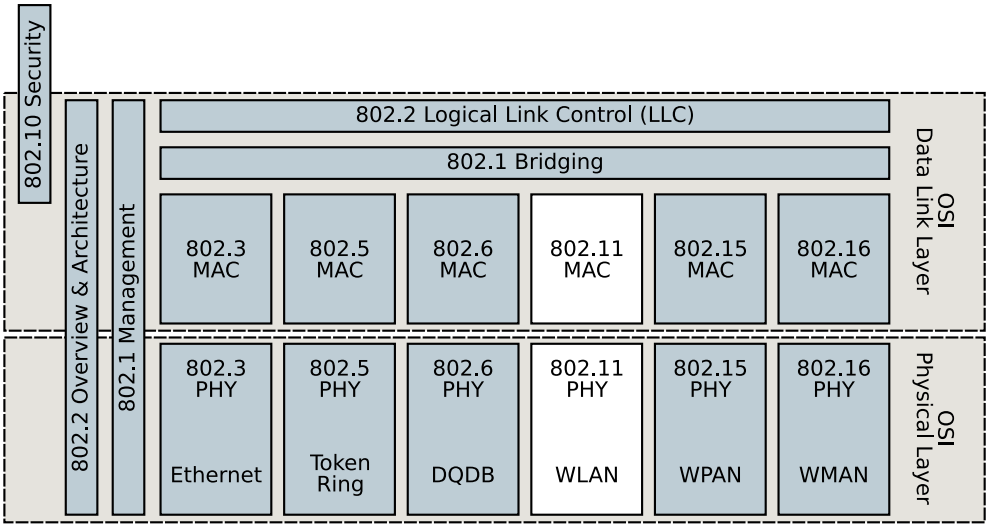


Abbildung 2.1: Übersicht der wichtigsten IEEE 802.x Standards nach [107]

Aus dem ersten Standard 802.11 für kabellose LANs [55], der 1997 nach einer etwa sieben Jahre dauernden Entwicklungs- und Genehmigungsphase durch das IEEE verabschiedet wurde, sind die drei Erweiterungen a, b und g hervorgegangen, auf denen die meisten der heute verfügbaren WLAN-Geräte basieren. Um die Interoperabilität zwischen Geräten unterschiedlicher Hersteller zu gewährleisten, wurde 1999 aus zahlreichen Unternehmen die *Wireless Ethernet Compatibility Al-*

liance[1] (*Wi-Fi Alliance*) gegründet, welche Produkte auf Basis der verschiedenen 802.11-Standards zertifiziert. Da dieses Zertifizierungsprogramm sich sehr schnell international durchsetzen konnte, war *Wi-Fi* schon bald ein Synonym für WLAN und wird heute besonders im englischsprachigen Raum bevorzugt verwendet. Seit Verabschiedung des ersten Standards sind zahlreiche Erweiterungen hinzugekommen, die sich hauptsächlich mit den Aspekten Sicherheit (802.11i), *Quality of Service* (802.11e) und Erhöhung der Bandbreite (802.11n) beschäftigen. Aber auch im Bereich der bereits erwähnten VANETs gibt es eine IEEE Arbeitsgruppe, die ihre Arbeit speziell auf Erweiterungen für die Kommunikation zwischen Fahrzeugen richtet (802.11p). Tabelle 2.1 gibt eine Übersicht aller Erweiterungen des ursprünglichen 802.11-Standards, beziehungsweise der entsprechenden IEEE Arbeitsgruppen, sowie deren Relevanz für den Inhalt dieser Arbeit. Seit 2007 existiert eine aktualisierte Version des 802.11-Standards, in der schon die Erweiterungen a, b, d, e, g, h, i und j eingeflossen sind [60].

IEEE Standard	Verabschiedet	Beschreibung
802.11	1997	erster Standard für Datenraten bis zu 2 Mbps im 2,4 GHz Band, spezifiziert PHY- und MAC-Schicht
802.11a	1999	physikalische Erweiterung für Datenraten bis zu 54 Mbps im 5 GHz Band basierend auf OFDM
802.11b	1999	physikalische Erweiterung für Datenraten bis zu 11 Mbps im 2,4 GHz Band basierend auf DSSS
802.11c	1998	spezifiziert Bridging auf MAC-Ebene
802.11d	2001	Anpassung für unterschiedliche Frequenznutzungsvorschriften
802.11e	2005	Erweiterungen für Quality Of Service (QoS)
802.11F	2003	Einführung eines Inter Access Point Protocols (IAPP)
802.11g	2003	physikalische Erweiterung für Datenraten bis zu 54 Mbps im 2,4 GHz Band basierend auf DSSS oder OFDM
802.11h	2003	Frequenzspektrum-Management von 802.11a für Einsatz in Europa
802.11i	2004	Erweiterungen von Sicherheitsmechanismen auf MAC-Ebene
802.11j	2004	Erweiterungen von 802.11a für Japan
802.11k	vsl. 2011	spezifiziert interne Methoden zur Messung, Auswertung und Verwaltung von Funkparametern
802.11m	vsl. 2011	Pflege und Verbesserungen des Standards und dessen Erweiterungen
802.11n	vsl. 2009	physikalische Erweiterung für $\geq$100 Mbps basierend auf MIMO
802.11p	vsl. 2009	Wireless Access for Vehicular Environments (WAVE), Erweiterung für Inter-Fahrzeug Kommunikation
802.11r	vsl. 2008	Erweiterung für Fast-Roaming zwischen Access Points, besonders wichtig für VoWLAN
802.11s	vsl. 2010	Erweiterung für Wireless Mesh Networks (WMNs)
802.11T	vsl. 2009	spezifiziert Metriken, externe Messmethoden und Testverfahren für die Leistungsfähigkeit von WLAN Komponenten
802.11u	vsl. 2010	regelt Zusammenspiel mit nicht 802 konformen Netzen wie UMTS
802.11v	vsl. 2010	spezifiziert neue Management-Funktionen
802.11w	vsl. 2009	spezifiziert Schutzmechanismen für Management Pakete
802.11y	vsl. 2008	physikalische Erweiterung für Nutzung des 3,5 GHz bis 3,70 GHz Band in den USA
802.11z	vsl. 2009	spezifiziert Mechanismen für Direct Link Setup (DLS)

Tabelle 2.1: Übersicht der verschiedenen IEEE 802.11-Erweiterungen. Besonders hervorgehoben werden die Erweiterungen, die für den Inhalt dieser Arbeit relevant sind.

[1] http://www.wi-fi.org

2.2 Systemarchitektur

Die 802.11-Systemarchitektur besteht aus verschiedenen Komponenten, die miteinander interagieren, um höheren Schichten den transparenten Zugriff auf ein WLAN zu ermöglichen, unabhängig von dessen Eigenschaften wie Mobilität von Stationen, Größe oder Betriebsart des Netzes. Die beiden grundlegenden Arten, mit denen ein 802.11-Netz betrieben werden kann, sind der Infrastruktur-Modus und der Ad-hoc-Modus. Beide Betriebsarten und die dazugehörigen Komponenten werden in den folgenden Abschnitten kurz erläutert.

2.2.1 Infrastruktur-Modus

Ein im Infrastruktur-Modus betriebenes WLAN besteht aus mindestens einem *Basic Service Set* (BSS), dem grundlegenden Baustein eines jeden WLANs. Ein Infrastruktur-BSS besteht wiederum aus einem *Access Point* (AP) und einer beliebigen Anzahl von Stationen (STAs). Der AP stellt eine Station mit zusätzlicher Management-Funktionalität dar. Damit zwei Stationen miteinander kommunizieren können, müssen sie sich zuvor mit dem AP verbinden. Dieser leitet die Nachrichten zwischen den Stationen weiter. Ein Ausnahme stellen *Quality of Service* (QoS) BSSs dar. Diese Art von Netzwerken zur Unterstützung von QoS-Diensten ist durch die Erweiterung 802.11e hinzugekommen und ermöglicht die optionale Etablierung direkter Verbindungen zwischen zwei Station durch ein *Direct Link Setup* (DLS).

Bei einem *Extended Service Set* (ESS) werden zwei oder mehrere Infrastruktur-BSSs über ein *Distribution System* (DS) zusammengeschlossen und somit ein Wechsel (*Roaming*) von Stationen zwischen den verschiedenen BSSs ermöglicht. Dieser Wechsel ist für höhere Schichten wie die LLC-Schicht transparent. Die Kommunikation im DS kann sowohl über Ethernet als auch über 802.11 geschehen. Die Verwendung von 802.11 für ein DS ist noch nicht im Standard vorgesehen, soll aber in Zukunft durch die Erweiterung 802.11s ermöglicht werden. Ein ESS kann dann ausschließlich aus 802.11-Komponenten bestehen, um somit den Aufbau von kabellosen Mesh-Netzen zu erlauben. In Abbildung 2.2 sind noch einmal alle vorgestellten Komponenten inklusive des im nächsten Abschnitt beschriebenen IBSS dargestellt.

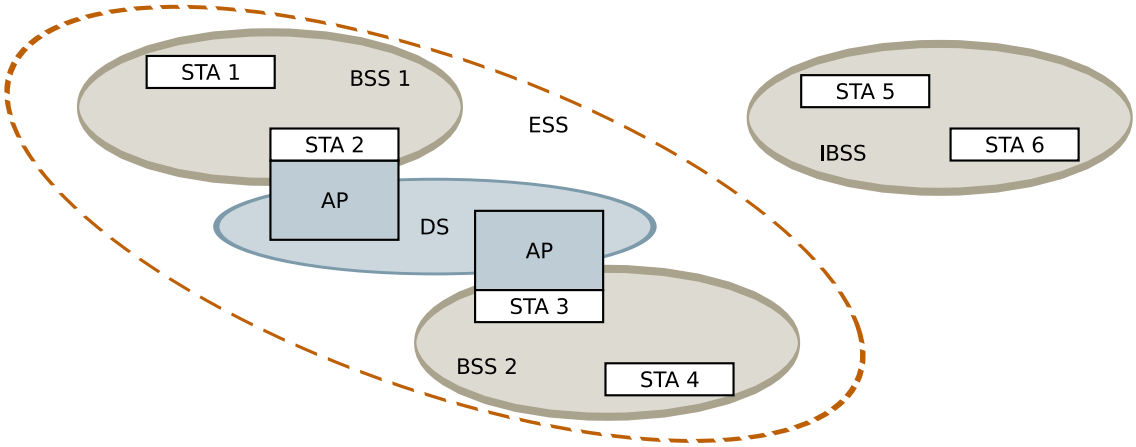


Abbildung 2.2: Übersicht der Komponenten der 802.11-Architektur

2.2.2 Ad-hoc-Modus

Der Ad-hoc-Modus stellt theoretisch die einfachste Betriebsart eines 802.11-Netzes dar. Stationen kommunizieren im Ad-hoc-Modus auf direktem Wege miteinander und bilden ein sogenanntes *Independent BSS* (IBSS), wie in Abbildung 2.2 auf der vorherigen Seite dargestellt. Um Verwechslungen zu vermeiden, wird ein Infrastruktur-BSS teilweise auch nur als BSS und niemals als IBSS bezeichnet. Der Standard sieht die Verwendung eines IBSS insbesondere für den spontanen Aufbau von kurzlebigen, kabellosen Verbindungen zwischen einer kleinen Anzahl von Stationen in einer statischen Topologie vor. Dies hat zur Folge, dass manche Hersteller ihre eigenen Ad-hoc-Modi implementieren, die zu einem 802.11-IBSS meist inkompatibel sind. Der bekannteste dieser Art ist der von Lucent entwickelte Ad-hoc-Demo-Modus. Diese auch als *Pseudo-IBSS Mode* (PIM) bekannte Variante wird von verschiedenen Treibern wie beispielsweise Madwifi[2] unterstützt. Sie bietet vor allem eine bessere Performanz und einen höheren Datendurchsatz als der Standard IBSS-Modus. Der Hauptunterschied besteht in der Aussparung von jeglichen Management-Nachrichten, die in Abschnitt 2.5.3 genauer beschrieben werden. Der Ad-hoc-Demo-Modus ist besonders bei Forschungsprojekten beliebt, die den Overhead eines IBSS vermeiden wollen.

Der Ad-hoc-Modus des 802.11-Standards ermöglicht wesentlich flexiblere Netztopologien als der Infrastruktur-Modus. Stationen können zusätzlich mobil sein und bilden somit ein *Mobile Ad Hoc Network* (MANET). Um eine Kommunikation zwischen allen Teilnehmern eines MANETs zu ermöglichen, muss das Netz sich selbst organisieren. Das bedeutet, dass Routen gefunden und kontrolliert werden müssen. Dies ist trivial falls alle Stationen in Reichweite sind, kann aber bei großen Netzen, in denen Nachrichten über mehrere Stationen hinweg an ein Ziel weitergeleitet werden müssen, zu einer komplexen Aufgabe werden. Netze dieser Art werden als *Multihop* MANETs bezeichnet. Das *Routing* geschieht dabei durch höhere Schichten, da der 802.11-Standard hierfür momentan keine Unterstützung bietet.

Die Arbeitsgruppe 802.11p [62, 67] beschäftigt sich mit MANETs, die speziell für die Kommunikation zwischen Fahrzeugen ausgelegt sind. Solche Netze werden als *Vehicular Ad Hoc Networks* (VANET) bezeichnet. Um 802.11-Stationen die Operation in einem VANET zu ermöglichen, spezifiziert 802.11p einige Änderungen des Standards, die insbesondere den Overhead zum Aufbau oder zum Eintritt in ein BSS reduzieren. Dies geschieht durch einen Wechsel der Stationen in den sogenannten *Wireless Access in Vehicular Environments* (WAVE) Modus. Außerdem spezifiziert 802.11p die Verwendung eines neuen Frequenzbandes im 5.9-GHz-Bereich für die ausschließliche Nutzung durch VANETs und den Aufbau eines WAVE BSS. Des Weiteren sollen Stationen über eine *Wildcard* BSSID direkt miteinander kommunizieren können, ohne zuvor ein BSS aufbauen zu müssen. Auch die Anzahl unterstützter Management- und Control-Nachrichten wurden auf ein Minimum beschränkt. So werden nur Beacons, RTS-, CTS- und ACK-Nachrichten sowie QoS-Datenpakete unterstützt, vergleiche Abschnitt 2.5.2.

Eine weitere Form von Ad-hoc-Netzen stellen *Wireless Mesh Networks* (WMN) dar. Diese bestehen im Gegensatz zu MANETs aus mehreren *Mesh Routern*, die keine bis wenig Mobilität besitzen und *Mesh Clients*, die eine hohe Mobilität besitzen können. Auch WMNs organisieren sich selbst, allerdings übernehmen nur die *Mesh Router* die Aufgabe neue Routen zu finden und Nachrichten weiterzuleiten. Bewegungen der *Mesh Clients* wirken sich somit geringfügiger auf das Routing aus, als dies bei MANETs der Fall ist. Die Arbeitsgruppe 802.11s [63] beschäftigt sich mit der Erweiterung des Standards um den Aufbau eines solchen Mesh-Netzes zu ermöglichen.

[2] http://madwifi.org/wiki/UserDocs/AhdemoInterface

2.3 Protokollarchitektur

Der IEEE 802.11-Standard spezifiziert die beiden Schichten *Physical Layer* (PHY) und *Medium Access Control* (MAC), welche jeweils auf der Bitübertragungsschicht sowie der Sicherungsschicht des ISO/OSI-Referenzmodells anzusiedeln sind, siehe Abbildung 2.3. Die Sicherungsschicht besteht neben der MAC-Teilschicht aus der weiteren Teilschicht *Logical Link Control* (LLC) für die logische Steuerung von Verbindungen. Diese besitzt die Aufgabe, die verschiedenen Verfahren der MAC Teilschicht zu abstrahieren, um somit eine einheitliche Schnittstelle für höher gelegene Schichten wie die Vermittlungsschicht bereitzustellen. Die PHY-Schicht ist ebenfalls noch einmal in die beiden Teilschichten *Physical Layer Convergence Procedure* (PLCP) und *Physical Medium Dependent* (PMD) unterteilt. Die PMD-Teilschicht übernimmt dabei die Modulation und Kodierung, während die PLCP eine medienunabhängige Schnittstelle für die MAC-Teilschicht zur Verfügung stellt. Die jeweiligen Schnittstellen zwischen den Teilschichten werden durch sogenannte Dienstzugangspunkte (*Service Access Points*) bereitgestellt.

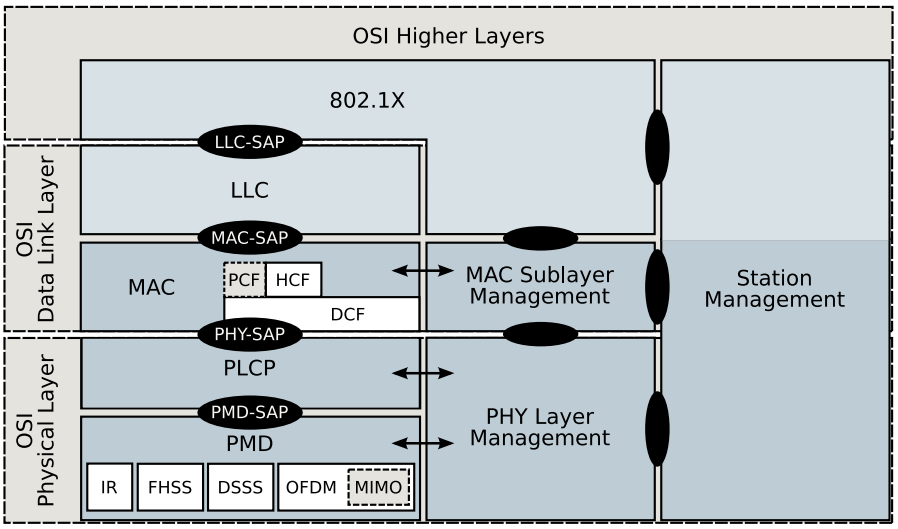


Abbildung 2.3: IEEE 802.11-Referenzmodell [60]

Einen wesentlichen Unterschied zu kabelgebundenen Protokollstapeln wie Ethernet (IEEE 802.3), stellt die zusätzliche Notwendigkeit von Management-Einheiten dar. Diese Einheiten sind für die Kontrolle und Steuerung der Verbindung verantwortlich. Zur ihren Aufgaben gehört die Wahl des Netzes bei mehreren sichtbaren APs, der Aufbau einer sicheren Verbindung bevor Datenpakete übertragen werden sowie der Umgang mit mobilen Stationen. Eine genaue Darstellung der 802.11-Management-Funktionen wird in Abschnitt 2.5.3 gegeben. Diese zu realisieren ist wesentlich komplexer als es bei kabelgebundenen Netzen der Fall ist. Bei diesen kann bereits das Umstecken eines Kabels genügen, um eine Verbindung zu einem gewünschte Netz herzustellen. Folgende Punkte stellen die wesentlichen Unterschiede von 802.11-Netzen zu kabelgebundenen Netzen dar:

- Benutzung eines Mediums ohne physische Grenzen und daher keine physische Beschränkung der Stationen, die Pakete empfangen können.
- Fehlender Schutz vor anderen Signalen, die das gleiche Medium benutzen.
- Kommunikation über ein weniger zuverlässiges Medium aufgrund externer Störeinflüsse.
- Dynamische Änderung der Topologie durch mobile Stationen.

- Fehlen voller Konnektivität, d.h. nicht jede Station kann jede andere Station immer hören (vgl. Hidden-Station-Problem Abschnitt 2.5 auf Seite 13).
- Auftreten von Zeitunterschieden und asymmetrischen Ausbreitungseigenschaften.
- Auftreten von Interferenzen durch andere 802.11-Netze, die in überschneidenden Frequenzbereichen arbeiten.

2.4 PHY-Schicht

Die PHY-Schicht spezifiziert auf Ebene der PMD-Teilschicht die verschiedenen Modulations- und Kodierverfahren für die konkrete Datenübertragung durch Funkwellen in den lizenzfreien ISM-Frequenzbändern im 2,4-GHz- und 5-GHz-Bereich. Der erste Standard von 1997 sah zunächst nur drei Modulationstechniken basierend auf *Pulse Position Modulation* (PPM), *Frequency Hopping Spread Spectrum* (FHSS) und *Direct Sequence Spread Spectrum* (DSSS) vor, mit denen lediglich Datenraten von 1 und 2 Mbps erreicht werden konnten. Erst mit der Erweiterung 802.11a kam ein weiteres Modulationsverfahren namens *Orthogonal Frequency Division Multiplexing* (OFDM) hinzu, das in Verbindung mit einem höheren Frequenzband, Datenraten bis zu 54 Mbps ermöglicht. Noch höhere Datenraten von mindestens 100 Mbps werden mit der noch in Bearbeitung befindlichen Erweiterung 802.11n angestrebt. Diese basiert ebenfalls auf OFDM in Verbindung mit der sogenannten *Multiple Input Multiple Output* (MIMO) Technik.

Um die verschiedenen Modulationstechniken transparent der MAC-Schicht zur Verfügung zu stellen, spezifiziert die PHY-Schicht weiterhin die PLCP-Teilschicht, welche die grundlegenden Dienstprimitiven am Zugangspunkt PHY-SAP bereitstellt, siehe Abbildung 2.3. Eine der wichtigsten Primitiven ist die Methode *PHY-CCA.indicate*. Diese signalisiert der MAC-Schicht ob der Kanal aktuell belegt (*busy*) oder frei ist (*idle*). Weitere Dienstprimitive sind in Tabelle 2.2 aufgelistet. Eine Übersicht wichtiger Parameter der PHY-Schicht, abhängig von dem verwendeten Modulationsverfahren, ist in Tabelle 2.3 zu sehen. In den folgenden Abschnitten werden die verschiedenen Verfahren zur Modulation kurz erläutert. Obwohl der Standard die PPM basierend auf Infrarot vorgesehen hatte, wurde diese in keiner Umsetzung verwendet und soll an dieser Stelle auch nicht weiter betrachtet werden.

2.4.1 Frequency Hopping Spread Spectrum

Bei Frequenzspreizverfahren wie dem FHSS-Modulationsverfahren werden Signale mit einer höheren Bandbreite übertragen, als dies für die Symbolrate der Signale notwendig wäre. Durch diese Spreizung lässt sich zum einen eine größere Resistenz gegen Störeinflüsse auf dem Übertragungskanal und zum anderen ein schwächerer Einfluss des Senders auf seine Umgebung erreichen. Dazu wechseln Sender und Empfänger nach pseudozufälligen Zeiteinheiten gleichzeitig die Trägerfrequenz und führen somit einen gemeinsamen Sprung (eng. Hop) durch. Da existierende Systeme basierend auf FHSS nur geringe Datenraten bis 2 Mbps unterstützen, sind diese heute kaum noch verbreitet. Daher wird an dieser Stelle ebenfalls auf eine ausführlichere Darstellung verzichtet.

Primitive		Beschreibung
PHY-DATA	*.request*	Transfer eines Datenbytes von der MAC- zur PHY-Schicht
	.confirm	Bestätigung des Transfers von MAC- zur PHY-Schicht
	.indicate	Datentransfer von PHY- zur MAC-Schicht
PHY-TXSTART	*.request*	Aufforderung der MAC-Schicht zur Übertragung einer MPDU durch die PHY-Schicht
	.confirm	Bestätigung des Begins der Übertragung an die MAC-Schicht
PHY-TXEND	*.request*	Aufforderung der MAC-Schicht zum Beenden der Übertragung an die PHY-Schicht
	.confirm	Bestätigung des Abschluss einer Übertragung an die MAC-Schicht
PHY-CCARESET	*.request*	Aufforderung der MAC-Schicht zur Zurücksetzung der CS/CCA Timer an die PHY-Schicht
	.confirm	Bestätigung der Zurücksetzung an die MAC-Schicht
PHY-CCA	*.indicate*	Signalisiert der MAC-Schicht den aktuellen Zustand des Kanals
PHY-RXSTART	*.indicate*	Signalisierung der PHY-Schicht über Empfang eines korrekten PLCP-Headers an die MAC-Schicht
PHY-RXEND	*.indicate*	Signalisierung der PHY-Schicht über Abschluss des Empfangs einer MPDU an die MAC-Schicht

Tabelle 2.2: Dienstprimitive der PHY-Schicht

2.4.2 Direct Sequence Spread Spectrum

Das DSSS-Modulationsverfahren ist ebenfalls ein Frequenzspreizverfahren und basiert auf einer logischen *Exclusive Or* (XOR) Verknüpfung der Daten mit einer Pseudozufallsfolge, die eine höhere Bitrate als das Ausgangssignal aufweist. Dies hat den Vorteil, dass schmalbandige Störungen mit hoher Intensität in ein breitbandiges Rauschen mit niedriger Intensität gespreizt werden und somit eine größere Störunempfindlichkeit erreicht wird. Bei 802.11 wird hierbei für alle Kanäle der gleiche 11 Bit lange *Barker Code* [10] als Pseudozufallsfolge verwendet. Jeder Kanal besitzt eine Bandbreite von je 22 MHz. Da der Standard 13 Kanäle für die in Europa verfügbare Bandbreite von 2,412 GHz bis 2,472 GHz vorsieht, sind jeweils nur 3 überlappungsfreie Kanäle parallel nutzbar.

PLCP Protocol Data Unit (PPDU)						
PLCP Preamble		**PLCP Header**				**MPDU**
SYNC	SFD	PSF	SERVICE	LENGTH	HEC	
128 Bits	*16 Bits*	*8 Bits*	*8 Bits*	*16 Bits*	*16 Bits*	*0 bis 4095 Bytes*

Abbildung 2.4: Rahmenformat für die PLCP bei DSSS

In Abbildung 2.4 ist das vorgeschriebene Rahmenformat dargestellt, welches von der PLCP-Schicht bei Verwendung von DSSS zur Übertragung an die PMD-Schicht übergeben wird. Das SYNC-Feld dient der Erkennung ankommender Signale und der Synchronisation. Der *Start Frame Delimiter* (SFD) ist die feste Abfolge der Bits 1111 0011 1010 0000 zur Kennzeichnung eines Frame-Anfangs. Die Nutzdaten werden als *MAC Protocol Data Unit* (MPDU) bezeichnet und stellen das von der MAC-Schicht übergebene Paket dar. Das LENGTH-Feld gibt die Zeit in Mikrosekunden an, die für die Übertragung der MPDU benötigt wird, und kann einen Wert zwischen 16 und $2^{16} - 1$ annehmen. Der Header wird zusätzlich durch eine CRC-16 Prüfsumme zur Fehlererkennung im Feld *Header Error Check* (HEC) versehen. Das SERVICE-Feld des Headers wird momentan noch nicht verwendet und ist für zukünftige Nutzung reserviert.

Parameter	DSSS	OFDM	Beschreibung
aSlotTime	20 μs	9 . . . 21 μs	Die Zeit, die die MAC-Schicht benötigt um PIFS- und DIFS-Zeitspannen zu definieren. Berechnet aus der Summe: $aCCATime + aRxTxTurnaroundTime + aAirPropagationTime + aMACProcessingDelay$.
aSIFSTime	10 μs	16 . . . 64 μs	Die nominelle Zeit, die die MAC- und PHY-Schicht benötigen, um das letzte Symbol eines Frames über die Funkschnittstelle zu empfangen, zu verarbeiten und mit dem ersten Symbol des frühestmöglichen Frames über die Funkschnittstelle zu antworten. Berechnet aus der Summe: $aRxRFDelay + aRxPLCPDelay + aMACProcessingDelay + aRxTxTurnaroundTime$.
aCCATime	$\leq$ 15 μs	4 . . . 16 μs	Die minimale Zeit, die dem CCA-Mechanismus mit jedem Zeitschlitz zur Verfügung steht um, das Medium auf Belegung zu überprüfen.
aRxTxTurnaroundTime	$\leq$ 5 μs	$<$ 2 μs	Die maximale Zeit, die die PHY-Schicht zum Wechsel zwischen Empfang und Senden des Anfangs des ersten Symbols benötigt, als Ergebnis der Summe: $aTxPLCPDelay + aRxTxSwitchTime + aTxRampOnTime + aTxRFDelay$
aTxPLCPDelay	-	-	Die nominelle Zeit, die die PLCP benötigt, um ein Symbol von der MAC-Schnittstelle bis zum *Transmit Data Path* der PMD zu übermitteln.
aRxPLCPDelay	-	-	Die nominelle Zeit, die die PLCP benötigt, um das letzte Bit eines empfangenen Frames vom PMD *Receive Path* bis zur MAC zu übermitteln.
aRxTxSwitchTime	$\leq$ 5 μs	$<<$ 1 μs	Die nominelle Zeit, die die PMD-Schicht für den Wechsel zwischen Empfang und Senden benötigt.
aTxRampOnTime	-	-	Die maximale Zeit, die die PMD-Schicht zum Einschalten des Senders benötigt.
aTxRFDelay	-	-	Die nominelle Zeit zwischen Rückgabe der Methode `PMD_DATA.request` an die PMD und Anfang des zugehörigen Symbols an der Funkschnittstelle.
aRxRFDelay	-	-	Die nominelle Zeit zwischen Ende eines Symbols an der Funkschnittstelle und Rückgabe der Methode `PMD_DATA.indicate` an die PLCP.
aAirPropagationTime	1 μs	$<<$ 1 μs	Die doppelte Laufzeit, die ein Signal für die maximale Strecke zwischen den am weitesten entfernten Stationen benötigt.
aMACProcessingDelay	$\leq$ 2 μs	$<$ 2 μs	Die maximal verfügbare Zeit für die MAC-Schicht, um die Primitive `PHYTXSTART.request` nach einer `PHY-RXEND.indication` oder `PHY-CCA.indication(IDLE)` auszuführen.
aPreambleLength	144 μs	16 . . . 64 μs	Die Länge der aktuellen Präambel.
aPLCPHeaderLength	48 μs	4 . . . 16 μs	Die Länge des aktuellen PLCP-Headers.
aMPDUMaxLength	4095	4095	Die maximale Länge einer MPDU in Bytes, die eine PPDU enthalten kann.
aCWmin	31	15	Die minimale Größe des *Contention Window* in Einheiten von aSlotTime.
aCWmax	1023	1023	Die maximale Größe des *Contention Window* in Einheiten von aSlotTime.

Tabelle 2.3: Übersicht wichtiger PHY-Parameter in Abhängigkeit der beiden Modulationsverfahren DSSS und OFDM

2.4.3 Orthogonal Frequency Division Multiplexing

Das OFDM-Verfahren, auch als *Multi-Carrier* oder *Discrete Multi-Tone Modulation* bekannt, basiert auf der parallelen Datenübertragung auf mehreren Kanälen durch Frequenzmultiplexverfahren (FDMA). Der Hauptunterschied zu den herkömmlichen FDMA-Verfahren liegt in der Möglichkeit Kanäle überlappen zu können, indem Unterfrequenzen gewählt werden, die zueinander orthogonal sind. Dies ermöglicht höhere Datenraten bei vergleichsweise niedriger Störanfälligkeit. Der 802.11-Standard, insbesondere die Erweiterung 802.11a, definiert drei verschiedene Kanalabstände von 5 MHz, 10 MHz und 20 MHz für die Hauptkanäle. Die Hauptkanäle sind wiederum in 52 Unterkanäle von jeweils 300 KHz Bandbreite aufgeteilt. Dabei sind 48 Unterkanäle für Nutzdaten und 4 Unterkanäle für die Synchronisation vorgesehen. Durch die vier möglichen Modulationsverfahren BPSK, QPSK, 16-QAM und 64-QAM können in Verbindung mit unterschiedlichen Code-Raten Datenraten von 6 Mbps bis 54 Mbps erreicht werden, siehe Tabelle 2.4.

Modulation	**Coding Rate**	**Brutto Datenrate bzgl. Kanalabstand [Mbps]**		
		20 MHz	*10 MHz*	*5 MHz*
BPSK	1/2	6	3	1.5
BPSK	3/4	9	4.5	2.25
QPSK	1/2	12	6	3
QPSK	3/4	18	9	4.5
16-QAM	1/2	24	12	6
16-QAM	3/4	36	18	9
64-QAM	2/3	48	24	12
64-QAM	3/4	54	27	13.5

Tabelle 2.4: Modulationsarten und Datenraten bei Verwendung von OFDM

Das in Abbildung 2.5 dargestellte Format eines PLCP-Rahmens unterscheidet sich besonders durch den PLCP-Header von dem zuvor beschriebenen Rahmenformaten. Dieser besteht aus den Feldern RATE, Reserved, LENGTH, Parity, Tail und SERVICE. Das RATE-Feld legt dabei gemäß Tabelle 2.4 die Modulationsart und somit die Datenrate fest, mit der der DATA-Teil übertragen wird. Der DATA-Teil besteht aus den Feldern SERVICE, PSDU, Tail und Pad. Der SIGNAL-Teil, bestehend aus den Feldern RATE, Reserved, LENGTH, Parity und Tail, wird hingegen immer mit der widerstandsfähigsten Kombination aus BPSK und einer Coding Rate von 1/2 übertragen. Die Tail-Felder sowie die ersten sechs Bit des SERVICE-Feldes bestehen aus einer festen Abfolge von sechs Nullen. Der Rest des SERVICE-Feldes ist für spätere Nutzung reserviert.

<table>
<tr><td colspan="9">PLCP Protocol Data Unit (PPDU)</td></tr>
<tr><td rowspan="2">Preamble</td><td colspan="6">PLCP Header</td><td rowspan="2">PSDU</td><td rowspan="2">Tail
6 Bits</td><td rowspan="2">Pad Bits</td></tr>
<tr><td>RATE
4 Bits</td><td>Rsvd
1 Bit</td><td>LENGTH
12 Bits</td><td>Parity
1 Bit</td><td>Tail
6 Bits</td><td>SERVICE
16 Bits</td></tr>
<tr><td>12 Symbols</td><td colspan="5">SIGNAL (One OFDM Symbol)</td><td colspan="3">DATA (Var. OFDM Symbols)</td></tr>
</table>

Abbildung 2.5: Rahmenformat für die PLCP bei OFDM

2.4.4 Multiple Input Multiple Output

MIMO beschreibt kein Modulationsverfahren im eigentlichen Sinne, sondern ist die Bezeichnung für die gleichzeitige Verwendung mehrerer Sende- und Empfangsantennen, um durch Techniken wie *Spatial Division Multiplexing* (SDM), *Space Time Block Coding* (STBC) oder Beamforming eine höhere spektrale Effizienz und somit höhere Datenraten zu erreichen [91, 93]. Hierbei wird besonders die Kenntnis über die Kanalbeschaffenheit und physikalische Effekte bei der Übertragung von Funkwellen wie die Mehrwegeausbreitung ausgenutzt. In Verbindung mit OFDM und einer größeren Kanalbandbreite von 40 MHz sollen Datenraten bis 600 Mbps erreicht werden können. Da die Verwendung von MIMO Bestandteil der Erweiterung 802.11n [61] ist und diese durch die IEEE noch nicht verabschiedet wurde, soll an dieser Stelle auf eine genauere Darstellung der MIMO-Techniken verzichtet werden.

2.5 MAC-Schicht

Die MAC-Schicht des 802.11-Standards spezifiziert die verschiedenen Methoden für den geteilten Kanalzugriff auf das kabellose Medium eines WLANs. Eine besondere Herausforderung der MAC-Schicht ist dabei der Umgang mit Problemen, die durch das kabellose Medium impliziert werden. Dazu gehört beispielsweise das Auftreten von Interferenzen durch Mehrwegeausbreitung des Funksignals oder das Hidden-Station-Problem, bei dem es zu Störungen durch Stationen außerhalb der eigenen Sendereichweite kommen kann. Um letzteres Problem zu lösen spezifiziert der Standard auf Ebene der MAC-Schicht den RTS/CTS-Mechanismus. Des Weiteren stellt die MAC-Schicht die unterschiedlichen Paketformate für Daten-, Management- und Kontrollnachrichten bereit. Die folgenden Abschnitte beschreiben zunächst die grundlegenden Verfahren für den Kanalzugriff und gehen anschließend auf die unterschiedlichen Paketformate und Management-Funktionen der MAC-Schicht ein.

2.5.1 Kanalzugriff

Der grundlegende Mechanismus für den Kanalzugriff des 802.11-Standards basiert auf der *Distributed Coordination Function* (DCF). Auf dieser setzen wiederum die beiden weiteren Mechanismen *Point Coordination Function* (PCF) für den konkurrenzlosen Zugriff durch eine Steuereinheit und *Hybrid Coordination Function* (HCF) auf. Bei der HCF, welche durch die Erweiterung 802.11e [59] hinzugekommen ist, unterscheidet der Standard noch einmal zwischen den Zugriffsmethoden *Enhanced Distributed Channel Access* (EDCA) und *HCF Controlled Channel Access* (HCCA). Diese beiden Methoden dienen der Unterstützung von QoS-Übertragungen basierend auf vordefinierten Prioritäten oder Parametern. Dies ist besonders bei Audio- und Videoübertragungen von großer Bedeutung, bei denen die Mechanismen des herkömmlichen Kanalzugriffs nicht ausreichend sind. Abbildung 2.6 veranschaulicht die Beziehungen zwischen den verschiedenen Coordination Functions der MAC-Schicht.

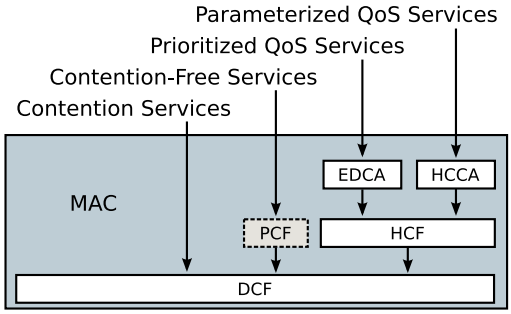


Abbildung 2.6: Übersicht der verschiedenen MAC Coordination Functions

Interframe Spaces

Im Rahmen des Zugriffsmechanismus werden unterschiedliche Arten von *Interframe Spaces* (IFS) definiert, die in Einheiten der vorgegebenen PHY-Zeitschlitze $aSlotTime$ angegeben werden. Diese Intervalle sind die Basis aller Zugriffsverfahren und ermöglichen eine einfache Form der Priorisierung. Je kleiner das Zeitintervall ist, desto größer ist die Priorität. Die DCF definiert zu diesem

Zweck die drei Intervalle *Distributed (Coordination Function) Interframe Space* (DIFS), *Short Interframe Space* (SIFS) und *Extended Interframe Space* (EIFS). Die PCF und HCF führen zusätzlich die beiden weiteren Intervalle *Point (Coordination Function) Interframe Space* (PIFS) und *Arbitration Interframe Space* (AIFS) ein. Die grundlegende Zugriffsmethode und die zeitlichen Beziehungen zwischen den einzelnen IFSs ist in Abbildung 2.7 skizziert. Die verschiedenen Zeitintervalle sind abhängig von festen Parametern der zugrundeliegenden PHY-Schicht, siehe Tabelle 2.3 auf Seite 10. Sie lassen sich wie folgt berechnen:

- $SIFS = aSIFSTime$
- $DIFS = SIFS + 2 \cdot aSlotTime$
- $EIFS = SIFS + DIFS + ACKTxTime$
  mit $ACKTxTime$ als die Dauer für die Übertragung eines ACK Frames
- $PIFS = SIFS + aSlotTime$
- $AIFS[AC] = SIFS + AIFSN[AC] \cdot aSlotTime$
  mit $AIFSN[AC] = \begin{cases} 7 & \text{für } AC = \text{Background} \\ 3 & \text{für } AC = \text{Best Effort} \\ 2 & \text{für } AC = \text{Audio/Video} \end{cases}$

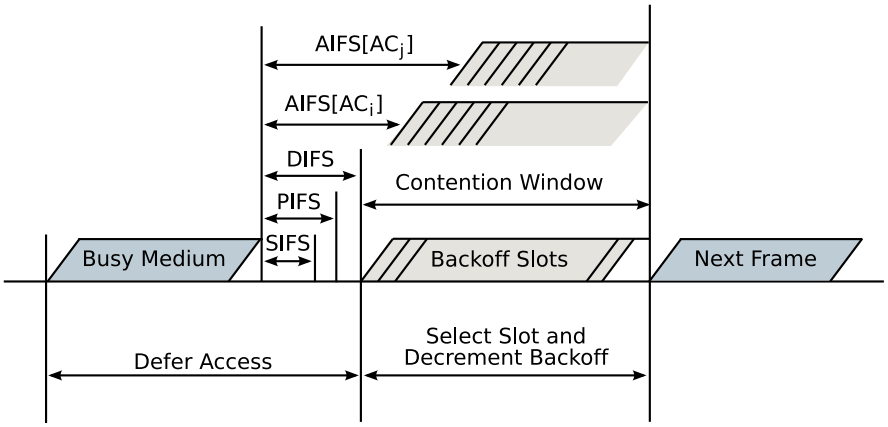


Abbildung 2.7: Grundlegende Zugriffsmethode und zeitliche Beziehungen verschiedener IFSs

Distributed Coordination Function

Die *Distributed Coordination Function* (DCF) beschreibt das grundlegende Verfahren für den Kanalzugriff auf Basis von CSMA/CA, welches in jedem BSS unterstützt werden muss. *Carrier Sense Multiple Access* (CSMA) bedeutet in diesem Zusammenhang, dass jede Station fortlaufend den physischen Zustand des gemeinsamen Kanals überwacht, um festzustellen ob der Kanal für eine Übertragung frei ist. Grundsätzlich darf eine Station nicht senden wenn sie den Kanal als belegt erkannt hat. Zusätzlich besteht die Möglichkeit einer virtuellen Erkennung des Kanalzustands (*virtual Carrier Sense*) durch zeitliche Reservierung eines *Network Allocation Vectors* (NAV) mit Hilfe des RTS/CTS-Mechanismus.

Informationen über den physischen Zustand des Kanals erhält die MAC-Schicht über die *Clear Channel Assessment* (CCA) Dienstprimitive `PHY-CCA.indicate`, vergleiche Tabelle 2.2. Möchte eine Station senden und ist der Kanal für eine Zeitdauer eines DIFS frei, darf diese umgehend

mit dem Senden beginnen. Möchte eine Station senden und erkennt hingegen den Kanal als belegt, muss sie nach der Wartezeit eines DIFS zusätzlich einen Backoff-Prozess starten, der das Senden um eine zufällige Dauer verlängert. Dieser Prozess ist Teil der *Collision Avoidance* (CA) und verringert die Wahrscheinlichkeit, dass mehrere konkurrierende Stationen gleichzeitig auf das Medium zugreifen und somit eine Kollision verursachen. Eine Ausnahme stellt der Empfang eines fehlerhaften Pakets dar, welcher durch die Dienstprimitive `PHY-RXEND.indicate` oder durch eine fehlerhafte Prüfsumme (FCS) festgestellt wird. In diesem Falle muss der Kanal für die Dauer eines EIFS frei sein, bevor der Backoff-Prozess gestartet werden kann.

Ein weiterer grundlegender Bestandteil der DCF ist das Bestätigen jedes korrekt empfangenen Pakets mit eindeutiger Adresskennung durch ein ACK-Paket (*positive Acknowledgement*). Die Wartezeit beträgt hierbei nur einen SIFS, um das ACK möglichst hoch zu priorisieren. Ein Ausbleiben des ACKs lässt den Sender nach einem vordefinierten Timeout von einem Fehler bei der Übertragung ausgehen und veranlasst diesen zu einer Wiederholung der Übertragung (*Retransmission*). An dieser Stelle sei anzumerken, dass der Sender nicht feststellen kann, ob der Fehler bei der Übertragung des Datenpakets oder des ACKs aufgetreten ist.

Block Acknowledgement. Das *Block Acknowledgement* ist eine erweiterte Form des *positive Acknowledgements*, welche durch die Erweiterung 802.11e [59] eingeführt wurde und auch innerhalb der Erweiterung 802.11n [61] verwendet wird. Der Grundgedanke des Block Acknowledgements ist die Bestätigung mehrerer Datenpakete durch ein einzelnes ACK. Der Ablauf wird in die drei Phasen *Setup*, *Data & Block Ack* und *Tear Down* unterteilt, siehe Abbildung 2.8.

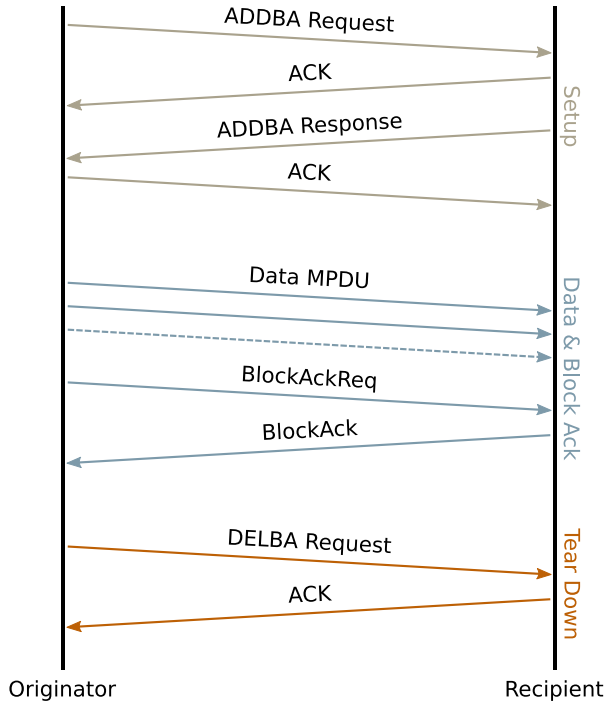


Abbildung 2.8: Ablaufdiagramm eines Block Acknowledgements

Während der *Setup*-Phase wird durch den Sender ein *Add Block Acknowledgment* (ADDBA) *Request* gesendet. Dieser spezifiziert verschiedene Parameter des Block ACKs wie *Traffic Identifier* (TID), *Buffer Size* und die *Starting Sequence Number*. Die *TID* wird durch höher gelegene Schich-

ten vergeben, um der MAC-Schicht die Zuordnung einer MSDUs zu einem bestimmten QoS-Strom zu ermöglichen. Die *Buffer Size* gibt die Anzahl der zu reservierenden Buffer auf der Seite des Empfängers und somit die Anzahl der zu erwartenden Pakete an. Ein Buffer hat dabei die maximale Größe einer MSDU von 2304 Bytes und wird von einer MPDU unabhängig von einer eventuellen Fragmentierung der MSDU komplett in Anspruch genommen. Die *Starting Sequence Number* identifiziert letztlich die erste Sequenznummer des zu erwartenden Datenstroms. Der Empfänger sendet nach Erhalt des ADDBA Request eine ADDBA *Response*, in welcher die Buffer Größe noch einmal angepasst werden kann, falls die vorgeschlagene Größe nicht verfügbar ist.

In der folgenden *Data & Block Ack* Phase kann der Sender mehrere Datenpakete (MPDUs) hintereinander versenden, die lediglich durch einen SIFS voneinander getrennt werden. Die erlaubte Anzahl der Datenpakete entspricht hierbei der zuvor festgelegten Größe des Buffers und kann maximal 1024 betragen. Nach dem alle Datenpakete gesendet wurden, fordert der Sender mit einer *BlockAckReq*-Nachricht die explizite Bestätigung des Empfängers an. Erst nach Erhalt dieser *BlockAckReq* werden die empfangenen MPDUs aus dem Buffer an den nächsten Verarbeitungsprozess der MAC-Schicht übergeben. Die Bestätigung der empfangenen Pakete erfolgt innerhalb der *BlockAck*-Nachricht durch eine 1024 Bit große Bitmap, mit der bis zu 64 MSDUs bestätigt werden können. Diese Zahl ergibt sich aus der Tatsache, dass eine MSDU in maximal 16 Fragmente unterteilt werden kann ($16 \cdot 64 = 1024$). Ein gesetztes Bit an Stelle n bestätigt die korrekte Übertragung für das Paket mit der Sequenznummer $StartingSequenceNumber + n$. Der Sender kann daraufhin nicht bestätigte Pakete erneut übertragen (*selective Retransmission*). In der abschließenden *Tear Down* Phase wird die Kommunikation mit einer *Delete Block Acknowledgment* (DELBA)-Nachricht des Senders beendet und alle Buffer wieder freigegeben.

Backoff-Prozess. Der Backoff-Prozess bestimmt die zusätzliche Wartezeit einer Station nach einem DIFS beziehungsweise EIFS als einen zufälligen Wert aus einem vorgegebenen Zeitintervall, dem *Contention Window* (CW). Dieses Intervall besteht aus Integern von 0 bis CW wobei gilt: $aCWmin \leq CW \leq aCWmax$. Für den initialen Aufruf des Backoff-Prozesses gilt $CW = aCWmin$. Die Grenzen des Intervalls $aCWmin$ und $aCWmax$ sind jeweils durch die PHY-Schicht festgelegt. Um also die Backoff-Dauer zu berechnen wird ein pseudozufälliger Wert aus dem Intervall $[0, CW]$ gewählt und mit der Dauer eines Zeitschlitzes $aSlotTime$ multipliziert. Für jeden nun folgenden Zeitschlitz wird der Kanal weiterhin auf seinen Zustand überprüft. Ist der Kanal frei, wird die Backoff-Dauer um einen Zeitschlitz reduziert. Erst wenn der Backoff den Wert 0 erreicht, darf die Station senden. Wird der Kanal hingegen als belegt erkannt, wird der Backoff-Prozess pausiert. Die Backoff-Dauer wird gespeichert und beim nächsten Backoff-Prozess als initialer Wert übernommen. Bei jedem fehlgeschlagenen Versuch ein Paket zu senden wird das CW exponentiell bis zum maximalen Wert $aCWmax$ erhöht, siehe Abbildung 2.9 auf der nächsten Seite. Zusätzlich wird für jede Übertragungswiederholung ein *Retransmission*-Zähler erhöht. Je nach Größe des Pakets wird entweder der *Station Long Retry Count* (SLRC) oder der *Station Short Retry Count* (SSRC) erhöht. Wird das Limit bei einem der Zähler erreicht oder wird ein Paket erfolgreich übertragen, wird das CW wieder auf $aCWmin$ zurückgesetzt. Der Standard schlägt als Vorgabewert für die maximale Anzahl an Retransmissions den Wert 7 vor.

RTS/CTS-Mechanismus. Der RTS/CTS-Mechanismus beschreibt die virtuelle Carrier Sense Methode mit der Stationen Informationen zur Reservierung des Kanals austauschen. Diese Methode löst einerseits das Hidden Station Problem und bietet andererseits eine bewährte Methode zur Kollisionsvermeidung beim Betrieb überlappender BSS oder IBSS. Das Hidden-Station-Problem

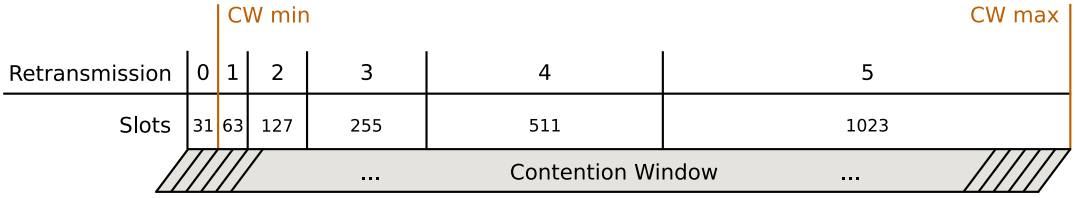


Abbildung 2.9: Exponential Backoff bei DSSS

beschreibt die in Abbildung 2.10 dargestellte Situation dreier Stationen. STA1 sowie STA3 befinden sich außerhalb des jeweiligen Empfangsbereichs, sind also voreinander versteckt. Wollen jetzt beide Stationen zur selben Zeit mit STA2 kommunizieren, kommt es zur Kollision, die sich nur bei STA2 bemerkbar macht.

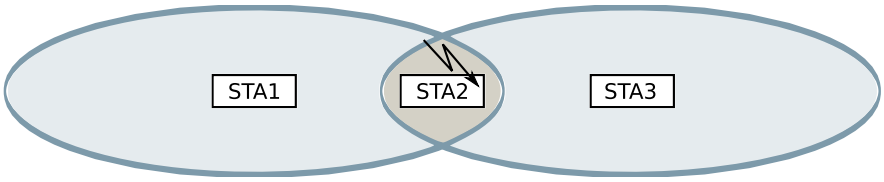


Abbildung 2.10: Anordnung dreier Stationen beim Hidden-Station-Problem

Bei Verwendung des RTS/CTS-Mechanismus muss STA1 (*Source*), bevor sie Daten an STA2 (*Destination*) senden kann, nach einer normalen Wartezeit eines DIFS und einem eventuellen Backoff zunächst ein *Request To Send* (RTS)-Paket senden, siehe Abbildung 2.11 auf der nächsten Seite. Wird dieses von STA2 korrekt empfangen, wird nach einer Wartezeit eines SIFS ein *Clear To Send* (CTS) als Antwort zurückgesendet. Die Verwendung eines SIFS sorgt hierbei für eine höhere Priorität des CTS-Pakets im Vergleich zu normalen Datenpaketen. Nach einem weiteren SIFS darf STA1 die Daten senden. Das RTS als auch das CTS-Paket enthalten dabei die Zeitdauer, die für die Übertragung des Datenpakets inklusive des zugehörigen ACKs benötigt wird. Diese Zeitdauer kann maximal 32767 µs betragen. Jede Station, die eines der beiden Pakete empfängt, muss für die angegebene Zeitdauer einen *Network Allocation Vector* (NAV) setzen. Dieser ist die Basis der virtuellen CS-Methode. Solange der NAV größer als Null ist, geht eine Station davon aus, dass der Kanal belegt ist. Mit jedem Zeitschlitz wird der NAV um eine Einheit reduziert und sobald er Null erreicht, kann eine Station wieder versuchen nach einem normalen Backoff-Prozess auf das Medium zuzugreifen. Obwohl der Standard die Verwendung des RTS/CTS-Mechanismus als optional spezifiziert, müssen alle Stationen in der Lage sein anhand der Reservierungsinformationen den NAV entsprechend zu setzen. Des weiteren sei anzumerken, dass nicht nur RTS- und CTS-Pakete, sondern auch andere Pakete der MAC-Schicht, insbesondere alle Management-Pakete, eine Zeitangabe enthalten, die für das Setzen des NAV genutzt werden kann.

Point Coordination Function

Der 802.11-Standard spezifizierte in seiner ursprünglichen Version neben der DCF zusätzlich die optionale *Point Coordination Function* (PCF). Diese ist insbesondere für die Unterstützung zeitkritischer Dienste vorgesehen. Dabei übernimmt eine zentrale Instanz, der *Point Coordinator* (PC), die Zuteilung des Kanals innerhalb eines BSS während einer konkurrenzfreien Phase, der sogenannten *Contention Free Period* (CFP). Diese Phase wechselt sich in regelmäßigen Abständen mit der

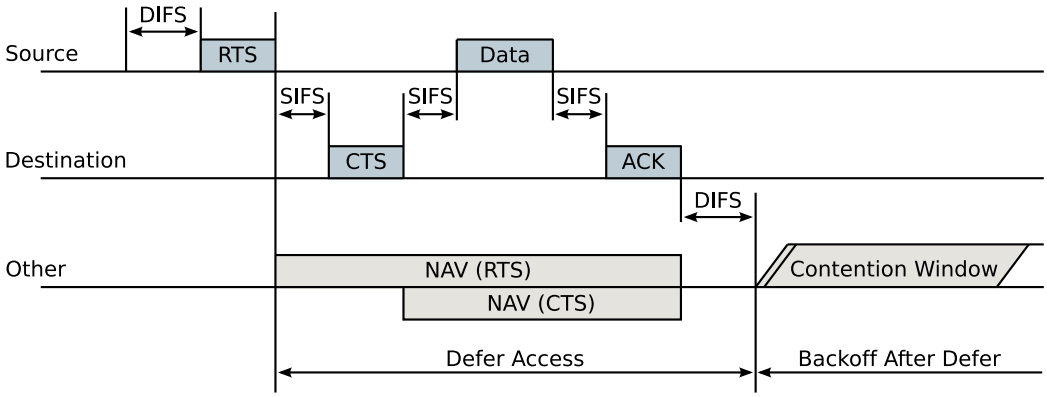


Abbildung 2.11: Virtual Carrier Sense in 802.11 [60]

Konkurrenzphase (*Contention Period*) ab. Den Beginn jeder CFP signalisiert ein Beacon-Paket (siehe Abschnitt 2.5.2). Dieses Paket enthält die Dauer der CFP, mit der jede Station innerhalb eines BSS ihren NAV setzt und somit den Kanal als belegt markiert. Da das Beacon-Paket nach einer Wartezeit eines PIFS gesendet wird, hat es eine höhere Priorität als Zugriffsversuche innerhalb der DCF, die für die längere Zeit eines DIFS warten müssen. Der Point Coordinator gibt nun während der CFP allen angemeldeten Stationen der Reihe nach die Möglichkeit Daten zu senden (*Polling*). Wenn diese keine Daten zu senden haben, wird lediglich ein leeres Paket ohne Nutzdaten zurückgeschickt (*Null Frame*). Gleichzeitig übermittelt der PC den Stationen beim *Polling* die für die jeweilige Station adressierten Datenpakete. Dies garantiert den gerecht verteilten Zugriff auf den Kanal und schließt Kollisionen der beteiligten Stationen aus. Das Ende der CFP signalisiert der PC durch den Versand eines *CF-End*-Pakets. Hiernach folgt eine erneute Konkurrenzphase. Obwohl die PCF für zeitkritische Dienste sinnvoll erscheint, wird sie in der Praxis kaum eingesetzt und nur in wenigen APs implementiert [45]. Neben der Optionalität der PCF kann auch die fehlende Berücksichtigung im Zertifizierungsverfahren der *Wi-Fi Alliance* ein Grund für die fehlende Verbreitung dieses Zugriffsverfahrens sein.

Hybrid Coordination Function

Die *Hybrid Coordination Function* (HCF) ist Teil der Erweiterung 802.11e, die Aspekte der konkurrenzbasierten und konkurrenzlosen Zugriffsmethoden sowohl kombiniert als auch erweitert, um eine Unterstützung von QoS-Diensten bereitzustellen. Stationen bekommen die Möglichkeit mehrere Warteschlangen für unterschiedliche Dienste zu verwalten, um somit den Zugriff auf den Kanal abhängig von Qualitätsanforderungen einzelner Anwendungen zu gewährleisten. Für diesen Zweck spezifiziert der Standard für die HCF die beiden Zugriffsmethoden *Enhanced Distributed Channel Access* (EDCA) für den konkurrenzbasierten und *HCF Controlled Channel Access* (HCCA) für den kontrollierten Kanalzugriff. Beide Methoden können innerhalb eines BSS simultan eingesetzt werden und sind ebenfalls kompatibel mit der PCF. Wie bei der PCF wird auch bei der HCF zwischen den beiden Phasen *Contention Period* (CP) und *Contention Free Period* (CFP) in alternierenden Intervallen gewechselt. HCCA kann hierbei sowohl während der CP als auch der CFP genutzt werden. Die Verwendung von EDCA ist hingegen nur während der CP möglich. Die grundlegende Einheit für die Vergabe des Zugriffsrechts und somit das Recht zur Übertragung von Daten wird als *Transmission Opportunity* (TXOP) bezeichnet. Jede TXOP ist durch eine Startzeit

und eine maximale Dauer[3] definiert, während der eine Station die Möglichkeit hat ihre Daten zu übertragen.

Konkurrenzbasierter Kanalzugriff. EDCA basiert auf der Erweiterung der grundlegenden DCF. Um innerhalb des EDCA-Mechanismus QoS zu unterstützen, werden für den Kanalzugriff die vier verschiedenen *Access Categories* (ACs) *Background*, *Best Effort*, *Video* und *Voice* definiert. Jedes Datenpaket muss mit einer dieser vier ACs versehen werden, um somit eine Priorität für die Übertragung festlegen zu können. Falls ein Datenpaket über keine Angabe der AC verfügt, wird standardmäßig *Best Effort* angenommen. Mit jeder AC wird sowohl ein spezieller *Interframe Space* ($AIFS[AC]$), siehe Abschnitt 2.5.1 auf Seite 13, als auch ein *Contention Window* ($CWmin[AC]$,$CWmax[AC]$) verknüpft. Diese Parameter stellen, abhängig von der Priorität eines Datenpakets, angemessene Wahrscheinlichkeiten für den Kanalzugriff und somit die Zuteilung einer TXOP sicher. Die Vorgabewerte für die Parameter des Contention Window sind in Tabelle 2.5 angegeben. Diese sind allerdings abhängig von der jeweiligen Kanalbeschaffenheit und können somit davon abweichen. Eine Station verwaltet für jede AC eine separate Warteschlange die intern wie verschiedene DCF-Konkurrenten agieren und somit unabhängig voneinander versuchen auf den Kanal zuzugreifen. Sollen zwei Pakete unterschiedlicher ACs nach einem Backoff-Prozess gleichzeitig gesendet werden, wird eine interne virtuelle Kollision ausgelöst und das Paket mit der höheren Priorität wird gesendet. Auch in 802.11p ist die Verwendung von EDCA vorgesehen.

AC	AIFSN	CWmin	CWmax
AC_BK	7	$aCWmin$	$aCWmax$
AC_BE	3	$aCWmin$	$aCWmax$
AC_VI	2	$(aCWmin+1)/2-1$	$aCWmin$
AC_VO	2	$(aCWmin+1)/4-1$	$(aCWmin+1)/2-1$

Tabelle 2.5: Vorgabewerte für das Contention Window bei EDCA

Kontrollierter Kanalzugriff. Der HCCA-Mechanismus koordiniert unter der Leitung eines *Hybrid Coordinator* (HC) den Zugriff auf den Kanal ähnlich der PCF. Der HC kann dabei jeder QoS-Station eine TXOP zuweisen. Im Gegensatz zum *Point Coordinator* der PCF kann der *Hybrid Coordinator* allerdings auch während der konkurrenzbasierten Phase (CP) einzelnen Stationen eine TXOP zuweisen und diesen somit den Vorrang vor DCF- oder EDCA-basierten Stationen verschaffen. Die Zeit, während der ein HC den Zugriff auf den Kanal sowohl innerhalb der CP als auch der CFP steuert, wird als *Controlled Access Phase* (CAP) bezeichnet. Damit eine Station eine TXOP des HCs zugeordnet bekommt, muss diese zuvor eine QoS-Reservierungsanfrage an den HC senden. Diese Anfrage ist in einer speziellen Management-Nachricht enthalten und spezifiziert den anstehenden Datenverkehr der Station durch Parameter innerhalb einer *Traffic Specification* (TSPEC). Wichtige Parameter sind beispielsweise die durchschnittliche Datenrate, die maximale Verzögerung, der maximale Abstand benachbarter TXOPs, die Größe einer MSDU oder die minimale Bitrate der PHY-Schicht zur Übertragung. Obwohl eine Station für jeden Datenstrom eine eigene TSPEC ausstellen muss, bekommt sie jeweils nur eine TXOP des HC zugewiesen, welche sie intern auf die einzelnen Ströme aufteilen muss.

[3] Das TXOP-Limit wird in einem 8 Bit Feld in der Einheit von 31 µs angegeben und hat eine Maximaldauer von 8160 µs.

2.5.2 Paketformat

Da der 802.11-Standard für ein Paket oder auch Rahmen durchgängig die englische Bezeichnung *Frame* verwendet, wird ab dieser Stelle ebenfalls auf diese Bezeichnung im Bezug auf MAC-Pakete zurückgegriffen. Ein MAC-Frame ist die Einheit, die an die PLCP-Teilschicht übergeben und somit auf Ebene der PHY-Schicht in einen PLCP-Frame eingebettet wird. Der MAC-Frame wird dabei auch als *MAC Protocol Data Unit* (MPDU) bezeichnet. Alle Frames der MAC-Schicht bestehen des Weiteren aus den folgenden drei grundlegenden Bestandteilen:

1. *MAC Header:* Beinhaltet Informationen über Verbindungssteuerung, Paketdauer, Adressierung, Sequenzsteuerung und QoS-Kontrolle, falls QoS-Datenpakete übertragen werden.
2. *Frame Body:* Besteht aus einer variablen Länge von Informationen abhängig von den Feldern *Type* und *Subtype* des Pakets.
3. *FCS:* Beinhaltet eine 32 Bit Checksumme über das Paket zur Fehlererkennung.

In Abbildung 2.12 auf der nächsten Seite ist das Format eines MAC-Frames sowie der zugehörige Aufbau des *Frame Control* Feldes dargestellt. An dieser Stelle sei anzumerken, dass nur die Felder *Frame Control*, *Duration/ID*, *Address 1* sowie *FCS* in jedem MAC-Frame enthalten sind. Das Vorkommen aller übrigen Felder ist abhängig von dem verwendeten Typ eines Pakets. Das Feld *Frame Control* besteht aus insgesamt 16 Bit und ist in die folgenden Subfelder unterteilt:

- *Protocol Version:* Kennzeichnet die Version des MAC-Protokolls. Diese ist für den 802.11-Standard bisher immer die Version 0. Nur bei inkompatiblen Änderungen soll die Versionsnummer hochgezählt werden.
- *Type:* Gibt den Typ eines Frames an. Dieser kann entweder *Management*, *Control* oder *Data* sein. Für *Control*-Frames ist der vordefinierte Wert für alle folgenden Felder mit Ausnahme von *Subtype* und *Pwr Mgmt* immer 0.
- *Subtype:* Beschreibt in Abhängigkeit vom angegebenen Typ die konkrete Funktion des Frames. Mögliche Kombinationen und die daraus resultierenden Funktionen sind in Tabelle (3) in Abbildung 2.12 auf der nächsten Seite aufgelistet.
- *To DS, From DS:* Diese beiden Felder definieren die Verwendung der vier Adressfelder. Der Standard unterscheidet hierbei zwischen Sender und Quelle beziehungsweise Empfänger und Ziel. Somit können die Adressfelder abhängig von den beiden *To/From DS* Bits verschiedene Adressarten beinhalten. Die genaue Beziehung ist in Tabelle 2.6 auf Seite 22 dargestellt.
- *More Frag:* Das MAC-Protokoll bietet die Möglichkeit Daten- und Management-Pakete zu fragmentieren. Ist dieses Bit gesetzt, folgen noch weitere Fragmente des selben Pakets, andernfalls handelt es sich um das letzte Fragment oder ein unfragmentiertes Paket.
- *Retry:* Dieses Bit wird gesetzt falls ein Frame erneut übertragen wird.
- *Pwr Mgmt:* Beschreibt den *Power Management* Modus, den eine Station nach Beendigung einer Frame Austauschsequenz annehmen wird. Dieser bleibt während einer Austauschsequenz konstant. Ein gesetztes Bit bedeutet *PS Mode*, ein nicht gesetztes Bit *active Mode*.
- *More Data:* Ein gesetztes Bit signalisiert, dass ein AP oder eine Station weitere Pakete zum senden gespeichert hat. Dies kann sowohl in *Data*- als auch in *Management*-Paketen gesetzt sein.

- *Protected Frame:* Ein gesetztes Bit signalisiert, dass der Inhalt des *Frame Body* Feldes verschlüsselt ist. Dies kann in allen *Data*-Paketen und in *Management*-Paketen vom Subtyp *Authentication* gesetzt sein.
- *Order Field:* Dieses Feld signalisiert die optionale Sortierung eines Frames bevor er an höhere Schichten weitergereicht wird. Für QoS-Stationen ist dieses Feld immer ungesetzt.

(1)

MAC Header								**Frame Body**	**FCS**
Frame Control	Duration / ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	QoS Control		
2 Bytes	*2 Bytes*	*6 Bytes*	*6 Bytes*	*6 Bytes*	*2 Bytes*	*6 Bytes*	*2 Bytes*	*0...2312 Bytes*	*4 Bytes*

(2)

Frame Control										
Protocol Version	Type	Subtype	To DS	From DS	More Frag	Retry	Pwr Mgmt	More Data	Protected Frame	Order
2 Bits	*2 Bits*	*4 Bits*	*1 Bit*	*1 Bit*	*1 Bit*	*1 Bit*	*1 Bit*	*1 Bit*	*1 Bit*	*1 Bit*

(3)

Subtype	Type		
	Management (00)	**Control (01)**	**Data (10)**
0000	Association Request	-	Data
0001	Association Response	-	Data + CF-Ack
0010	Reassociation Request	-	Data + CF-Poll
0011	Reassociation Response	-	Data + CF-Ack + CF-Poll
0100	Probe Request	-	Null (no Data)
0101	Probe Response	-	CF-Ack (no Data)
0110	-	-	CF-Poll (no Data)
0111	-	-	CF-Ack + CF-Poll (no Data)
1000	Beacon	Block Ack Request	QoS Data
1001	ATIM	Block Ack	QoS Data + CF-Ack
1010	Disassociation	PS-Poll	QoS Data + CF-Poll
1011	Authentication	RTS	QoS Data + CF-Ack +CF-Poll
1100	Deauthentication	CTS	QoS Null (no Data)
1101	Action	ACK	-
1110	-	CF-End	QoS CF-Poll (no Data)
1111	-	CF-End + CF-Ack	QoS CF-Ack + CF-Poll (no Data)

Abbildung 2.12: (1) Allgemeines Format eines MAC-Frames, (2) Aufbau des zugehörigen Control-Feldes, (3) Framefunktionen in Abhängigkeit von Type und Subtype

Die Bedeutung des 16 Bit langen *Duration/ID* Felds im MAC-Header ist abhängig von der Funktion eines Frames, der Übertragungsphase (CFP oder CP) und den QoS-Fähigkeiten einer Station. Grundsätzlich wird der Inhalt nach den folgenden Regel bestimmt:

- Für PS-Poll-Frames ist der *Association Identifier* (AID) der Senderstation enthalten. Diese wird durch den Access Point vergeben und kann aus einem Wert zwischen 1 und 2007 bestehen.
- Für Frames, die während einer *Contention Free Period* (CFP) übertragen werden, ist ein fester Wert von 32768 vorgeschrieben.
- Frames, die an eine Multicast- oder Broadcast-Adresse gesendet werden, enthalten immer den festen Wert 0.

- Für alle anderen Pakete ist der Inhalt meist die nötige Zeit für die Übertragung inklusive der Interframe-Spaces und zugehöriger ACKs.

Bis auf die Angabe der AID innerhalb eines PS-Poll-Frames, gibt das *Duration*-Feld somit die Dauer für die Reservierung des *Network Allocation Vectors* (NAV) anderer Stationen an.

In Tabelle 2.6 ist die Verwendung der vier Adressfelder in Abhängigkeit der Control-Felder *To DS* und *From DS* dargestellt. Indirekt ist die Verwendung somit abhängig von dem zugrundeliegenden Netzwerktyp. Das Feld *Address 1* beinhaltet immer die Adresse der Station, die das Paket zunächst auf der MAC-Ebene empfängt. Der Standard bezeichnet die Adresse als *Receiving Station Address* (RA). Diese muss aber nicht zwangsläufig die Zieladresse sein, die als *Destination Address* (DA) bezeichnet wird. Die DA adressiert die Station, die das Paket letztlich an die Netzwerkschicht weiter gibt. Parallel dazu existieren die *Transmitting Station Address* (TA), die im Feld *Address 2* enthalten ist und die *Source Address* (SA). Lediglich in IBSS-Netzen sind Ziel und Empfänger sowie Quelle und Sender identisch. In Infrastruktur-BSS-Netzen ist hingegen der AP Sender oder Empfänger und die jeweilige Ziel- beziehungsweise Quelladresse ist im Feld *Address 3* enthalten. Die Verwendung der vierten Adresse wird durch den 802.11-Standard nicht spezifiziert, ist aber für den Aufbau eines *Wireless Distribution System* (WDS) vorgesehen. Die *Basic Service Set Identification* (BSSID) ist ein eindeutiger Kennzeichner für ein BSS, der im Falle eines Infrastruktur-BSS der MAC-Adresse des APs entspricht. Im Falle eines IBSS wird dieser Kennzeichner durch eine vorgegebene pseudozufällige Funktion bestimmt.

To DS	From DS	Funktion	Address 1 (RA)	Address 2 (TA)	Address 3	Address 4
0	0	IBSS	DA	SA	BSSID	-
0	1	From AP	DA	BSSID	SA	-
1	0	To AP	BSSID	SA	DA	-

Tabelle 2.6: Verwendung der Adressfelder in Datenpaketen

Das Feld *Sequence Control* beinhaltet eine Fragment- und eine Sequenznummer, die Auskunft über die Reihenfolge von fragmentierten und nicht fragmentierten Daten- oder Management-Paketen geben. Jedes Paket erhält eine laufende Sequenznummer von 0 bis 4095 und jedes Paketfragment eine laufende Fragmentnummer. Diese bleiben bei wiederholten Übertragungen konstant. Für alle Datenpakete des Subtyps *QoS Data* existiert des Weiteren das Feld *QoS Control*. Dieses ist für die Steuerung von QoS-Übertragungen notwendig und bietet beispielsweise Informationen zur Identifizierung der QoS-Kategorie beziehungsweise des QoS-Stroms zu dem das Paket gehört.

2.5.3 Management

Der Standard spezifiziert im Rahmen der Protokollarchitektur drei verschiedene Management-Einheiten. Diese werden als *Physical Layer Management Entity* (PLME), *MAC Sublayer Management Entity* (MLME) und *Station Management Entity* (SME) bezeichnet, siehe Abbildung 2.3. Die Hauptaufgaben der Management-Einheiten sind der Aufbau von Verbindungen, die Gewährleistung einer sicheren Kommunikation, wie sie im Vergleich zu kabelgebundenen Netzen besteht, und die Bereitstellung zusätzlicher Energiesparmechanismen im Hinblick auf mobile Stationen mit begrenzter Energieversorgung. Alle diese Aufgaben werden durch den Austausch von *Management-Frames* zwischen MAC-Einheiten einzelner Stationen erzielt. Der grundsätzliche Aufbau eines Management-Frames ist in Abbildung 2.13 dargestellt. Im Gegensatz zu Datenpaketen ist die

Bedeutung der Adressfelder eindeutig und nicht vom *Subtype* des Frames abhängig. Das *Duration*-Feld gibt, genau wie bei RTS- und CTS-Paketen, die zu reservierende Dauer des NAVs für andere Stationen an.

MAC Header						**Frame Body**	**FCS**
Frame Control 0000...	Duration	DA	SA	BSSID	Sequence Control		
2 Bytes	*2 Bytes*	*6 Bytes*	*6 Bytes*	*6 Bytes*	*2 Bytes*	*0...2312 Bytes*	*4 Bytes*

Abbildung 2.13: Aufbau eines Management-Frames

Innerhalb des Feldes *Frame Body* sind abhängig vom *Subtype* des Management-Frames verschiedene Informationen für die Durchführung der Management-Aufgaben enthalten. Diese Informationen können in Feldern fester Größe oder in *Information Elements* (IE) mit variabler Größe angegeben werden. Die Felder fester Größe geben meist grundlegende Informationen für den Verbindungsaufbau an, während die IEs erweiterte Parameter und Informationen für spezielle Management-Funktionen bereitstellen. Grundlegende Informationen werden beispielsweise durch die Felder *Beacon Interval*, *Capability Information* oder *Timestamp* bereitgestellt. Oft verwendete IEs sind beispielsweise das *SSID Element*, das *Supported Rates Element*, das *TIM Element* oder das *RSN Information Element*. Jedes IE besitzt dabei die drei Felder *Type*, *Length* und *Information*, siehe Abbildung 2.14.

Element ID	**Length**	**Information**
1 Byte	*1 Byte*	*0 - 256 Bytes*

Abbildung 2.14: Aufbau eines Information-Elements

Da das vier Bit große *Subtype*-Feld nur eine Definition von 16 Management-Frames zulässt, definiert der Standard seit der Erweiterung 802.11h (2003) sogenannte *Action-Frames*, die eine Erweiterung der Management-Funktionen erlauben. Bisher spezifiziert der Standard die vier verschiedenen Action-Kategorien *Spectrum Management*, *QoS*, *DLS* und *Block Ack*. Jede Kategorie erlaubt es wiederum verschiedene IEs für die Durchführung von Management-Aufgaben anzugeben. Empfängt eine Station einen *Unicast Action Frame* mit einer unbekannten Action-Kategorie, soll die Station laut Standard den *Action Frame* an den Sender zurückschicken.

Verbindungsaufbau

Der Verbindungsaufbau lässt sich noch einmal in die drei Schritte Auffinden von Stationen (*Scanning*), Authentisierung (*Authentication*) und Anmeldung (*Association*) unterteilen. Das Auffinden von Stationen, beziehungsweise von vorhandenen Netzen, basiert auf der periodischen Versendung von *Beacons* zu einer vordefinierten *Target Beacon Transmission Time* (TBTT). Als *Beacon* wird ein spezieller Management-Frame bezeichnet, welcher grundlegende Informationen für den Verbindungsaufbau und somit den Eintritt in ein BSS oder IBSS enthält. Die genaue Angabe des Zeitintervalls zwischen zwei TBTTs ist in jedem Beacon als Anzahl an *Time Units* (TU)[4] enthalten. Innerhalb eines Infrastruktur-BSS ist der Access Point für das Versenden von Beacons zuständig, während bei einem IBSS durch ein Zufallsverfahren eine beliebige Station das nächste Beacon versendet. Dieses Zufallsverfahren ist vergleichbar mit dem Backoff-Prozess der DCF. Zu

[4]Eine TU entspricht 1024 µs.

jedem Zeitpunkt, an dem das nächste Beacon versendet werden soll, wählen alle Stationen eine zufällige Wartezeit aus dem Bereich $[0, aCWmin \cdot aSlotTime \cdot 2]$. Die Station mit der kürzesten Wartezeit sendet das nächste Beacon, während alle anderen den Vorgang abbrechen. Das Auswerten von Informationen aus empfangenen Beacons wird als *Passive Scanning* bezeichnet. Die Stationen haben allerdings auch die Möglichkeit durch einen *Probe Request* diese Informationen explizit anzufordern. Dieser Vorgang wird als *Active Scanning* bezeichnet. Ein AP, beziehungsweise innerhalb eines IBSS die Station, die das letzte Beacon versendet hat, antworten mit einem *Probe Response*. Dieser Frame enthält wiederum die gleichen Informationen wie ein normales Beacon.

Der nächste Schritt zum Aufbau einer Verbindung ist die Authentisierung. Diese beschreibt die Erbringung eines Nachweises über die Identität einer Station gegenüber einer anderen Station oder eines APs. Die erfolgreiche Authentisierung ist die notwendige Bedingung bevor eine Station sich an einem AP anmelden kann. Das Konzept der Anmeldung (*Association*) ist eine Voraussetzung, um das *Distribution System* (DS) mit den nötigen Informationen für die Übermittlung von Nachrichten zu versorgen. Hierzu gehört beispielsweise der *Service Set Identifier* (SSID) zur eindeutigen Kennzeichnung eines ESS oder IBSS. Die Anmeldung kann nur durch eine Station selbst initiiert werden und eine Station kann zu jedem Zeitpunkt nur an einem AP angemeldet sein. Um eine sichere *Association* innerhalb eines *Robust Security Network* (RSN) herzustellen, ist eine Authentisierung basierend auf einem vorgeschriebenen *4-Way-Handshake* nötig. Soll eine Verbindung getrennt werden, wird entweder durch den AP oder durch die Station selbst eine *Disassociation*-Nachricht gesendet.

Zeitsynchronisierung

Der Mechanismus zur Synchronisierung der lokalen Zeitgeber aller Stationen in einem BSS oder IBSS wird als *Timing Synchronization Function* (TSF) bezeichnet. Innerhalb eines Infrastruktur-BSS ist der Access Point der zentrale Zeitgeber. Dieser übermittelt seinen aktuellen Zeitstempel mit jedem Beacon an alle Station in Reichweite, die ihre lokalen Zeitgeber bei einer Abweichung anpassen. Bei einem IBSS wird die TSF verteilt bestimmt. Eine Station, die nach der im letzten Abschnitt beschriebenen Zufallsmethode das nächste Beacon oder die nächste Probe Response sendet, übermittelt ihren aktuellen Zeitstempel. Alle Stationen des IBSS, die das Beacon empfangen und deren aktueller Zeitstempel älter als der Empfangene ist, passen ihre Zeit an. Die Synchronität der Zeitgeber ist besonders wichtig für den im nächsten Abschnitt beschriebenen Energiesparmechanismus sowie für Modulationstechniken basierend auf FHSS.

Energiesparmechanismen

Der 802.11-Standard stellt einen Mechanismus bereit, der es Stationen innerhalb eines BSS oder IBSS erlaubt ihre Energiereserven aufzusparen (*Power Saving*). Hierzu kann eine Station in einen Schlafmodus übergehen, während dem sie keine Funksignale senden oder empfangen kann. Innerhalb eines BSS muss eine Station, die in den Schlafmodus wechseln will, zuvor den Access Point benachrichtigen. Der AP speichert daraufhin alle eingehenden Pakete für diese Station in einem Puffer. In regelmäßigen Abständen wacht die Station auf und empfängt ein Beacon des APs mit einer integrierten *Traffic Indication Map* (TIM). Diese TIM informiert eine Station anhand ihres *Association Identifiers* (AID) über gespeicherte Pakete innerhalb des APs, welche daraufhin durch die Station mit einer *PS-Poll*-Nachricht von dem AP angefordert werden können.

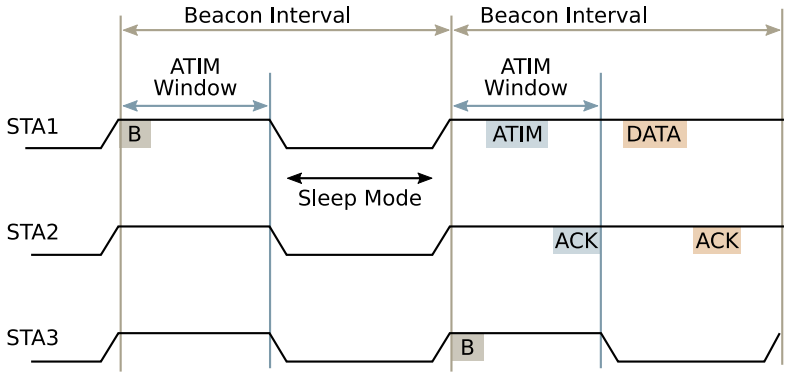


Abbildung 2.15: Energiesparmechanismus in einem IBSS

Während der AP in einem Infrastruktur-BSS die zentrale Instanz für die Zwischenspeicherung von Paketen ist, muss diese Aufgabe in einem IBSS verteilt gelöst werden. Die Grundlage hierfür ist die Angabe eines *Announcement Traffic Indication Message* (ATIM) *Windows*, während dessen sich keine Station im Schlafmodus befinden darf. Das ATIM Window hat eine fest vorgegebene Dauer, welche durch die initiale Station des IBSS bestimmt wird und in jedem Beacon, beziehungsweise jeder Probe Response, enthalten ist. Während dieser Zeitdauer, die unmittelbar nach jeder TBTT folgt, können Stationen, die ein Paket zu versenden haben, eine ATIM-Nachricht verschicken. Diese Nachricht signalisiert dem Empfänger, dass für ihn Pakete anstehen und dieser somit für die Dauer eines Beacon-Intervalls nicht in den Schlafmodus übergehen darf. Nach Ablauf des ATIM Windows werden die Pakete unter Verwendung der normalen DCF versendet. Damit ATIM-Nachrichten nicht mit Beacons kollidieren können, dürfen diese erst nach Empfang oder Übertragung eines Beacons und einem zufälligen Backoff aus $[0, aCWmin]$ gesendet werden. In Abbildung 2.15 ist der Ablauf des Mechanismus mit drei Stationen dargestellt. Während der ersten TBTT sendet STA1 das Beacon und da keine Station ein Paket zu versenden hat, gehen alle nach Ablauf des folgenden ATIM Windows wieder in den Schlafmodus über. Zur zweiten TBTT versendet STA3 das Beacon und STA1 hat nun ein Paket für STA2 vorliegen. Daher sendet STA1 eine ATIM-Nachricht an STA2. Beide Stationen bleiben bis zum Ablauf des nächsten ATIM Windows wach und tauschen die anstehenden Pakete aus.

Frequenzspektrum-Management

Die 2003 verabschiedete Erweiterung 802.11h [56] spezifiziert verschiedene Mechanismen, um Kanäle im 5-GHz-Bereich, wie bei 802.11a und 802.11n, auch in Europa verwenden zu dürfen. Der erste Mechanismus, *Transmit Power Control* (TPC), war ursprünglich für die Einhaltung europäischer Leistungsgrenzwerte vorgesehen, um eine Störung von Satelliten- oder Radarsignalen im 5-GHz-Bereich zu verhindern. Dieser Mechanismus bringt aber auch für andere Frequenzbänder den Vorteil, dass Stationen, basierend auf vorherigen Messungen, nur mit minimal notwendiger Leistung senden müssen. Die Sendeleistung kann dabei pro Paket dynamisch angepasst werden. Neben TPC spezifiziert 802.11h mit *Dynamic Frequency Selection* (DFS) einen Mechanismus zum dynamischen Wechsel des Übertragungskanals, falls auf dem aktuellen Kanal ein Radarsignal festgestellt wurde. Der DFS-Mechanismus ist in Europa von dem *European Telecommunications Standards Institute* (ETSI) laut EN 301 893 [38] für die Frequenzbereiche von 5,25 GHz bis 5,35 GHz und von 5,47 GHz bis 5,725 GHz vorgeschrieben. Um auf Kanälen in diesen Frequenzbereichen zuverlässige Messungen durchführen zu können, müssen Beacons oder Probe Responses ein soge-

nanntes *Quiet Element* enthalten. Dieses *Information Element* gibt ein Zeitintervall an, in dem keine Station eines BSS senden sollte. Das Zeitintervall wird innerhalb des Quiet-Elements durch die Felder *Quiet Count*, *Quiet Period*, *Quiet Duration* und *Quiet Offset* angegeben, siehe Abbildung 2.16. *Quiet Count* gibt die Anzahl der TBTTs bis zum Start des Zeitintervalls an. Falls sich das Zeitintervall periodisch wiederholen soll, kann dies durch die Angabe der Anzahl an Beacon-Intervallen im Feld *Quiet Period* geschehen. Das Feld *Quiet Duration* spezifiziert die eigentliche Dauer des Intervalls in TUs, für die eine Station ihren NAV reservieren sollte. Das Feld *Quiet Offset* kann zusätzlich eine Verzögerung nach dem Startpunkt angeben, die kürzer als ein Beacon-Interval sein muss. Der Standard erlaubt eine beliebige Anzahl an Quiet-Elementen innerhalb eines Beacons. Somit kann ein Beacon verschiedene Quiet-Intervalle definieren.

Element ID	**Length**	**Quiet Count**	**Quiet Period**	**Quiet Duration**	**Quiet Offset**
1 Byte	*1 Byte*	*1 Byte*	*1 Byte*	*2 Bytes*	*2 Bytes*

Abbildung 2.16: Aufbau eines Quiet-Elements

Wurde nun durch Messungen während der Quiet-Zeitdauer die Existenz eines Radars festgestellt, muss der Kanal des BSSs gewechselt werden. Dieser Wechsel wird in einem Infrastruktur-BSS als auch IBSS durch ein *Channel Switch Announcement Information Element* angekündigt, siehe Abbildung 2.17. Dieses IE kann entweder in einem Beacon, einer Probe Response oder einem Action-Frame enthalten sein

Element ID	**Length**	**Channel Switch Mode**	**New Channel Number**	**Channel Switch Count**
1 Byte	*1 Byte*	*1 Byte*	*1 Byte*	*1 Byte*

Abbildung 2.17: Aufbau eines Channel Switch Announcement Elements

Das Feld *Channel Switch Mode* kann entweder auf 0 oder 1 gesetzt werden. Der Wert 1 signalisiert einer Station, dass sie bis zum angegebenen Zeitpunkt des Kanalwechsels keine weiteren Pakete versenden soll. Der Wert 0 signalisiert hingegen keine besonderen Maßnahmen bis zum Kanalwechsel. Das Feld *New Channel Number* gibt den neuen Kanal an, zu dem gewechselt werden soll. Das letzte Feld *Channel Switch Count* gibt die Anzahl der TBTTs an, die bis zum Zeitpunkt des endgültigen Kanalwechsels noch vergehen werden.

2.6 Sicherheit

Allgemein lässt sich der Begriff der Sicherheit im Bereich von Computersystemen in die vier Sicherheitsanforderungen *Vertraulichkeit*, *Integrität*, *Authentizität* und *Verfügbarkeit* unterteilen [16]. Der 802.11-Standard unterscheidet in seiner aktuellsten Version im Bereich der Sicherheit zwischen zwei Typen von Netzwerken, dem Pre-Robust Security Network und dem *Robust Security Network* (RSN). Das RSN beinhaltet alle Sicherheitserweiterungen der IEEE Arbeitsgruppe 802.11i [57] und ist Bestandteil des aktuellen Standards von 2007. Eine Übersicht der beiden Typen und deren beinhalteten Sicherheitsmechanismen ist in Abbildung 2.18 auf der nächsten Seite zu sehen.

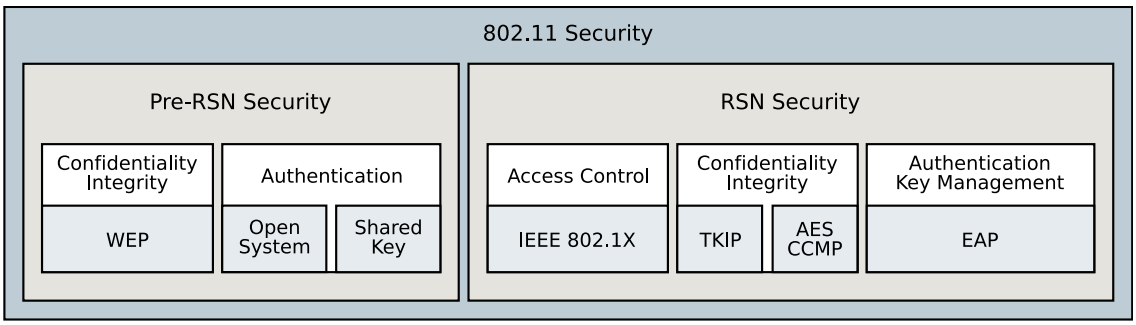


Abbildung 2.18: Übersicht der Sicherheitsmechanismen von Pre-RSNs und RSNs [109]

2.6.1 Vertraulichkeit und Integrität

Unter der Vertraulichkeit versteht man die Sicherheitsanforderung, dass während einer Kommunikation keine Daten von Dritten mitgelesen werden können. Die Integrität beschreibt die Anforderung, eine unbemerkte Manipulation von kommunizierten Daten durch Dritte zu verhindern. Um die Vertraulichkeit zu garantieren, spezifiziert der Standard drei Protokolle basierend auf verschiedenen Verfahren der kryptographischen Verschlüsselung: *Wired Equivalent Privacy* (WEP), *Temporal Key Integrity Protocol* (TKIP) und *Counter Mode with CBC-MAC Protocol* (CCMP).

WEP ist das erste Sicherheitsprotokoll des ursprünglichen Standards von 1997 und basiert auf der Stromchiffre RC4 [119]. Hierbei werden Nachrichten mit einem gemeinsamen Schlüssel (*shared Key*) verschlüsselt. Dieser Schlüssel wird mit einem 24 Bit *Initialization Vector* (IV) verknüpft, um einen frischen RC4-Schlüssel für jedes Paket zu erhalten. Die Integrität soll durch die Berechnung eines *Integrity Check Value* (ICV) mit Hilfe eines einfachen *Cyclic Redundancy Checks* (CRC) gewährleistet werden. Der *Frame Body* wird zusammen mit dem ICV durch den zuvor bestimmten RC4-Schlüssel verschlüsselt. WEP weist allerdings einige Schwachstellen auf, die in zahlreichen wissenschaftlichen Arbeiten bereits ausführlich diskutiert wurden [118, 43, 8, 18, 112, 9, 31]. Diese Schwachstellen haben zur Folge, dass weder die Vertraulichkeit noch die Integrität durch WEP garantiert werden kann. Gründe hierfür sind die Längenbeschränkung des *Shared Key* von nur 40 Bit, eine zu geringe Größe des IVs, eine Schwachstelle der RC4-Stromchiffre und die lineare Berechnung des unverschlüsselten ICVs. So ist es durch relativ geringen Aufwand möglich Nachrichten zu entschlüsseln oder unbemerkt zu verändern. Im aktuellen Standard von 2007 wird WEP daher als *deprecated* gekennzeichnet und sollte, wenn möglich, nicht weiter verwendet werden.

Neue Verfahren, die im Rahmen der Erweiterung 802.11i entwickelt wurden, um die bekannten Schwachstellen von WEP zu beheben, sind das ebenfalls auf RC4 basierende *Temporal Key Integrity Protocol* (TKIP) und das auf dem *Advanced Encryption Standard* (AES) aufbauende *Counter Mode with CBC-MAC Protocol* (CCMP). Da die *Wi-Fi Alliance* nach Aufkommen der ersten Schwachstellen von WEP eine schnelle Lösung suchte, wurde ein Teil des damaligen 802.11i-Entwurfs, darunter TKIP, in das Zertifizierungsverfahren mit dem Namen *Wi-Fi Protected Access* (WPA) aufgenommen. Bei der Entwicklung von TKIP wurde besonders das Designziel verfolgt, die bekannten Schwachstellen lediglich durch Software- oder Firmwareupdates bei bestehender Hardware zu beseitigen. Die Vertraulichkeit wird bei TKIP ebenfalls durch eine RC4-Stromverschlüsselung erreicht. Allerdings wird eine *Key Mixing* Funktion und ein erweiterter Raum für IVs eingesetzt, um für jedes Paket einen neuen Schlüssel bereitstellen zu können. Die Integrität wird des Weiteren durch einen *Message Integrity Code* (MIC) mit dem Namen *Michael*

garantiert [40]. Seit kurzem gilt auch TKIP nicht mehr als sicher. Ein neues Verfahren ermöglicht bereits nach 15 minütiger Datenaufzeichnung ARP-Pakete zu entschlüsseln und beliebige Pakete in ein TKIP-verschlüsseltes Netz einzuschleusen [12].

Das dritte Verfahren zur Sicherstellung der Vertraulichkeit und Integrität ist CCMP, das den *Counter Mode* (CTR) für die Gewährleistung der Vertraulichkeit mit dem *Cipher-Block Chaining Message Authentication Code* (CBC-MAC) zum Schutz der Integrität [122] kombiniert. Der Modus *Counter Mode with CBC-MAC* dient somit als Ersatz für RC4 und *Michael*. Hierbei kommt der AES Verschlüsselungsalgorithmus [111] mit einem 128 Bit Schlüssel und einer Blockgröße von ebenfalls 128 Bit zum Einsatz. Während bei WEP und TKIP ein Schlüssel pro Paket erzeugt werden muss, ist es mit AES möglich alle Pakete einer Sitzung mit dem selben 128 Bit Schlüssel zu chiffrieren und gleichzeitig eine verbesserte Sicherheit zu erreichen. Die *Wi-Fi Alliance* hat als WPA-Nachfolger das Zertifizierungsprogramm WPA2 ins Leben gerufen, welches die ausschließliche Verwendung von CCMP vorsieht.

CCMP stellt eine aus heutiger Sicht solide Methode zur Sicherstellung von Vertraulichkeit und Integrität für Datenpakete der MAC-Schicht dar. Da der Standard die Verwendung von Verschlüsselungsmechanismen allerdings ausschließlich für Datenpakete vorgesehen hat, besteht für Control- sowie Management-Nachrichten bisher keinerlei Schutz der Vertraulichkeit und Integrität. Eine Manipulation dieser Nachrichten ist somit auch in CCMP-geschützten Netzen möglich. Die Arbeitsgruppe 802.11w [64] beschäftigt sich daher mit möglichen Erweiterungen des Standards, um ebenfalls einen Schutz für Management-Nachrichten zu gewährleisten. Eine Verabschiedung dieser Erweiterungen ist frühestens zum Ende des Jahres 2009 zu erwarten.

2.6.2 Authentizität und RSNA

Die Authentizität einer Station beschreibt die Echtheit und Glaubwürdigkeit, die anhand einer eindeutigen Identität in Verbindung mit einem zugehörigen Merkmal überprüft, beziehungsweise nachgewiesen wird. Der Vorgang der Überprüfung wird als Authentifizierung, der Vorgang des Nachweises als Authentisierung bezeichnet. Da die englische Bezeichnung *Authentication* nicht zwischen diesen Begriffen unterscheidet, wird ab dieser Stelle ebenfalls auf die konkrete Unterscheidung verzichtet und lediglich der Begriff der Authentisierung verwendet. Der Standard spezifiziert hierzu in seiner ersten Version die beiden Varianten *Open System* und *Shared Key Authentication*. Erstere basiert ausschließlich auf der MAC-Adresse einer Station und bietet keine sichere Authentizität im eigentlichen Sinne, da die Adresse leicht manipuliert werden kann. Verschiedene Arbeiten haben gezeigt, dass auch die *Shared Key Authentication*, basierend auf dem vordefinierten WEP-Schlüssel, keine zuverlässige Methode für die Authentisierung darstellt [9, 18].

Die Erweiterung 802.11i definiert daher das Vorgehen zum Aufbau einer *Robust Security Network Association* (RSNA), das eine sichere Methode zur gegenseitigen Authentisierung, Zugriffskontrolle und zur Generierung eines neuen Schlüssels für die verwendeten Sicherheitsprotokolle bereitstellt. Der Aufbau einer RSNA besteht aus Protokollen zum Schlüsselmanagement und zur Authentisierung basierend auf dem Standard 802.1X für die portbasierte Zugriffskontrolle in Netzwerken [58]. Drei Einheiten sind während des Aufbaus involviert: der *Supplicant*, der *Authenticator* und der *Authentication-Server*, zum Beispiel ein RADIUS [100] Server. Innerhalb eines Infrastruktur BSS ist der Access Point der Authenticator und jede Station stellt einen Supplicant dar. In einem IBSS ist hingegen jede Station gleichzeitig Supplicant und Authenticator. Eine erfolgreiche Authenti-

sierung bedeutet, dass sich der Supplicant und der Authenticator gegenseitig authentifizieren und der Authentication-Server ein gemeinsames Geheimnis, den *Master Session Key* (MSK), für die nachfolgende Aushandlung eines Schlüssels generiert. Der vollständige Ablauf zum Aufbau einer RSNA ist in Abbildung 2.19 dargestellt und lässt sich in folgende Phasen unterteilen [53]:

1. **Network and Security Capability Discovery:** In der ersten Phase wertet eine Station die Sicherheitsinformationen eines Beacons, beziehungsweise einer Probe Response, aus. Diese sind in einem RSN Information Element enthalten und beschreiben die unterstützten Sicherheitsmechanismen eines Access Points oder einer anderen Station.

2. **Pre-RSNA Authentication and Association:** Die zweite Phase besteht aus der *Open System Authentication* und dem üblichen Anmeldevorgang wie er durch den ursprünglichen Standard vorgesehen ist. Dieser Schritt dient vorrangig der Abwärtskompatibilität. Eine Station sendet dabei ihr RSN Information Element mit dem *Association Request*. Supplicant und Authenticator sind nach einem erfolgreichen Abschluss dieses Vorgangs theoretisch authentisiert und verbunden. Da die *Open System Authentication* aber nur auf der MAC-Adresse des Supplicants beruht und diese leicht gefälscht werden kann, ist die bestehende Authentizität als schwach einzustufen. Die 802.1X Ports bleiben daher zunächst blockiert und es kann noch keine Datenkommunikation stattfinden.

3. **802.1X Authentication:** In dieser Phase führen der Supplicant und der Authentication-Server ein gegenseitiges Authentisierungsprotokoll aus, wobei der Authenticator deren Nachrichten aneinander weiterleitet. Für diesen Zweck wird das *Extensible Authentication Protocol* (EAP) [3] eingesetzt, das die Möglichkeit bietet verschiedene Authentisierungsmethoden wie beispielsweise EAP-TLS [4] zu kapseln. Für die Benutzung von EAP in LANs spezifiziert 802.1X die Variante *Extensible Authentication Protocol over LANs* (EAPOL). Nach erfolgreicher Authentisierung sind der Supplicant sowie der Authenticator in Besitz eines *Master Session Key* (MSK), der durch den Authentication-Server generiert wurde. Dieser ermöglicht die Generierung eines *Pairwise Master Key* (PMK) für den 4-Way-Handshake im nächsten Schritt. Alternativ kann der PMK auch durch einen statischen *Preshared Key* (PSK) ersetzt werden, um diesen Schritt zu überspringen.

4. **4-Way-Handshake:** Der *4-Way-Handshake* ist die Prozedur, die vor jedem Aufbau einer RSNA zwingend durchgeführt werden muss. In dieser Phase bestätigen sich Supplicant und Authenticator durch den Austausch von vier Nachrichten gegenseitig die Existenz des PMKs, überprüfen die Auswahl der gemeinsamen *Cipher Suite*[5] und bestimmen einen neuen *Pairwise Transient Key* (PTK) für die folgende Datenkommunikation. Wichtige Elemente, die dabei ausgetauscht werden, sind die jeweiligen Adressen (AA, SPA), RSN Information-Elements (A/SP RSN IEs) und Nonces (ANonce, SPNonce). Der Authenticator hat zusätzlich die Möglichkeit einen *Group Temporal Key* (GTK) für Multicast- oder Broadcast-Nachrichten zu übermitteln. Nach erfolgreichem Abschluss des 4-Way-Handshakes werden die 802.1X Ports geöffnet und es kann eine Datenkommunikation erfolgen.

5. **Group Key Handshake:** Falls während der letzten Phase kein GTK übermittelt wurde, muss der Authenticator einen neuen GTK generieren und an den Supplicant weiterleiten.

6. **Secure Data Communication:** Die letzte Phase symbolisiert den sicheren Datenaustausch zwischen Supplicant und Authenticator unter Verwendung der zuvor ausgehandelten *Cipher Suite* und des PTK oder GTK.

[5] Die *Cipher Suite* spezifiziert die verfügbaren Mechanismen und Protokolle einer Station zur Gewährleistung der Sicherheit.

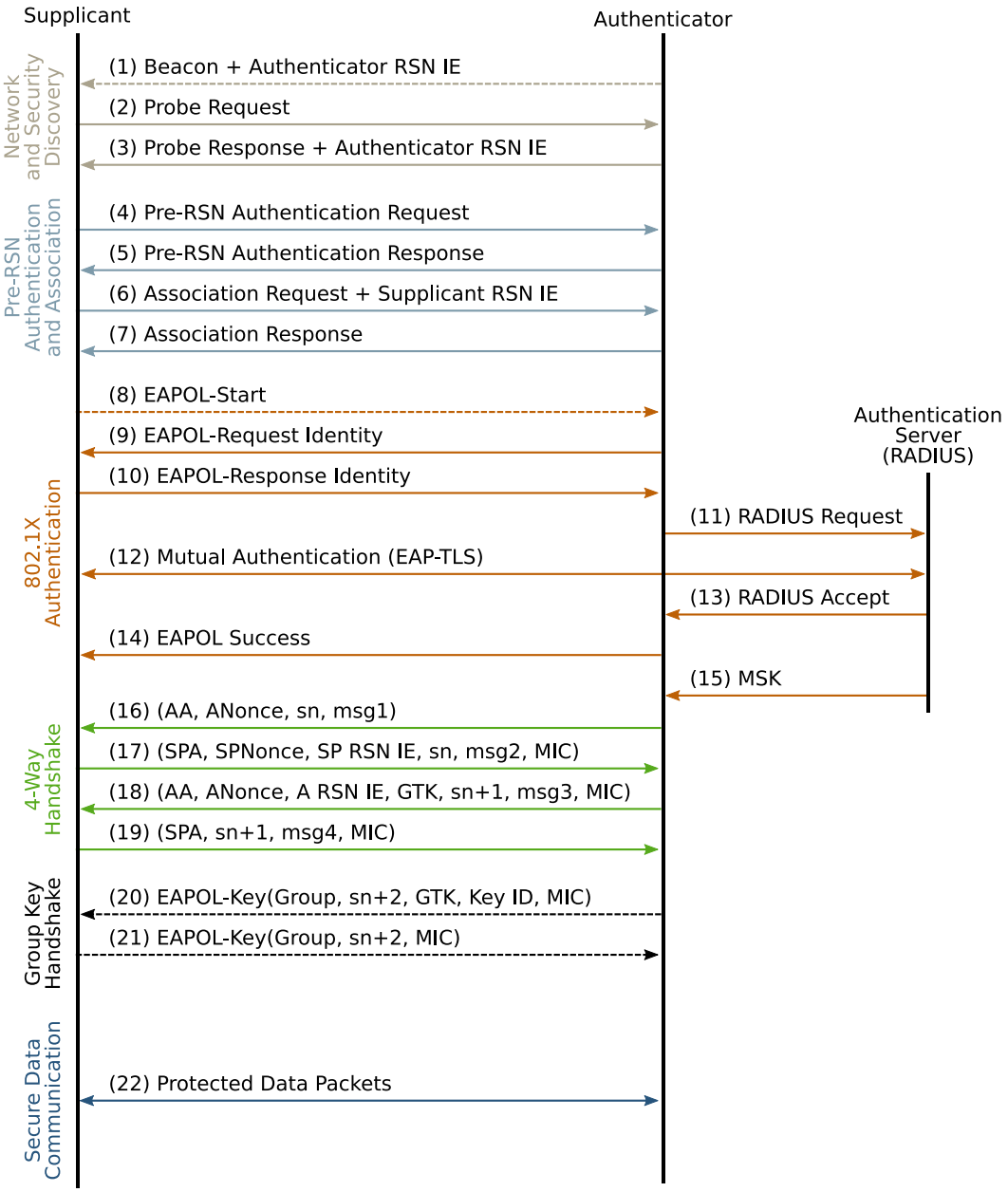


Abbildung 2.19: Ablauf zum Aufbau einer RSNA [53]

Unter der Voraussetzung, dass der komplette Ablauf zum Aufbau einer RSNA wie beschrieben durchgeführt wird, kann die Authentisierung und das Schlüsselmanagement als sicher eingestuft werden. Die Wahl eines PSK als PMK kann dabei die Sicherheit wiederum vermindern, da dies die Verwundbarkeit durch Wörterbuch-Angriffe erhöhen würde [80].

2.6.3 Verfügbarkeit

Allgemein beschreibt die Verfügbarkeit die Anforderung auf eine Information oder Ressource uneingeschränkt zugreifen zu können. Die Verfügbarkeit stellt damit einen besonders wichtigen Aspekt der Sicherheit im Hinblick auf Zuverlässigkeit dar, denn ein nicht verfügbares System ist ebenso gut oder schlecht wie ein nicht vorhandenes System. Bei der Entwicklung des 802.11-Standards stellte die Gewährleistung der Verfügbarkeit allerdings kein zentrales Designziel dar, welches eine Vielzahl von möglichen Angriffspunkten eröffnet hat. Selbst die Verwendung von Protokollen für Vertraulichkeit und Integrität mit höchster Sicherheitsstufe bietet keinen Schutz vor Angriffen gegen die Verfügbarkeit und kann sogar im Gegenteil weitere Angriffsmöglichkeiten eröffnen. Da bei kabellosen Netzen keine physischen Grenzen existieren, sind naive Angriffe gegen die Verfügbarkeit relativ leicht durchzuführen und stellen eine große Bedrohung dar. Obwohl einige derartiger Angriffe durch Anpassung des 802.11-Standards verhindert werden könnten, existieren bis heute noch keine Geräte, die einen derartigen Schutz gegen mögliche Angriffe bieten.

2.7 Zusammenfassung

Der IEEE Standard 802.11 für kabellose lokale Netze umfasst eine große Anzahl an Mechanismen und Protokollen auf der physikalischen Ebene sowie der MAC-Ebene. Seit Beginn der Standardisierung sind im Laufe der Jahre zahlreiche Erweiterungen hinzugekommen. Durch den immer größer werdenden Anspruch an steigender Bandbreite, höheren Reichweiten oder der Koexistenz mit anderen Funktechniken im gleichen Frequenzband, wird auch die Anzahl an Erweiterungen und die damit verbundene Komplexität des Standards in Zukunft noch weiter steigen. Grundsätzlich ist mit einer steigenden Komplexität auch immer die Gefahr neu entstehender Sicherheitslücken oder Angriffspunkte verknüpft.

Dieses Kapitel hat einen umfassenden Überblick über den derzeitigen Standard (Stand 2007) mit seinen wichtigsten Erweiterungen gegeben. Hierbei wurden insbesondere Erweiterungen vorgestellt, die eine Durchführung der im nächsten Kapitel diskutierten Angriffe gegen die Verfügbarkeit ermöglichen. Tabelle 2.7 auf der nächsten Seite zeigt noch einmal eine zeitliche Einordnung der vorgestellten Mechanismen des Standards und deren Relevanz für die späteren Angriffe gegen die Verfügbarkeit auf.

Mechanismus	Standard/Erweiterung	Jahr	Relevant für Angriffe
PHY			
Clear Channel Assessment	802.11	1997	•
FHSS	802.11	1997	
DSSS	802.11	1997	
OFDM	802.11a	1999	
MIMO	802.11n	2009	
MAC			
DCF	802.11	1997	
Interframe Spaces	802.11	1997	•
Block Acknowledgement	802.11e/n	2005/2009	•
Backoff-Prozess	802.11	1997	•
Virtual CS (RTS/CTS)	802.11	1997	•
PCF	802.11	1997	
HCF	802.11e	2005	
EDCA (AIFS)	802.11e	2005	•
HCCA	802.11e	2005	
Management			
Verbindungsaufbau	802.11	1997	•
Zeitsynchronisation	802.11	1997	•
Energiesparmechanismen	802.11	1997	•
DFS	802.11h	2003	•
Sicherheit			
WEP	802.11	1997	
TKIP	802.11i	2004	•
CCMP	802.11i	2004	•
802.X Authentication	802.11i	2004	•
4-Way Handshake	802.11i	2004	•

Tabelle 2.7: Übersicht der vorgestellten Mechanismen des aktuellen 802.11-Standards, deren zeitliche Einordnung und Relevanz für Angriffe gegen die Verfügbarkeit.

3 Angriffe gegen die Verfügbarkeit

Seit Verabschiedung des ersten 802.11-Standards 1997, haben sich bereits viele Forschungsarbeiten mit dessen Sicherheit auseinander gesetzt. Die ersten Arbeiten beschäftigten sich verstärkt mit der Analyse des WEP-Protokolls, wodurch verschiedene Sicherheitslücken aufgedeckt wurden und sich WEP zur Gewährleistung der Vertraulichkeit sowie Integrität als unzureichend erwies [118, 43, 8, 18, 112, 9, 31]. Für eine genauere Darstellung der Schwachstellen von WEP sei an dieser Stelle auf die angegebene Literatur verwiesen.

Mit der zunehmenden Verbreitung von 802.11-Netzen vor allem in öffentlichen Bereichen und einer stetig steigenden Anzahl unterschiedlicher Einsatz- und Anwendungsgebiete, erhielt auch die Anforderung an die Verfügbarkeit eines WLANs größere Bedeutung. Die Untersuchung dieser Sicherheitsanforderung entwickelte sich daher zu einem zentralen Thema in vielen späteren Forschungsarbeiten und stellt bis heute ein wichtiges Gebiet im Bereich der Sicherheit dar [52, 13, 46, 34, 92, 47, 109]. Die Arbeiten, die sich in diese Kategorie einordnen lassen, beschäftigen sich hauptsächlich mit der Aufdeckung neuer Angriffsmöglichkeiten auf die Verfügbarkeit eines kabellosen Netzes und stellen teilweise Erweiterungen oder Änderungen des Standards vor, die derartige Angriffe erkennen oder auch abwehren sollen.

Mögliche Angriffe können einerseits die Verfügbarkeit komplett kompromittieren und somit das Ziel verfolgen eine Kommunikation zu unterbinden. Andererseits können sie die Verfügbarkeit beeinträchtigen und das Ziel verfolgen einen eigenen Vorteil wie höheren Datendurchsatz zu erreichen. Angriffe der ersten Kategorie werden als *Denial of Service* (DoS) Angriffe bezeichnet, die der zweiten Kategorie als unfaires Verhalten oder auch *Greedy Behaviour*.

Dieses Kapitel soll einen Überblick über die zur Zeit bekannten Angriffsmöglichkeiten und den aktuellen Stand der Forschung in diesen beiden Kategorien im Zusammenhang mit 802.11-Netzen geben. Gleichzeitig werden einige neue Angriffe vorgestellt, die Mechanismen der MAC-Schicht ausnutzen, um die Verfügbarkeit eines 802.11-Netzes zu kompromittieren. Zuvor wird kurz auf die Motivationen eines Angreifers und weitere Bedrohungen für WLANs eingegangen, die letztlich auch zu einer Bedrohung der Verfügbarkeit führen können. Außerdem werden Bewertungskriterien und Möglichkeiten für die Einordnung von Angriffen vorgestellt.

3.1 Motivation eines Angreifers

Wie bereits erwähnt kann ein Angreifer durch die später diskutierten Angriffe gegen die Verfügbarkeit zwei verschiedene Ziele verfolgen. Entweder möchte er die Kommunikation unterbrechen (*Denial of Service*) oder einen eigenen Vorteil erreichen (*Greedy Behaviour*). In den folgenden Abschnitten werden diese beiden Begriffe kurz erläutert.

- **Denial of Service**

  Der Begriff *Denial of Service* wird heute meist mit dem Bereich des Internets verbunden. In diesem Zusammenhang sind solche Angriffe gemeint, die Schwachstellen in der Implementierung der Netzwerkfunktionalität verschiedener Betriebssysteme oder bestimmter Software ausnutzen. Meist werden hierbei gezielt Rechner mit einer Flut an bestimmten Datenpaketen überhäuft, so dass diese mit der Verarbeitung der Pakete überfordert sind. Die IETF[1] definiert einen DoS-Angriff als einen *„Angriff, bei dem ein oder mehrere Rechner ein Ziel auswählen und versuchen zu verhindern, dass dieses Ziel nützliche Arbeit verrichtet“.*

  Im Gegensatz dazu bezeichnet Denial of Service im Bezug auf kabellose Netze den Verlust der Verfügbarkeit des Mediums, beziehungsweise den Verlust der Möglichkeit zur Kommunikation. Dies kann für das gesamte Netz oder nur für einzelne Stationen gelten, denen ganz oder zeitweise die Kommunikationsmöglichkeit entzogen wird. Ein Großteil der später diskutierten Angriffe ist in der Literatur stets im Zusammenhang mit dieser Motivation zu finden.

- **Greedy Behaviour**

  Unter dem Begriff *Greedy Behaviour* werden in der Literatur egoistische oder unfaire Verhaltensmuster einer Station zusammengefasst, die zum Erlangen eines eigenen Vorteils eingesetzt werden. Der erwünschte Vorteil kann dabei ein verbesserter Datendurchsatz, eine verringerte Latenz oder ein verringerter Energieverbrauch sein. Die Maßnahmen zum Erlangen dieses Vorteils gehen meist auf Kosten anderer Stationen im selben Netz und können im schlimmsten Fall auch zu einem DoS-Effekt für diese Stationen führen. Viele aktuelle Forschungsarbeiten beschäftigen sich mit dieser Problematik und versuchen insbesondere Systeme für die Erkennung und Abwehr egoistischer Stationen zu entwickeln [50, 75, 78, 48, 98, 72, 74, 33].

Grundsätzlich können Angriffe, die primär das Ziel des *Denial of Service* verfolgen, auch zum Erlangen eines eigenen Vorteils eingesetzt werden. Umgekehrt sind Angriffe, die primär das Ziel des eigenen Vorteils verfolgen, meist mit einer erheblichen Benachteiligung der restlichen Stationen verbunden und erreichen somit einen ähnlichen DoS-Effekt. Während allerdings DoS-Angriffe durch beliebige Stationen in Reichweite eines Netzes durchgeführt werden können, sind Stationen, die ein unfaires Verhalten aufweisen, meist Teilnehmer eines bestimmten BSS.

3.2 Grundsätzliche Bedrohungen für WLANs

Neben der Bedrohung durch Angriffe gegen die Verfügbarkeit wie *Denial of Service* Angriffe und *Greedy Behaviour*, lassen sich nach He et al. [53] noch sechs weitere grundlegende Formen von Sicherheitsbedrohungen für WLANs finden, die aber letztlich ebenfalls zu einer Bedrohung der Verfügbarkeit führen können. Diese werden im Folgenden kurz zusammengefasst:

1. **Passive Eavesdropping/Traffic Analysis:** Die physische Beschaffenheit eines kabellosen Netzes erlaubt es einem Angreifer die Kommunikation innerhalb eines WLANs mit zu verfolgen und zu speichern. Auch wenn die Nachrichten verschlüsselt sind, kann ein Angreifer eventuell Teilinformationen gewinnen oder durch die kontinuierliche Analyse der Nachrichten mehr und mehr über die darin enthaltenen Informationen lernen. Dieses Vorgehen erlaubt

[1] http://tools.ietf.org/html/rfc4732#section-1

es bei der Verwendung von WEP, nach einer bestimmten Anzahl analysierter Datenpakete, den Schlüssel zu rekonstruieren und nachfolgende Nachrichten zu entschlüsseln. Bei Verwendung einer RSNA, wie sie in Abschnitt 2.6.2 auf Seite 28 vorgestellt wurde, besteht diese Bedrohung allerdings nicht mehr. Die Möglichkeit des Einsehens von Teilinformationen wie Adresse des Senders oder Empfängers einer Nachricht bleibt allerdings vorhanden.

2. **Message Injection/Active Eavesdropping:** Des Weiteren bietet die physische Eigenschaft eines kabellosen Netzes einem Angreifer die Möglichkeit, eigene Nachrichten in ein Netz einzuspeisen (*Injection*). Mit handelsüblichen Netzwerkkarten (NICs) und geeigneter Software ist es möglich jedes Feld einer Nachrichten fast uneingeschränkt anzupassen und die Übertragung von Nachrichten zu kontrollieren. Somit kann ein Angreifer beliebige Nachrichten generieren, aber auch zuvor empfangene Nachrichten erneut übertragen (*Replay*). Durch das kontrollierte Einspeisen von Nachrichten, die bei Empfängern eine bestimmte Antwort hervorrufen, kann ein Angreifer gezielt Informationen sammeln. Dieses Vorgehen wird als *Active Eavesdropping* bezeichnet. Bei Verwendung von CCMP (WPA2) zur Sicherung der Vertraulichkeit und Integrität sind zumindest Datenpakete von dieser Gefahr nicht mehr betroffen. Für Control- und Managementpakete bleibt aber die Gefahr auch bei Verwendung von CCMP weiterhin bestehen.

3. **Masquerading and Malicious AP:** Da, wie bereits erwähnt, die MAC-Adressen innerhalb der Pakete von einem Angreifer jederzeit ausgelesen werden können, kann dieser mit der Zeit alle gültigen Adressen eines Netzes erfahren. Im nächsten Schritt kann er seine eigene MAC-Adresse ändern und somit die Identität eines beliebigen Knotens vortäuschen (*Masquerading*). Innerhalb eines Infrastruktur-BSS kann ein Angreifer auch die Identität eines vorhandenen Access Points vortäuschen, indem er dessen MAC-Adresse sowie SSID annimmt und mit entsprechender Software (z.B. HostAP[2]) die benötigte Funktionalität bereitstellt.

4. **Session Hijacking:** Die Gefahr des *Session Hijacking* bezeichnet die fremde Übernahme einer rechtmäßigen Sitzung durch einen Angreifer nach erfolgreicher Authentisierung. Eine mögliche Vorgehensweise um dies zu erreichen, ist die bestehende Verbindung des Opfers zunächst zu unterbrechen und dann dessen Identität vorzutäuschen. Ein erfolgreicher Angriff umginge somit eine Authentisierung. Falls aber eine Verbindung basierend auf einer RSNA durch einen Angreifer übernommen wird, besteht für diesen wiederum das Problem der verschlüsselten Nachrichten. Die Integrität und Vertraulichkeit der Daten wird somit durch einen solchen Angriff nicht gefährdet.

5. **Man in the Middle:** Ein *Man in the Middle* (MitM)-Angriff beschreibt die Platzierung eines Angreifers zwischen zwei Teilnehmern einer legitimen Verbindung. Ein Angreifer hat, basierend auf den vorher beschriebenen Gefahren, verschiedene Möglichkeiten eine solche Platzierung zu erreichen. Vereinfacht ausgedrückt muss der Angreifer in einem Infrastruktur-BSS aus Sicht der Station den AP und aus Sicht des APs die Station vortäuschen. Der Angreifer hat danach die Möglichkeit die Kommunikation zu kontrollieren und kann Nachrichten somit weiterleiten oder verwerfen. Bei der Verwendung einer RSNA stellt ein solcher Angriff allerdings ebenfalls keine Bedrohung für Integrität und Vertraulichkeit dar. Lediglich die Verfügbarkeit kann durch das Verwerfen von Nachrichten beeinträchtigt werden.

6. **Message Deletion and Interception:** Das Zerstören von Nachrichten beschreibt die Möglichkeit eines Angreifers die korrekte Übertragung einer Nachricht zu verhindern. Dies kann durch das einfache Aussenden eines Störsignals geschehen, wie in Abschnitt 3.5 noch ausführlich erläutert wird. Ein Angreifer hat des Weiteren die Möglichkeit eine Nachricht abzufangen

[2] http://hostap.epitest.fi

(*Interception*). Dies bedeutet konkret, dass der Angreifer eine Nachricht korrekt empfängt, aber den Empfang beim eigentlichen Empfänger verhindert. Dies kann beispielsweise durch die Verwendung einer Richtantenne zur Störung des Empfängers oder durch einen MitM-Angriff realisiert werden.

Alle zuvor beschriebenen Gefahren stellen bei der Verwendung neuer Sicherheitsprotokolle wie CCMP (WPA2) zwar keine Bedrohung für die Vertraulichkeit und Integrität eines Netzes dar, bieten aber Potential für Angriffe gegen die Verfügbarkeit. Derartige Angriffe bedeuten somit ein hohes Sicherheitsrisiko für WLANs und sind daher der Schwerpunkt der folgenden Abschnitte.

3.3 Bewertungskriterien von Angriffen

Um Angriffe gegen die Verfügbarkeit bewerten und miteinander vergleichen zu können, sind verschiedene Kriterien nötig. Die Gewichtung dieser Kriterien ist abhängig von der Motivation und dem Anwendungsgebiet des jeweiligen Angriffs und muss daher von Fall zu Fall unterschieden werden. In der Literatur lassen sich einige Kriterien finden, die sich durch die Kriterien *Wahrscheinlichkeit der Entdeckung*, *Standardkonformität*, *Einfluss der Betriebsart* und *Genauigkeit* zu der folgenden Liste ergänzen lassen [6, 92]:

- **Energieeffizienz:** Die benötigte Energie, um einen Angriff durchzuführen, ist besonders bei mobilen Geräten mit begrenzter Stromversorgung von großer Bedeutung. Die einfachste Möglichkeit, um beispielsweise einen DoS-Angriff durchzuführen, besteht in der kontinuierlichen Übertragung eines Störsignals, auch *Constant Jamming* genannt, siehe Abschnitt 3.5. Dieses Vorgehen benötigt allerdings sehr viel Energie und ist somit besonders für mobile Angreifer ungeeignet. In Bezug auf DoS-Angriffe definieren Brown et al. [127] daher die Energieeffizienz als das *„proportionale Verhältnis zwischen der Energie, die benötigt wird, um einen bestimmten Effekt mit einem Constant Jamming Angriff zu erreichen, und der benötigten Energie des zu untersuchenden Angriffs, welcher den selben Effekt erreicht“*. Auch für das *Greedy Behaviour* kann die Energieeffizienz ein wichtiger Faktor sein. Innerhalb eines MANETs ist beispielsweise ein Vorgehen, das den doppelten Datendurchsatz erzielt zwecklos, falls die Energieressourcen dadurch halbiert werden.
- **Wahrscheinlichkeit der Entdeckung:** Je nach Motivation eines Angreifers kann die Wahrscheinlichkeit der Entdeckung ein zentrales Kriterium darstellen. Besonders für Angriffe, die während einer längeren Zeitspanne durchgeführt werden sollen, ist es wichtig, dass die Wahrscheinlichkeit entdeckt zu werden möglichst gering bleibt. Insbesondere für das *Greedy Behaviour* stellt dieses Kriterium einen wichtigen Faktor dar, da eine Entdeckung möglicherweise zu einer Bestrafung der egoistischen Station führen kann.
- **Standardkonformität:** Insbesondere bei Angriffen gegen Protokolle der MAC-Schicht kann man zwischen standardkonformen und nicht standardkonformen Angriffen unterscheiden. Ein standardkonformer Angriff weist keine auffälligen Abweichungen zum Verhalten normaler Stationen eines Netzes auf und ist somit schwerer zu entdecken. Nicht standardkonforme Angriffe halten sich hingegen nicht an das vorgeschriebene Verhalten und können somit meist leichter ihr Ziel erreichen. Durch auffällige Abweichungen vom Standard erhöht sich allerdings gleichzeitig die Wahrscheinlichkeit entdeckt zu werden.
- **Einfluss der Betriebsart:** Viele der später diskutierten Angriffe setzen ein Netz auf Basis des Infrastruktur-BSS voraus oder wurden lediglich in diesen Netzen auf ihre Umsetzbarkeit

getestet. Da aber die Verbreitung und Bedeutung von Ad-hoc-Netzen (IBSS oder WBSS) immer weiter zunimmt, ist der Einfluss der Betriebsart auf die Anwendbarkeit und Wirkungsweise eines Angriffs ebenfalls ein wichtiges Bewertungskriterium.

- **Voraussetzung der Authentisierung:** Einige Angriffe setzen die zuvor erfolgreiche Authentisierung eines Angreifers voraus. Angriffe dieser Art können bei Verwendung geeigneter Protokolle zur Authentisierung verhindert werden. Somit sind diese nur bei Netzen mit schwacher bis keiner Authentisierung von Bedeutung.
- **Einfluss von Techniken der PMD-Schicht:** Die Wirkungsweise von Angriffen, die sich gegen die PHY-Schicht des 802.11-Standards richten, sind meist abhängig von den zugrundeliegenden Modulationstechniken der PMD-Schicht wie sie in Abschnitt 2.4 beschrieben wurden. Frequenzspreizverfahren wie DSSS sind beispielsweise gegen schmalbandige Störsignale relativ resistent.
- **Genauigkeit:** Die Genauigkeit eines Angriffs spielt insbesondere dann eine wichtige Rolle wenn nur bestimmte Knoten eines Netzes angegriffen werden sollen. Dies kann beispielsweise bei Angriffen auf Multihop Ad-hoc-Netze der Fall sein, wenn wichtige Verbindungsknoten gestört werden sollen, um somit die Weiterleitung von Paketen zu unterbinden. Auch für die Vermeidung der Entdeckung kann eine hohe Genauigkeit von Vorteil sein und ist somit auch bei *Greedy Behaviour* wichtig.
- **Wirkungsgrad:** Der Wirkungsgrad oder auch die Effektivität eines Angriffs ist ein Maß für dessen erzielte Wirkung. Abhängig von der Motivation und der verwendeten Methode existieren verschiedene Metriken, mit denen der Wirkungsgrad gemessen werden kann [127, 89].
  - *Packet Send Ratio:* Das *Packet Send Ratio* (PSR) definiert das Verhältnis zwischen der Anzahl erfolgreich gesendeter Pakete m zur Anzahl der zum Senden vorgesehenen Pakete n mit $n \geq m$ als:
    $$PSR = \frac{m}{n}$$
    Dies ist eine einfache Art und Weise den Wirkungsgrad eines DoS-Angriffs auf Senderseite zu messen. Diese Metrik ist somit interessant, um den Wirkungsgrad verschiedener DoS-Angriffe zu vergleichen, die versuchen den Kanal zu belegen und das Senden von Paketen verhindern wollen.
  - *Packet Delivery Ratio:* Das *Packet Delivery Ratio* (PDR) beschreibt den Wirkungsgrad eines DoS-Angriffs aus Sicht des Empfängers. Mit m als Anzahl der gesendeten Pakete und q als Anzahl der erfolgreich empfangenen Pakete (d.h. mit korrekter CRC) definiert sich das Verhältnis als:
    $$PDR = \frac{q}{m}$$
    Diese Metrik kann beispielsweise bei der Bewertung verschiedener *Corruption Jamming* Angriffe herangezogen werden.
  - *Jamming-to-Signal Ratio:* Das *Jamming-to-Signal Ratio* beschreibt die herkömmliche Metrik mit der der Wirkungsgrad eines DoS-Angriffs auf physikalischer Ebene gemessen werden kann. Mit Sendeleistung P, Antennengewinn G, Distanz D, Signaldämpfung L, Bandbreite B, Jammer j, Empfänger r, Sender t und den jeweiligen Richtungen rt, tr, jr und rj ergibt sich die folgende Formel für die Berechnung des Jamming-to-Signal Verhältnisses [101]:
    $$\frac{J}{S} = \frac{P_j G_{jr} G_{rj} D_{tr}^2 L_r B_r}{P_t G_{tr} G_{rt} R_{jr}^2 L_j B_j}$$
    Ein hohes Jamming-to-Signal Verhältnis deutet somit auf einen starken DoS-Angriff hin.

- *Connectivity-Index:* Der *Connectivity-Index* beschreibt die Verfügbarkeit von Verbindungen innerhalb eines Ad-hoc-Netzes. Da ein Graph als zusammenhängend gilt, falls zwischen jedem Paar von Knoten mindestens ein Weg existiert, kann der Grad der Konnektivität als durchschnittliche Anzahl an Knoten beschrieben werden, die von jedem anderen Knoten des Graphen aus erreicht werden können. Um eine maximale Wirkung zu erzielen würde ein Jammer somit innerhalb eines Ad-hoc-Netzes eine möglichst geringe Konnektivität anstreben. Eine Verbindung von Knoten a nach Knoten b wird nach Noubir et al. [89] als *non-jammed* definiert falls gilt: $d(a,b) < r \wedge \forall j \in J : d(j,b) > r_j$ mit Funktion d als die euklidische Distanz zwischen zwei Knoten, r als Reichweite einzelner Knoten, J als Menge aller Jammer und r_j als Reichweite eines Jammers. Der Graph eines Ad-hoc-Netzes während der Präsenz eines aktiven Jammers ist daher ein gerichteter Graph $G = (V, E)$, der lediglich aus den zuvor definierten *non-jammed* Verbindungen besteht. Mit $G' = (V, E')$ als transitive Hülle von G definiert sich der Connectivity-Index von G als:
$$\frac{\mid E' \mid}{\mid V \mid^2}$$
Ein zusammenhängender Graph hat somit einen Connectivity-Index von 1. Ein Graph, der in zwei gleich große, zusammenhängende Graphen gesplittet wurde, hat einen Index von 0,5.
- *Fairness-Index:* Die Fairness ist insbesondere bei der Bewertung von *Greedy Behaviour* interessant. Ein *Fairness-Index* misst wie sich ein solcher Angriff auf die Fairness des Kanalzugriffs oder verteilte Bandbreite auswirkt. Dabei existieren verschiedene Verfahren zur Berechnung wie der *Jain Fairness Index* [65] oder der *Min-Max Index* [81]. Auch für Ad-hoc-Netze existieren spezielle Verfahren, um die Fairness zu beschreiben [68]. Für eine genauere Darstellung der Berechnung des Fairness-Indexes sei auf die angegebene Literatur verwiesen.

3.4 Einordnung von Angriffen

Unabhängig von der Motivation eines Angreifers kann kann man bei Angriffen gegen die Verfügbarkeit zwischen naiven und intelligenten Angriffen unterscheiden. Diese können sich jeweils gegen die MAC- oder PHY-Schicht des 802.11-Standards richten. Hierbei sind im Allgemeinen naive Angriffe gegen die PHY-Schicht und intelligenten Angriffe gegen die MAC-Schicht gerichtet, siehe Abbildung 3.1 auf der nächsten Seite.

Intelligente Angriffe zeichnen sich meist durch die Berücksichtigung eines Großteils der oben genannten Kriterien aus. Die Hauptziele liegen in der Regel auf einer Steigerung der Energieeffizienz, Verringerung der Entdeckungswahrscheinlichkeit und einer hohen Genauigkeit. Als Unterkategorien der intelligenten Angriffe lassen sich Angriffe gegen allgemeine Mechanismen der MAC-Schicht, gegen Protokolle der 802.11i-Erweiterung und gegen Treiber und Firmware identifizieren.

Als *Radio Frequency Jamming* (RF Jamming) werden DoS-Angriffe bezeichnet, die auf der physikalischen Störung des Funksignals basieren und sich somit gegen die PHY-Schicht richten [92, 116, 6]. Die meisten dieser Angriffe lassen sich wie in Abbildung 3.1 verdeutlicht in die Kategorie der naiven Angriffe einordnen. Allerdings existieren auch hier Methoden wie *Reactive Jamming* oder *Corruption Jamming*, die sich durch die Berücksichtigung der MAC-Protokolle auszeichnen und somit ebenfalls zu den intelligenten Angriffen zählen.

In Abbildung 3.1 ist eine Übersicht einiger der in den nächsten Abschnitten vorgestellten Angriffe und deren Einordnung in die beschriebenen Kategorien zu sehen. Hier wird deutlich, dass sich ein Großteil der folgenden Angriffe gegen die Mechanismen der MAC-Schicht richtet.

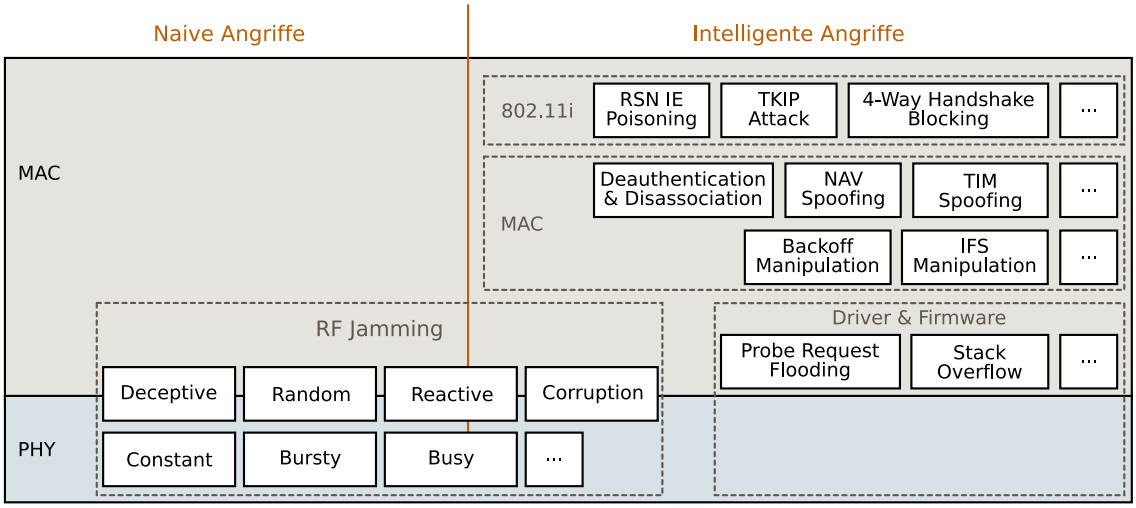


Abbildung 3.1: Einordnung verschiedener Angriffe gegen die Verfügbarkeit

3.5 RF Jamming

Wie bereits erwähnt bezeichnet *RF Jamming* die physikalische Störung eines Funksignals mit dem Ziel eine Verbindung zu unterbrechen. Xu et al. definieren demnach einen *Jammer* als eine „*Einheit, die bewusst versucht den physikalischen Versand und Empfang einer kabellosen Kommunikation zu behindern*“ [127]. Da 802.11-Netze in Frequenzbereichen der lizenzfreien ISM-Bänder wie 2,4 GHz (802.11b/g) arbeiten, sind Interferenzen durch Sender im gleichen Frequenzbereich nicht unwahrscheinlich. Diese unbewussten Störeinflüsse können beispielsweise durch Bluetooth-Geräte, kabellose Telefone, Geräte zur drahtlosen Audio- und Videoübertragung oder Mikrowellenherde verursacht werden. Nach der oben genannten Definition können solche Geräte bei bewusster Verwendung zur Störung einer Kommunikation ebenfalls als Jammer bezeichnet werden. Insgesamt lassen sich nach Xu et al. [127] und Acharya et al. [5] sieben verschiedene Modelle des *RF Jammings* finden, die in folgenden Abschnitten im Detail erläutert werden.

3.5.1 Constant Jamming

Das *Constant Jamming* stellt die einfachste Form eines DoS-Angriffs dar. Bei diesem Ansatz wird kontinuierlich ein Funksignal auf der Frequenz des zu störenden Kanals ausgesendet. Dieser Vorgang lässt jede andere Station in Reichweite das Medium als belegt erkennen. Genauer betrachtet würde somit die PLCP-Schicht jeder Station durch die Dienstprimitive `PHY-CCA.indicate(BUSY)` die Belegung des Kanals der MAC-Schicht signalisieren.

Dies kann beispielsweise wie bei Ståhlberg [113] durch einen Signalgenerator geschehen oder wie von Wullems et al. [125] und Chen et al. [34] beschrieben, durch das Eingreifen in die Firmware bestimmter 802.11b-Schnittstellen (NICs). Hierbei wurden Prism NICs durch die optiona-

le PLME-Dienstprimitive `PLME-DSSSTESTMODE.request` in einen Test-Modus versetzt. In diesem Test-Modus wird ein bestimmtes 16 Bit langes Muster kontinuierlich auf einem zuvor eingestellten Kanal übertragen. Die Signalstärke betrug dabei bis zu 200 mW[3].

Auch Gummadi et al. [51] verwenden in ihrer Arbeit diesen Ansatz und stellen fest, dass schon eine Sendeleistung von 16 mW genügt, um die Kommunikation vollständig zu unterbrechen. Um ein Viertel der übertragenen Daten zu zerstören, genügte bei ihren Tests bereits eine Sendeleistung von 10 µW. Die Tests wurden in einem Raum mit einer Größe von 30 mal 30 Metern durchgeführt.

Acharya et al. [5] benötigen in ihren Simulationen lediglich ein Störsignal mit der Leistung von 1 mW, um die Kommunikation komplett zu unterbinden. Um mindestens noch die Hälfte der Kommunikation verhindern zu können, bestimmen sie eine minimale Leistung von nur 1 µW. Diese Art des Jammings bezeichnen sie dabei als *Continuous Low Power Jamming*. Die Ergebnisse ihrer Simulationen basieren auf einem Umgebungsmodell mit Stationen auf verschiedenen Etagen eines Hauses und einem AP im Erdgeschoss. Die Abstände zwischen den einzelnen Stationen sind nicht größer als 50 Meter. Der Jammer ist 10 Meter über dem AP platziert.

Da die Abstände zwischen Jammer und Stationen bei den Tests von Gummadi et al. und den Simulationen von Acharya et al. einer ähnlichen Größenordnung entsprechen, ist der enorme Unterschied der benötigten Signalstärke von 1 mW und 16 mW umso erstaunlicher. Dies zeigt, dass Simulationsergebnisse insbesondere bei der Einschätzung von physikalischen Auswirkungen nicht ohne weiteres auf die Praxis übertragbar sind.

Auch wenn die Sendeleistung von 16 mW relativ gering erscheinen mag, bleibt ein relativ hoher Energieverbrauch durch das kontinuierliche Senden eines Störsignals grundsätzlich ein Nachteil des *Constant Jammings*. Neben dem Energieverbrauch erhöht sich ebenfalls die Wahrscheinlichkeit der Entdeckung durch das kontinuierliche Senden [127]. Für lang andauernde Angriffe ist ein solches Vorgehen daher ungeeignet. Ein weiterer Nachteil neben diesen Aspekten stellt die Abhängigkeit von der verwendeten PMD-Schicht dar. So konnten Chen et al. [34] bei der Verwendung von OFDM keinerlei Einfluss auf die Übertragung während ihres Feldversuchs feststellen. Bei der Verwendung von DSSS wurde hingegen die komplette Kommunikation während des Angriffs unterbunden. In der Arbeit von Pelechrinis et al. [92] werden einige Möglichkeiten zum Erkennen und zur Vermeidung von Jamming-Angriffen zusammengefasst. Als Anti-Jamming-Techniken werden die Erhöhung der Sendeleistung, die Verwendung gerichteter Antennen oder die Verwendung bestimmter Frequenzspreizverfahren bei schmalbandigen Jamming Signalen diskutiert, vergleiche Abschnitt 2.4.1. Gerichtete Antennen könnten allerdings auch durch einen Angreifer für die Durchführung gezielter Angriffe gegen einzelne Stationen genutzt werden.

Ein Vorteil des *Constant Jammings* ist zum einen die Unabhängigkeit von der Betriebsart eines Netzes. So sind Jamming-Angriffe sowohl in Infrastruktur-BSS als auch in IBSS gleichermaßen wirksam. Zum anderen ist der hohe Wirkungsgrad ein Vorteil, da zumindest bei hoher Sendeleistung die Kommunikation vollständig unterbrochen werden kann. Noubir et al. [89] diskutieren einige Möglichkeiten, um den Wirkungsgrad eines solchen Angriffs und somit das Jamming-to-Signal-Verhältnis zu reduzieren.

[3]Die Signalstärke wird auch oft in dBm angegeben. Das Verhältnis zwischen dBm und mW ist dabei wie folgt: $P = 10^{(x/10)} mW$, mit P als Leistung in mW und x in dBm

3.5.2 Deceptive Jamming

Auch bei Angriffen dieser Art wird kontinuierlich gesendet. Allerdings werden keine beliebigen Funksignale, sondern reguläre Pakete ohne Rücksicht auf vorgeschriebene Zeitabstände übertragen. Dies kann durch die Manipulation von Protokollparametern der DCF erreicht werden, siehe Abschnitt 3.6.4 auf Seite 53. Als Folge geht ein ahnungsloser Teilnehmer des Netzes davon aus, dass eine normale Datenübertragung stattfindet. Auch bei diesem Ansatz ist der Energieverbrauch äußerst hoch und da das Nichtbeachten korrekter Zeitabstände nicht standardkonform ist, sind Angriffe dieser Art ebenfalls leicht zu entdecken.

3.5.3 Bursty und Busy Jamming

Als *Bursty High Power Jamming* bezeichnen Acharya et al. [6] das periodische Aussenden eines Störsignals mit hoher Signalstärke. Eine spezielle Form dieser Methode ist das *Busy Jamming*, bei dem versucht wird das Verhalten der Distributed Coordination Function auszunutzen. Hierzu wird der Abstand zwischen zwei Störsignalen kleiner gewählt als die Dauer eines DIFS. Dies bewirkt, dass eine Station in Reichweite das Medium als belegt erkennt, da die Dienstprimitive `PHY-CCA.indicate` nie für die benötigte Dauer eines DIFS einen freien Kanal signalisiert. Diese Methoden benötigen deutlich weniger Energie als das kontinuierliche Aussenden eines Störsignals. Bei Verwendung von DSSS beträgt die Dauer eines DIFS 50 µs. Geht man davon aus, dass ein Angreifer nun alle 50 µs ein Störsignal für 10 µs mit einer Signalstärke von 16 mW sendet, dann benötigt er nur noch eine Leistung von 3,2 mW pro Sekunde.

Eine einfache Möglichkeit für die Umsetzung von *Bursty High Power Jamming* stellt die Verwendung eines Mikrowellenherdes dar. Diese arbeiten ebenfalls auf dem 2,4-GHz-ISM-Band und durchlaufen einen stetigen Kreislauf bestehend aus einer An-Phase und einer Aus-Phase. Taher et al. [115] haben in ihrer Arbeit die Auswirkungen von Mikrowellenherden auf 802.11-Netze untersucht und gleichzeitig eine praktikable Lösung für die Vermeidung von Störeinflüssen präsentiert. Der Lösungsansatz garantiert eine zuverlässige Datenübertragung auf Kosten eines verringerten Datendurchsatzes um 50 Prozent.

3.5.4 Random Jamming

Das *Random Jamming* versucht ähnlich der Vorgehensweise des *Bursty Jammings* den Energieverbrauch durch alternierendes Ein- und Ausschalten der Sendeeinheit zu verringern. Hierbei wird für eine bestimmte zufällige Zeitspanne t_j ein Störsignal oder ein Paket gesendet und anschließend eine ebenfalls zufällige Zeitspanne t_s gewartet. Durch gezielte Veränderung dieser beiden Werte kann ein Kompromiss zwischen Energieeffizienz und Wirkungsgrad des Angriffs erzielt werden.

3.5.5 Reactive Jamming

Bei allen bisher genannten Ansätzen ist das Vorgehen unabhängig von dem tatsächlichen Zustand des Mediums. Beim *Reactive Jamming* hingegen wird das Medium fortlaufend beobachtet und nur bei der Feststellung einer Datenübertragung ein Störsignal gesendet. Zhou et al. [129] kombinieren *Reactive Jamming* mit *Random Jamming* und bezeichnen diese Form des Jammings als *Random Packet Destruction DoS*. Hierbei wird nach der Feststellung einer Übertragung ein Störsignal mit einer Wahrscheinlichkeit p_a für die Dauer t_a gesendet. Ihre Simulationen zeigen, dass ein Angreifer durch Verwendung dieser Technik mit $p_a \geq 0,425$ und $t_a = 10\ \mu s$ eine TCP-basierte Verbindung komplett unterbrechen kann. Bayraktaroglu et al. [11] untersuchen in ihrer Arbeit die Auswirkungen und Effizienz des *Reactive Jammings* anhand von Simulationen und realen Feldversuchen. Dabei kommen sie zu dem Ergebnis, dass *Reactive Jamming* um das Vierfache effizienter ist als *Constant Jamming*. Dadurch, dass ein Angreifer nur zu bestimmten Zeitpunkten ein Störsignal sendet, ist auch die Entdeckungswahrscheinlichkeit geringer.

3.5.6 Corruption Jamming

Noch einen Schritt weiter als *Reactive Jamming* gehen Angriffe der Kategorie *Corruption Jamming*. Hierbei wird versucht durch das Wissen über den Ablauf der Protokolle auf der MAC-Schicht, einzelne Pakete gezielt zu beschädigen. Alle Angriffe dieser Art führen zu einem Verlust des beschädigten Pakets, da die PHY-Schicht von 802.11 keine Fehlerkorrektur durchführt und somit die Veränderung eines einzelnen Bits zu einer fehlerhaften Checksumme (CRC) im PLCP-Header führt. Ein PLCP-Paket mit einer fehlerhaften CRC wird auf PHY-Ebene sofort verworfen und führt folglich zu einer erneuten Übertragung des Pakets durch den Sender. Jedes nicht bestätigte Paket wird bis zu einem vordefinierten Limit[4] erneut übertragen. Eine Übersicht der relevanten Pakete im Zusammenhang des Kanalzugriffs von 802.11 wurde bereits in Kapitel 2.5.1 gegeben und ist in Abbildung 2.11 auf Seite 18 dargestellt. Acharya et al. [6] unterscheiden zwischen vier möglichen Ansätzen des *Corruption Jammings*:

- *CTS Corruption:* Dieser Angriff setzt die Verwendung der DCF mit dem virtuellen Carrier Sense-Mechanismus voraus, vergleiche Abschnitt 2.5.1. Der Angreifer wartet zunächst auf den Empfang eines RTS-Pakets und sendet nach Abwarten eines SIFS ein Störsignal, welches das nachfolgende CTS-Paket beschädigt. Durch diesen Angriff würde also keine Datenübertragung zustande kommen.

[4] Das Retransmission-Limit ist ein konfigurierbarer Parameter. Der 802.11-Standard schlägt hierfür einen Wert zwischen 4 und 7 vor.

- *DATA Corruption:* Auch dieser Ansatz setzt die DCF mit virtuellem CS voraus. Hierbei wird auf den Empfang eines CTS-Pakets gewartet und nach Abwarten eines DIFS das nachfolgende Datenpaket durch ein Störsignal beschädigt.
- *ACK Corruption:* Bei diesem Angriff wird auf den Empfang eines Datenpakets gewartet und nach Abwarten eines SIFS ein Störsignal gesendet, welches in diesem Falle das nachfolgende ACK-Paket beschädigt. Da der Sender somit keine Empfangsbestätigung erhält, wiederholt dieser den Sendeversuch und bricht nach mehrfacher, erfolgloser Wiederholung ab.
- *DIFS Waiting:* Ist das Medium für die Zeitdauer eines DIFS unbenutzt, wird bei diesem Angriff unmittelbar danach ein Störsignal gesendet, ohne zu wissen ob und welches Paket beschädigt wird. Bei Erfolg wird entweder ein Datenpaket oder, bei Verwendung des virtuellen CS-Mechanismus, ein RTS-Paket zerstört. Dieses Vorgehen benötigt in den Simulationen von Acharya et al. [5] die gleiche Energie wie das *Busy Jamming*, beispielsweise für einen 10 minütigen Angriff theoretisch nur 12 µW.

Die Corruption-Angriffe sind die effizientesten der Kategorie des *RF Jammings* und sind aufgrund ihres nahen Bezugs zur MAC-Schicht auch als intelligente Angriffe einzustufen. Die Simulationen von Acharya et al. [5] ergeben eine benötigte Energie von nur 1 µW, um einen Angriff für 10 Minuten durchzuführen. Durch das gezielte Zerstören einzelner Pakete ist einerseits der Wirkungsgrad des DoS-Angriffs sehr hoch und andererseits verringert sich gleichzeitig die Wahrscheinlichkeit entdeckt zu werden. Um einen gezielten Angriff gegen einzelne Stationen durchzuführen, kann neben der bereits erwähnten Verwendung von gerichteten Antennen ein weiterer Ansatz gewählt werden, wie er von Schoch et al. [102] beschrieben wird. Ein Angreifer müsste zunächst den Header des MAC-Pakets und die darin enthaltenen Adressangaben auslesen. Anschließend könnte er entscheiden ob er das Paket zerstören will oder nicht. Da sich durch die Zerstörung der Pakete auch das *Contention Window* der betroffenen Stationen erhöht (vgl. Abschnitt 2.5.1), kann dieser Angriff auch für eine egoistische Station genutzt werden, um einen höheren Datendurchsatz zu erlangen.

3.6 Angriffe gegen die MAC-Schicht

Der folgende Abschnitt gibt einen Überblick über bekannte DoS-Angriffe, die sich gegen die allgemeine Funktionalität und Protokolle der MAC-Schicht richten. Ein grundsätzlicher Vorteil dieser Angriffe im Vergleich zu den bisher beschriebenen Varianten des *RF Jammings* ist ihre Unabhängigkeit von den verwendeten Modulationsverfahren der PMD-Teilschicht. In der oft zitierten Arbeit von Bellardo et al. [13] werden viele dieser Angriffe vorgestellt und teilweise untersucht. Der Großteil von ihnen basiert auf der Vortäuschung einer anderen Identität (*Masquerading*) und dem Fälschen von Control- oder Management-Nachrichten. Das Vortäuschen einer anderen Identität bietet aus Sicht eines Angreifers den großen Vorteil, dass ein Angriff sehr gezielt gegen eine bestimmte Station gerichtet sein kann. Da wie bereits erwähnt für Control- und Management-Nachrichten keinerlei Schutz der Vertraulichkeit, Integrität sowie Authentizität besteht, sind die nachfolgenden Angriffe leicht durchzuführen.

Angriffe, die auf der Manipulation von Management-Nachrichten beruhen, sollen zukünftig durch die Erweiterung 802.11w [64] verhindert oder zumindest erschwert werden. Da diese Erweiterung zum Schutz von Management-Nachrichten noch nicht verfügbar ist, existieren alternative Ansätze, die insbesondere das Fälschen einer Identität erkennen oder verhindern sollen. Sheng et al. [105] stellen für die Erkennung ein Verfahren basierend auf dem Vergleich der empfangenen Signalstärken vor. Zur Vermeidung derartiger Angriffe diskutieren Khan et al. [73] die Erweiterung durch eine Authentisierung von Control- und Management-Nachrichten basierend auf einer pseudozufälligen Zahl. Da derartige Verfahren in der Praxis allerdings kaum eingesetzt werden, stellen die hier diskutierten Angriffe bis zur Verabschiedung und Integration der 802.11w-Erweiterung nach wie vor eine große Bedrohung dar.

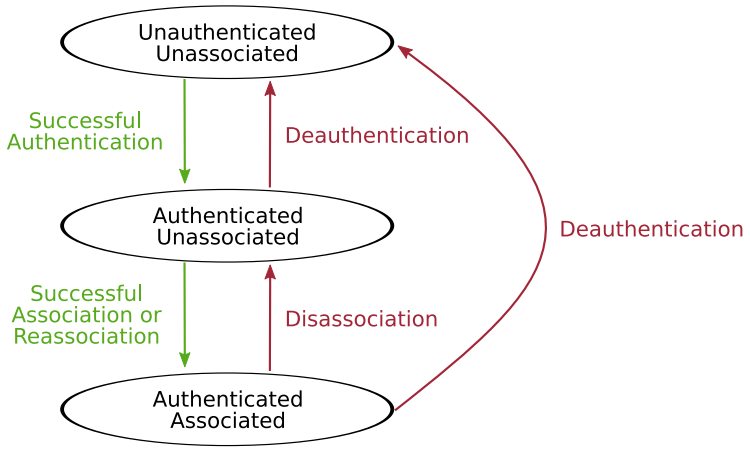


Abbildung 3.2: Zustandsdiagramm einer 802.11-Station [60]

3.6.1 Deauthentication und Disassociation

Bevor Daten zwischen einer Station und einem Access Point (AP) innerhalb eines Infrastruktur-BSS ausgetauscht werden können, muss wie in Abschnitt 2.5.3 auf Seite 23 beschrieben, zunächst eine *Authentication* und *Association* statt finden. Der Standard erlaubt des Weiteren das Versenden von Management-Nachrichten des Subtyps *Deauthentication* oder *Disassociation*. Dabei dient die Deauthentication-Nachricht zum Beenden einer bestehenden *Open System* oder *Shared*

Key Authentication und die Disassociation-Nachricht zum Beenden einer bestehende *Association*. Der Vorgang der Deauthentication veranlasst ebenfalls das Beenden einer bestehenden Association und bei Verwendung einer RSNA das Löschen des PTK sowie das Schließen des kontrollierten 802.1X-Ports. Vereinfacht ausgedrückt wird durch die Deauthentication eine Verbindung vollständig beendet. Für deren Wiederaufbau muss somit jeder Schritt genau wie bei einem Neuaufbau wiederholt werden. Der Vorgang der Disassociation lässt eine Station hingegen weiterhin im Zustand *Authenticated*, siehe Abbildung 3.2 auf der vorherigen Seite. Für einen Wiederaufbau der Verbindung muss sich eine Station daher lediglich erneut anmelden (*Reassociation*). Die Deauthentication und Disassociation kann sowohl von einer Station als auch von einem AP initiiert werden und sollte laut Standard von dem jeweiligen Empfänger nicht abgelehnt werden. Innerhalb eines Ad-hoc-RSN (IBSS RSN) ist die Verwendung der *Open System Authentication* optional. Dennoch soll nach der Spezifikation des Standards jede Station in der Lage sein eine Deauthentication-Nachricht zu erkennen und daraufhin den kontrollierten 802.1X-Port schließen sowie den PTK löschen.

Wie von Bellardo et al. [13] beschrieben, hat ein Angreifer nun die Möglichkeit diese Management-Nachrichten zu fälschen und somit eine Verbindung gezielt zu beenden. Da die Management-Nachrichten in keiner Form geschützt sind, hat eine Station grundsätzlich keine Möglichkeit die Echtheit der Nachrichten zu überprüfen. Die effektivste Angriffsmethode ist das Versenden der Deauthentication-Nachricht, da diese wie bereits erwähnt eine Verbindung komplett beendet. Diese kann entweder an den AP (SA = MAC-Adresse des Opfers), oder an das Opfer selbst adressiert sein (SA = MAC-Adresse des AP). Wird die Nachricht an den AP gesendet, werden alle folgenden Datenpakete des Opfers durch den AP verworfen (siehe Abbildung 3.3). Erst nach erneutem Verbindungsaufbau kann die Kommunikation wieder fortfahren. Um eine Kommunikation dauerhaft zu unterbrechen und somit einen DoS-Effekt zu erzielen, könnte ein Angreifer kontinuierlich Deauthentication-Nachrichten versenden. Dies würde allerdings einen hohen Energieverbrauch verursachen. Eine wesentlich effizientere Vorgehensweise basiert auf dem Beobachten der übertragenen Nachrichten und dem Versenden der Deauthentication-Nachricht an nur wenigen, aber gezielten Zeitpunkten. Optimale Zeitpunkte sind unmittelbar nach Beobachtung einer *Association Response* oder innerhalb eines RSN nach Beobachtung der letzten Nachricht des 4-Way-Handshakes. Diese wird bei der ersten Durchführung des 4-Way-Handshakes im Klartext übertragen und ist daher für einen Angreifer einsehbar.

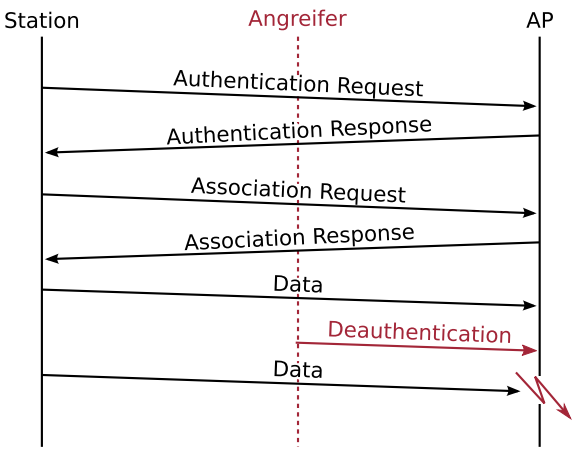


Abbildung 3.3: Ablauf eines Deauthentication-Angriff nach [13]

Bellardo et al. [13] wählten in ihren Tests die beiden Zeitpunkte nach Beobachtung einer *Association Response* oder eines Datenpakets für das Aussenden einer Deauthentication-Nachricht. Die Rate der Nachrichten beschränkten sie dabei auf zehn Pakete pro Sekunde. Sie untersuchten dabei zum einen die Auswirkung eines gezielten Deauthentication-Angriffs während einer TCP-Übertragung auf nur eine Station und zum anderen den Angriff auf vier Stationen zur selben Zeit. Obwohl der erste Angriff lediglich acht Sekunden andauerte, konnte die Übertragung erst nach mehr als einer Minute fortgesetzt werden. Diesen Effekt erklärten Bellardo et al. zum einen durch den Versuch der Station sich mit anderen APs zu verbinden und zum anderen durch die Auswirkungen des Angriffs auf die Überlaststeuerung (*Congestion Control*) von TCP. Während des Angriffs auf mehrere Stationen mit unterschiedlichen Betriebssystemen erhielten sie abweichende Ergebnisse. Dies erklärten sie durch die kurze Verzögerung zwischen Empfang einer *Association Response* und Senden der Deauthentication-Nachricht, die beispielsweise bei Stationen mit einem Windows XP Betriebssystem für das Versenden von wenigen UDP Datenpaketen ausreichte.

Eine interessante Variante des Deauthentication-Angriffs wurde von Ahmad et al. vorgestellt [7]. Sie stellten fest, dass bestimmte Access Points auf den Empfang speziell angepasster Pakete mit dem Versenden von Deauthentication-Nachrichten reagierten. Sie bezeichneten dieses Verhalten der APs als *Autoimmune Disorder* (Autoimmunstörung), bei der autorisierte Stationen fälschlicher Weise durch die APs attackiert wurden. Dieses Verhalten stellten sie bei Open Source als auch bei kommerziellen APs fest. Insgesamt konnten sie fünf verschiedene Ansätze identifizieren, durch die dieses Verhalten ausgelöst wurde:

- **Broadcast** MAC **Adresse als Source Address**: Wird durch einen Angreifer die Broadcast Adresse (FF:FF:FF:FF:FF:FF) als *Source Address* innerhalb eines Datenpakets angegeben, wird ein AP innerhalb seiner Association-Tabelle nach dieser Adresse suchen, aber in diesem Falle keinen Eintrag finden. Als Reaktion kann er nun eine Deauthentication-Nachricht an diese Adresse versenden. Da es sich um die Broadcast-Adresse handelt werden alle Stationen angesprochen, welche daraufhin ihre Verbindungen trennen. Dieser Angriff kann ebenfalls mit einer Multicast-Adresse (01:XX:XX:XX:XX:XX) durchgeführt werden.
- **Datenpaket mit Address 4 Feld**: Wird ein Datenpaket mir der Source Address der Opferstation und der Verwendung aller vier Adressfelder des MAC-Headers an den AP gesendet, kann es vorkommen, dass dieser mit dem Feld *Address 4* nicht umgehen kann und als Folge eine Deauthentication-Nachricht an das Opfer sendet.
- **Authentication Request mit gefälschten Angaben:** Durch gefälschte Angaben in einem *Authentication Request* kann ein Angreifer ebenfalls eine Deauthentisierung durch den AP auslösen, wenn die Angaben durch den AP nicht unterstützt werden. Beispiele hierfür sind eine ungültige *Authentication Algorithm Number* oder eine ungültige *Authentication Transaction Sequence Number*.
- **Association Request mit gefälschten Angaben:** Innerhalb eines Association Requests kann eine Station durch das Feld *Capability Information* ihre unterstützten Fähigkeiten angeben. Durch das *Supported Rates Information Element* (IE) kann sie des Weiteren die unterstützen Übertragungsraten angeben. Durch gezieltes Fälschen dieser Angaben, die durch den AP nicht unterstützt werden und das Senden des Requests an den AP, kann wiederum ein Deauthentisieren der Station hervorgerufen werden.
- **Reassociation Request mit gefälschter Current AP Adresse:** Eine weitere Möglichkeit das Versenden einer Deauthentication-Nachricht auszulösen, besteht im Fälschen des Feldes *Current AP address*. Dieses Feld gibt innerhalb eines Reassociation Requests die

Adresse des AP an, mit dem die Station zuletzt verbunden war. Wird hier eine andere als die des APs angegeben, kann es zur Trennung der Verbindung kommen.

Ahmad et al. [7] testeten die vorgestellten Angriffe mit sechs verschiedenen APs. Dabei waren besonders die Angriffe durch das Versenden der Authentication und Reassociation Requests erfolgreich. Bei allen getesteten APs trat der gewünschte Effekt ein. Weniger erfolgreich waren hingegen die Angriffe mit angegebener Broadcast- oder Multicast-Adresse. Diese führten nur bei der Hälfte der getesteten APs zum Ziel.

Alle Angriffe dieser Art, insbesondere die zuletzt beschriebenen Angriffe, die eine *Autoimmune Disorder* Reaktion des AP ausnutzen, sind äußerst effektiv und stellen innerhalb eines Infrastruktur-BSS eine hohe Bedrohung dar. In einem IBSS sind diese Angriffe allerdings lediglich bei Verwendung einer RSNA eine mögliche Gefahr. Beschränkt man das Versenden der gefälschten Pakete auf nur Wenige pro Sekunde, so verringert sich einerseits der Energieverbrauch, andererseits kann aber auch der Wirkungsgrad des Angriffs abnehmen. Sendet man hingegen sehr viele Pakete in kürzester Zeit, um einen möglichst großen DoS-Effekt zu erzielen, so erhöht sich entsprechend auch die Wahrscheinlichkeit der Entdeckung, die bei nur wenigen Paketen aufgrund der Standardkonformität der einzelnen Pakete relativ gering bleibt. Da aber die Ergebnisse von Bellardo et al. [13] gezeigt haben, dass schon durch wenige Pakete die Verbindungen teilweise bis zu einer Minute unterbrochen werden konnte, ist die Effizienz und der Wirkungsgrad dieser Angriffe grundsätzlich sehr hoch einzustufen.

3.6.2 Fälschen von Management-Informationen

Innerhalb von Beacons und Probe Responses sind verschiedene Management-Informationen enthalten, die ein Angreifer manipulieren und dadurch einen DoS-Angriff durchführen kann.

Fälschen des DS Parameter Sets

In einem Infrastruktur-BSS kann ein AP innerhalb eines Beacons oder einer Probe Response ein *DS Parameter Set* Information Element (IE) angeben. Dieses enthält Informationen, die es Stationen bei Verwendung von DSSS ermöglicht, den aktuellen Kanal zu identifizieren. Ein Angreifer kann durch Angabe eines ungültigen Kanals innerhalb dieses IEs dafür sorgen, dass jede Station des BSS versucht auf den angegebenen Kanal zu wechseln [77]. Ein ungültiger Kanal wäre beispielsweise 0 oder 255. Durch diese Angabe wird die Verbindung vorübergehend unterbrochen und erst nach einem Timeout erneut aufgebaut.

Fälschen der Channel Switch Announcement

Ein ähnlicher Effekt könnte durch das Fälschen eines Beacons mit Angabe eines *Channel Switch Announcement* IEs erreicht werden, wie es bei der Verwendung von DFS vorgesehen ist, vergleiche Abschnitt 2.5.3. Ein Angreifer hat dabei ebenfalls die Möglichkeit wie oben beschrieben einen ungültigen Kanal anzugeben. Um aber die Effizienz dieses DoS-Angriffs zu erhöhen, kann ein Angreifer zusätzlich den *Channel Switch Mode* auf den Wert 1 und den *Channel Switch Count* auf den maximalen Wert von 255 setzen. Somit würden alle Stationen, die ein derart modifiziertes Beacon

erhalten, für die Dauer von 255 TBTTs keine weiteren Pakete versenden und anschließend auf den ungültigen Kanal wechseln. Erst nach dem zusätzlichen Timeout würden sie erneut versuchen auf einem anderen Kanal eine neue Verbindung aufzubauen. Manche Entwickler von Treibern haben dieses Problem aber bereits erkannt und Gegenmaßnahmen eingeführt. So lässt der Madwifi[5] Treiber beispielsweise nur einen *Channel Switch Count* kleiner oder gleich 1 zu. Inwieweit andere Treiber und Firmware auf die Angabe einer *Channel Switch Announcement* reagieren bleibt zu untersuchen, da soweit bekannt, hierzu noch keine Quellen in der Literatur existieren.

Fälschen des Quiet-Elements

Ein weiteres IE, das ein Angreifer für die Durchführung eines DoS-Angriffs ausnutzen könnte, ist das *Quiet Element*, siehe Abschnitt 2.5.3. Soweit bekannt, existiert bisher noch keine Literatur, die dieses Vorgehen beschreibt. Wird dieses IE von Stationen berücksichtigt, so hat ein Angreifer theoretisch die Möglichkeit durch eine maximale Duration-Angabe von 65535 TUs die Kommunikation anderer Stationen für 67 Sekunden zu unterbinden. Durch die Angabe, dass sich dieses Intervall periodisch wiederholen soll, könnte ein Angreifer somit durch geringsten Aufwand einen andauernden DoS-Effekt erzielen. Genau wie das Channel Switch Announcement IE, ist auch das Quiet-Element Bestandteil des DFS-Mechanismus. Da der DFS-Mechanismus nur bei der Verwendung von 802.11a in Europa durch den Standard vorgeschrieben ist und für andere Frequenzbänder als das 5-GHz-Band eine optionale Erweiterung darstellt, ist die Wirkung dieses Angriffs abhängig von der Implementierung der Treiber und Firmware aktueller NICs. Hierzu gibt es in der Literatur bisher keine Informationen. In wie weit Hersteller den DFS-Mechanismus unterstützen bleibt somit zu untersuchen.

Das Fälschen von Management-Informationen stellt eine sehr effiziente Möglichkeit für starke DoS-Angriffe dar. Der 802.11-Standard erlaubt hier durch einfache Manipulation der Zeit- oder Kanalangabe eine theoretische Unterbrechung der Kommunikation bis zu einer Minute. Neben der hohen Effizienz dieser Angriffe ist die Unabhängigkeit von der Betriebsart des Netzes ein weiterer Vorteil. Da nur ein einzelnes gefälschtes Paket für diese Angriffe ausreichen kann, bleibt auch die Entdeckungswahrscheinlichkeit sehr gering. Hierbei muss allerdings beachtet werden, dass die Angabe eines ungültigen Kanals nicht standardkonform ist und somit die Wahrscheinlichkeit einer Entdeckung steigt. Fälscht ein Angreifer nur die Angaben in bestimmten Probe Responses so kann auch ein gezielter Angriff gegen einzelne Stationen durchgeführt werden.

3.6.3 Angriffe gegen Energiesparmechanismen

Die Mechanismen zur Energieeinsparung, die in Abschnitt 2.5.3 erläutert wurden, erlauben ebenfalls verschiedene Angriffe gegen die Verfügbarkeit. Bellardo et al. [13] diskutieren drei mögliche DoS-Ansätze, die innerhalb eines Infrastruktur-BSS durchführbar sind:

1. **Fälschen der TIM:** Ein Angreifer hat die Möglichkeit die TIM eines Beacons beziehungsweise einer Probe Response zu fälschen. Er könnte folglich eine leere TIM an eine Station senden, damit diese in den Glauben versetzt wird, dass der Access Point keine neuen Pakete bereithält. Somit würde eine Station nach Erhalt der gefälschten TIM sofort wieder in den Schlafmodus zurückkehren ohne eine PS-Poll-Nachricht an den AP zu senden. Eine

[5] http://madwifi.org/ticket/963

Voraussetzung, damit dieser Angriff durchgeführt werden kann, ist das Verhindern des Empfangs einer korrekten TIM bei der Station. Dies könnte ein Angreifer beispielsweise durch Nichtbeachtung der vorgeschriebenen Zeitabstände erreichen.

2. **Fälschen der PS-Poll-Nachricht:** Eine einfachere Alternative für den Angreifer ist das Fälschen der PS-Poll-Nachricht einer Station an den AP, während die Station sich im Schlafmodus befindet. Hierbei ist ein Angreifer an keinerlei Zeitvorgaben gebunden. Nach Empfang der gefälschten PS-Poll-Nachricht würde der AP alle gespeicherten Pakete der Station versenden und den Speicher wieder freigeben. Da die betroffene Station sich im Schlafmodus befindet, kann diese den Empfang der Pakete nicht bestätigen. Ein Angreifer muss daher ebenfalls die Bestätigung fälschen (ACK), damit der AP die Pakete löscht und den Speicher wieder frei gibt.
3. **Fälschen der Zeitangaben:** Neben dem Fälschen der TIM oder der PS-Poll-Nachricht hat ein Angreifer eine weitere Möglichkeit den Ablauf des Energiesparmechanismus zu beeinträchtigen. Die zeitliche Synchronisierung der beteiligten Station stellt eine Grundvoraussetzung für den fehlerfreien Ablauf des Energiesparmechanismus dar. Ein Angreifer kann daher durch gefälschte Zeitangaben innerhalb eines Beacons oder einer Probe Response dafür sorgen, dass Stationen zur falschen Zeit aufwachen. Diese können somit nicht mehr die echten Beacons und die darin enthaltene TIM des APs empfangen. Die falsche Zeitangabe kann dabei entweder der Zeitstempel oder das Beacon-Intervall innerhalb des Beacons oder der Probe Response sein. Durch einen gefälschten Zeitstempel werden die Uhren der Stationen falsch angepasst. Ein falsches Beacon-Interval beeinflusst die Dauer, für die sich eine Station im Schlafmodus befindet. Beide Angaben können dazu führen, dass Stationen und AP nicht mehr synchron arbeiten.

Auch in einem IBSS erlaubt der Standard ähnliche Angriffe gegen die Energiesparmechanismen, die in der Literatur bisher noch nicht diskutiert wurden und daher an dieser Stelle vorgestellt werden:

1. **Fälschen der Zeitangaben:** In IBSS-Netzen ist es ebenfalls möglich durch gefälschte Zeitstempel die Uhren aller Stationen in Reichweite zu beeinflussen. Dies funktioniert allerdings nur, wenn der gefälschte Zeitstempel in der Zukunft liegt. Die Wirkung ist nicht vergleichbar mit der Wirkung innerhalb eines Infrastruktur-BSS, da zumindest alle Stationen, die das Beacon empfangen, die Zeit übernehmen und somit wieder synchronisiert sind. Die Auswirkung eines solchen Angriffs hängt sicherlich mit dem Grad der Mobilität der einzelnen Stationen und der Größe des Ad-hoc-Netzes ab. Handelt es sich beispielsweise um ein größeres Netz, in dem nicht alle Stationen in Reichweite des Angreifers sind, würde sich die gefälschte Zeitangabe erst nach einigen zusätzlichen Beacons weiterer Stationen ausbreiten.
2. **Fälschen der ATIM-Nachricht:** Die Funktionsweise des Energiesparmechanismus innerhalb eines IBSS bietet eine weitere Angriffsmöglichkeit, die besonders in mobilen Adhoc-Netzen (MANETs) eine Gefahr darstellen könnte. Der Standard schreibt vor, dass eine Station, die eine an sie adressierte ATIM-Nachricht empfängt, für die Dauer des nächsten ATIM-Fensters wach bleiben muss (vgl. Abschnitt 2.5.3 auf Seite 24). Somit kann ein Angreifer durch das einfache Versenden von ATIM-Nachrichten dafür sorgen, dass bestimmte oder auch alle Stationen in Reichweite wach bleiben. Bei Geräten mit begrenzten Energieressourcen stellt dies eine große Gefahr dar. Diese werden bei einem andauernden Angriff ihre Energiereserven verbrauchen und ihren Betrieb zwangsweise einstellen müssen. Stajano und Anderson [110] bezeichnen diese Form von Angriffen als *Sleep Deprivation Torture Attacks*. Da im Zeitalter des *Ubiquitous Computing* [120] immer mehr kleine und mobile Geräte

mit begrenzten Energiereserven existieren, gilt es derartige Angriffe in Zukunft vermehrt zu betrachten. Verschiedene Arbeiten haben sich mit dieser Problematik bereits auseinander gesetzt [83, 97, 99, 121, 21].

Angriffe gegen die Energiesparmechanismen sind sehr effizient und schwierig zu entdecken, da die gefälschten Nachrichten standardkonform sind. Während das Fälschen der TIM oder einer PS-Poll-Nachricht nur innerhalb eines Infrastruktur-BSS möglich ist, ist das Fälschen der Zeitangaben auch in IBSS-Netzen möglich. Der Wirkungsgrad des Angriffs ist in einem Infrastruktur-BSS aber sehr wahrscheinlich größer. Das Fälschen der ATIM-Nachricht stellt besonders in MANETs eine Gefahr dar und kann bei vollständigem Verbrauch der Energiekapazitäten zu einem andauernden DoS-Effekt führen. Ob die Angriffe in der Praxis durchführbar sind, bleibt zu untersuchen. Da die Energiesparmechanismen in den meisten der heute existierenden WLANs keine Verwendung finden, ist die Gefahr durch diese Angriffe eher als gering einzustufen. Allerdings könnte sich dieser Zustand durch die Verbreitung mobiler Geräte in Zukunft ändern.

3.6.4 Angriffe gegen die Distributed Coordination Function

Reservierung des Network Allocation Vectors

Wie in Abschnitt 2.5.1 beschrieben, ist der virtuelle Carrier Sense-Mechanismus und der damit verbundene Austausch von RTS/CTS-Nachrichten ein wichtiger Bestandteil der Zugriffskontrolle zur Vermeidung des Hidden-Station-Problems (vgl. Abbildung 2.10 auf Seite 17). Die Zeitangaben in RTS- und CTS-Paketen bestimmen die Dauer, für die jede Station bei Empfang eines dieser Pakete ihren *Network Allocation Vector* (NAV) reservieren muss. Obwohl die Verwendung des RTS/CTS-Mechanismus optional ist, muss dennoch jede Station in der Lage sein die beiden Pakete zu erkennen und darf während der angegebenen Dauer nicht auf das Medium zugreifen.

Bellardo et al. [13] beschreiben in ihrer Arbeit die Möglichkeit die Zeitangaben innerhalb des RTS oder CTS zu fälschen um somit das Medium zu reservieren. Obwohl auch mit ACK- und Datenpaketen der NAV reserviert werden kann, hat vor allem die Verwendung des RTS-Pakets einen wesentlichen Vorteil. Eine Station, die als Empfänger eines gefälschten RTS-Pakets adressiert ist, wird mit einem CTS-Paket antworten und somit den gefälschten Wert zur Reservierung des NAVs auch an Stationen weiterleiten, die außerhalb der Reichweite des Angreifers liegen. Die maximale Dauer, die für die Reservierung angegeben werden kann, beträgt 32767 µs. Somit muss ein Angreifer 31 Pakete pro Sekunde senden, um das Medium kontinuierlich zu reservieren. Bellardo et al. analysierten die Auswirkungen dieses Angriffs in einem Infrastruktur-BSS unter Verwendung verschiedener APs und NICs. Dabei stellten sie fest, dass keine der verwendeten NICs den NAV standardkonform reservierte, sondern diesen nach kurzer Zeit zurücksetzten. Dadurch konnte lediglich eine kurze Verzögerung der Übertragung aber keine Unterbrechung erreicht werden. Die Versuche wurden sowohl mit RTS-, CTS- als auch ACK-Paketen und der jeweils maximalen Zeitangabe zur Reservierung durchgeführt. Der Standard erlaubt hinsichtlich der Auswertung eines RTS-Pakets eine gewisse Freiheit. Somit ist es möglich den NAV zurückzusetzen, falls kein Datenpaket nach Erhalt eines RTS empfangen wird. Das Rücksetzen des NAV nach Erhalt eines CTS ist allerdings nicht standardkonform und würde eine fehlerfreie Übertragung bei Anwesenheit einer *Hidden Station* verhindern.

Ähnliche Ergebnisse erhielten Chen et al. [34] in ihren Versuchen, bei denen sie ebenfalls eine maximale NAV-Reservierung in RTS-, CTS- und ACK-Paketen angaben. Allerdings konnten sie einen DoS-Effekt erzielen, wenn sie die Pakete an eine nicht existierende Station adressierten. In diesem Fall setzten die Stationen den NAV nicht zurück und griffen für die reservierte Dauer nicht auf das Medium zu.

Neben der ausbleibenden Wirkung des Angriffs bei vielen Geräten, ist ein weiterer Nachteil die hohe Entdeckungswahrscheinlichkeit bei Verwendung eines gefälschten RTS-Pakets, da ein freier Kanal nach Erhalt eines RTS nicht standardkonform ist. Auch die Energieeffizienz des Angriffs ist mit 31 Paketen pro Sekunde für eine kontinuierliche Reservierung relativ gering. Eine ausführliche Analyse der Effizienz bei Verwendung eines gefälschten RTS-Pakets ist in der Arbeit von Negi und Rajeswaran nachzulesen [87]. Ein Vorteil des Angriffs bleibt allerdings die theoretische Durchführbarkeit sowohl in Infrastruktur-BSS als auch in IBSS-Netzen.

Ausnutzen von Capture-Effekten

Neben dem virtuellen CS-Mechanismus existieren weitere Eigenschaften der DCF, die eine Durchführung von DoS-Angriffen ermöglichen. Hierzu zählen verschiedene *Capture-Effekte*, die Situationen beschreiben, in denen der Kanalzugriff zugunsten einer bestimmten Station beeinflusst wird. Einer der bekanntesten Capture-Effekte wird durch das Verhalten des Backoff-Prozesses hervorgerufen. Dieser begünstigt immer die Station, die zuletzt erfolgreich ein Datenpaket übertragen hat [14]. Somit hat eine Station mit hohem Datendurchsatz eine höhere Wahrscheinlichkeit auf den Kanal zugreifen zu können, als eine Station mit weniger Durchsatz. Zhou et al. [130] beschreiben in ihrer Arbeit einen DoS-Angriff, der auf der Übertragung großer Datenmengen zwischen zwei kooperierenden Angreifern und deren gezielte Positionierung innerhalb eines Ad-hoc-Netzes basiert. Diese sogenannte *Colluding Adversaries Attack* (CAA) macht sich den Einfluss von zwei Capture-Effekten zu Nutze, die in den folgenden Abschnitten kurz erläutert werden. Hierfür sind drei physikalische Reichweiten einer Station von Bedeutung. Diese sind die *Transmission*-Reichweite R_{TX}, in der andere Stationen ein Paket korrekt empfangen können, die *Carrier-Sensing*-Reichweite R_{CS}, in der Stationen ein Paket noch erkennen aber nicht mehr korrekt empfangen können, und die *Interferenz*-Reichweite R_{IF}, in der Stationen ein Paket nicht mehr erkennen können, aber dessen Übertragung dennoch zu einer Störung führen kann [126].

DIFS/EIFS-Zeitschema: Wie in Abschnitt 2.5.1 beschrieben, sieht die DCF die Verwendung verschiedener *Interframe Spaces* (IFS) vor. Nach erfolgreicher Übertragung oder Empfang eines Pakets, muss eine Station nur für die Dauer eines DIFS warten. Wird allerdings ein Paket nicht korrekt empfangen, muss eine Station für die längere Dauer eines EIFS warten. Dies kann zu einem Capture-Effekt führen. In Abbildung 3.4 a) ist eine Topologie bestehend aus den vier Stationen STA1 bis STA4 dargestellt. Dabei sind STA1 und STA3 jeweils Sender und STA2 sowie STA4 jeweils Empfänger. Der Abstand zwischen STA1 und STA2 beträgt in den Simulationen von Zhou et al. [130] 100 Meter, der Abstand zwischen STA3 und STA4 lediglich 50 Meter. STA3 und STA4 liegen außerhalb der Transmission-Reichweite von STA1 und außerhalb der Interferenz-Reichweite von STA2. Allerdings befinden sie sich noch innerhalb der Interferenz-Reichweite von STA1. Beendet nun STA1 eine Übertragung zum Zeitpunkt t_0, kann sie nach $t_1 = t_0 + \triangle_{ACK} + DIFS$ mit dem Backoff-Prozess fortfahren. STA3 kann das Paket von STA1 nicht korrekt empfangen und kann daher nach $t_2 = t_0 + EIFS$ mit dem Backoff-Prozess fortfahren. Da $EIFS = \triangle_{ACK} + DIFS$ gilt, ist

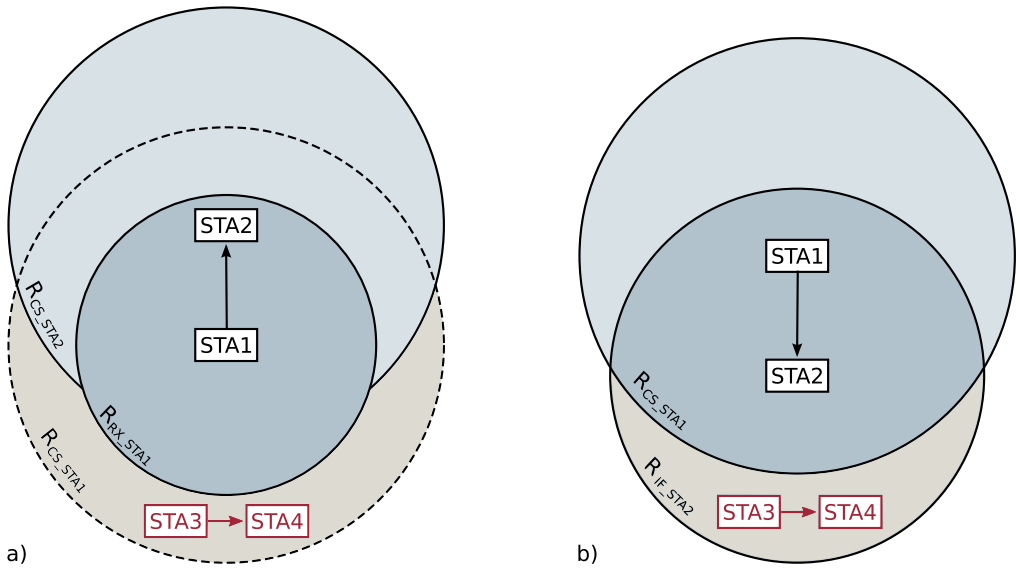


Abbildung 3.4: Topologien für einen Angriff durch Ausnutzen von Capture-Effekten basierend auf a) dem DIFS/EIFS-Zeitschema und b) dem Backoff-Prozess [130]

die Wahrscheinlichkeit auf den Kanal zugreifen zu können in diesem Fall für STA1 und STA3 gleich groß (vgl. Abschnitt 2.5.1). Beendet allerdings STA3 eine Übertragung zum Zeitpunkt t_0, verändert sich die Wahrscheinlichkeit zu Gunsten STA3, da diese schon nach $t_1 = t_0 + \triangle_{ACK} + DIFS$ den Backoff-Prozess starten kann, STA1 allerdings erst nach $t_2 = t_0 + \triangle_{ACK} + EIFS$. Dies resultiert aus dem ACK-Paket von STA4, welches durch STA1 erkannt, aber nicht korrekt empfangen werden kann. In den Simulationen von Zhou et al. [130], in denen sowohl STA1 als auch STA3 die höchst mögliche Bandbreite beanspruchten, ging der Durchsatz der Übertragung zwischen STA1 und STA2 in diesem Falle fast auf Null zurück.

Backoff-Prozess: Ein weiterer Capture-Effekt, der durch zwei kooperierende Angreifer ausgenutzt werden kann, wird durch den Backoff-Prozess verursacht. Das *Contention Window* wird bei jeder gescheiterten Übertragung eines RTS- oder Datenpakets exponentiell erhöht (vgl. 2.5.1 auf Seite 16). Geht man in Abbildung 3.4 a) davon aus, dass STA2 an STA1 senden möchte, wird STA2 zunächst ein RTS-Paket senden. Sollte nun eine laufende Datenübertragung zwischen STA3 und STA4 existieren, wird STA1 den Kanal als belegt erkennen und kann kein CTS senden. STA2 wird daraufhin sein *Contention Window* vergrößern. Auch bei Verzicht auf den RTS/CTS-Mechanismus wird der selbe Effekt erzielt, da STA2 nach Senden eines Datenpakets keine Bestätigung (ACK) von STA1 erhalten würde. Liegen STA3 und STA4 innerhalb der Interferenz-Reichweite von STA2 und außerhalb der CS-Reichweite von STA1 wie in Abbildung 3.4 b), dann werden die Pakete von STA1 nicht fehlerfrei bei STA2 ankommen, falls der Abstand zwischen STA1 und STA2 einen bestimmten Grenzwert überschreitet. Auch durch diesen Effekt können zwei kooperierende Angreifer durch gezielte Positionierung in einem Ad-hoc-Netz einen DoS-Angriff durchführen. Zhou et al. [130] erzielten bei ihren Simulationen mit dieser Methode noch stärkere Effekte als durch das Ausnutzen des IFS-Schemas.

Die Voraussetzung, um DoS-Angriffe basierend auf diesen Capture-Effekten erfolgreich durchführen zu können, ist somit die gezielte Positionierung von zwei kooperierenden Angreifern in bestimmten Angriffsregionen. Diese sind in Abbildung 3.4 jeweils hellgrau markiert. Da die beiden

Angreifer sich jeweils in der selben Region befinden, könnten diese auch durch einen Angreifer mit zwei voneinander unabhängig betriebenen NICs realisiert werden. Die genaue Bestimmung der Positionierung dürfte in der Praxis allerdings schwierig sein und somit ist ein Erfolg oder der Wirkungsgrad des Angriffs nicht garantiert. Ein weiterer Nachteil ist der hohe Energieverbrauch da eine kontinuierliche Kommunikation zwischen den beiden Angreifern stattfinden muss. Dennoch gibt es einige Vorteile dieser Methoden, wie der relativ hohe Wirkungsgrad bei korrekter Positionierung und die Unabhängigkeit von der Betriebsart des Netzes. Auch für egoistische Stationen stellen diese Angriffe eine interessante Möglichkeit für die Steigerung des Durchsatzes dar. Da es sich um eine legitime Kommunikation handeln könnte, ist auch die Entdeckungswahrscheinlichkeit sehr gering.

Manipulation von Protokollparametern

Ein Angreifer hat durch Manipulation bestimmter Protokollparameter der DCF die Möglichkeit sich einen Vorteil für den Kanalzugriff zu verschaffen. Dadurch kann er einerseits einen höheren Datendurchsatz erreichen und andererseits durch kontinuierliche Übertragung einen DoS-Effekt bei anderen Stationen verursachen. Zwei Parameter, die ein Angreifer dafür verändern kann, sind die *Interframe Spaces* (IFS) und die Backoff-Dauer.

1. **Manipulation der Interframe Spaces**

   Bei der Verwendung der DCF, ist die Einhaltung der vorgeschriebenen IFSs die Grundvoraussetzung für den gerecht verteilten Kanalzugriff. Ein Angreifer kann sich einen Vorteil verschaffen, indem er kürzere IFSs wählt, beispielsweise immer nur die Dauer eines SIFS abwartet bevor er versucht auf den Kanal zuzugreifen. Als Ergebnis weisen nur noch die ACK- oder CTS-Pakete anderer Stationen eine gleichhohe Zugriffswahrscheinlichkeit auf. Die Erweiterung 802.11e bietet bei Verwendung von EDCA die Möglichkeit die IFSs durch Wahl einer *Access Category* (AC) zu bestimmen, siehe Abschnitt 2.5.1 auf Seite 19. Je höher die Priorität des Datenstroms, desto kleiner ist der korrespondierende IFS (vgl. Abbildung 2.7). Ein Angreifer kann somit für seine Daten immer die höchste Priorität (AIFSN = 2) angeben, um folglich eine höhere Wahrscheinlichkeit für den Kanalzugriff zu erlangen. Thuente et al. [117] zeigen durch Simulationen, dass auf diese Weise und somit konform zum Standard, eine Verringerung des Datendurchsatzes anderer Stationen um 70 Prozent erreicht werden kann. Acharya et al. [6] untersuchen die Auswirkungen eines Angreifers, der das Beachten von IFSs gänzlich ignoriert und unmittelbar nach einem Zeitschlitz versucht auf den Kanal zuzugreifen. Interessanterweise stellen sie in ihren Simulationen einen nur gering reduzierten Durchsatz bei anderen Stationen fest. Ebenfalls ungewöhnlich an dem Ergebnis ist die Tatsache, dass der durchschnittliche Durchsatz des Angreifers geringer ist als der Durchsatz der Stationen, die sich normal verhalten. Ein Grund hierfür könnte die feste und relativ kleine Paketgröße von 150 Bytes sein, welche der Angreifer während der Simulation versendet.

   Auch Guang und Assi [49] beschäftigen sich in ihrer Arbeit mit der Manipulation von IFSs. Sie untersuchen neben den egoistischen Aspekten die Auswirkungen von kürzeren oder längeren SIFSs auf Routing-Protokolle in Verbindung mit dem RTS/CTS-Mechanismus. Ihre Simulationen zeigen, dass durch die Wahl eines kürzeren SIFS und das Senden eines RTS-Pakets, das darauf folgende CTS-Paket erst nach einem Timeout eintrifft und somit verworfen wird. Hierdurch kann die Route oder der Route-Discovery-Vorgang unterbrochen werden. Andererseits wird durch die Wahl eines längeren SIFS und Empfang eines RTS-Pakets, das

Versenden der CTS-Nachricht verzögert, sodass diese beim Empfänger auch erst nach einem Timeout eintrifft. Auch auf diese Weise kann eine Route unterbrochen werden.

2. **Manipulation der Backoff-Dauer**

   Der Backoff-Prozess, vergleiche Abschnitt 2.5.1 auf Seite 16, ist ein weiterer Teil des Standards, den ein Angreifer durch Anpassen der Protokollparameter für seine Ziele ausnutzen kann. Ein Angreifer hat dabei verschiedene Möglichkeiten:

   - Verwendung einer kurzen und konstanten Backoff-Dauer.
   - Definition eines beliebig kleinen *Contention Windows*.
   - Keine oder beliebige Vergrößerung des *Contention Windows* bei Fehlübertragungen.

   Acharya et al. [6] und Thuente et al. [116] untersuchen die Auswirkungen des ersten Ansatzes, bei dem ein Angreifer stets nur einen Zeitschlitz als Backoff-Dauer wählt. Ihr Fokus liegt dabei auf der Erzielung eines DoS-Effekts durch dieses Verhalten. Bei Präsenz nur eines Angreifers zeigt sich in ihren Simulationen allerdings nur eine Reduzierung des Durchsatzes bei anderen Stationen um 40 Prozent. Erst wenn zwei Angreifer vorhanden sind, lässt sich der Durchsatz um 60 Prozent verringern.

   Kyasanur et al. [74] stellen ein System vor, das derartiges Fehlverhalten erkennen und basierend auf einer modifizierten DCF verhindern soll. Der Grundgedanke des Ansatzes besteht in der Zuteilung der Backoff-Dauer an den Sender durch den jeweiligen Empfänger und einer Beobachtung des tatsächlich durchgeführten Backoffs. Wird ein Fehlverhalten entdeckt, kann eventuell eine Bestrafung erfolgen. Sie stellen zur Validierung ihres Ansatzes zwei Modelle für ein falsches Verhalten basierend auf dem Backoff-Prozess auf, das *Persistent Misbehavior Model* und das *Adaptive Misbehavior Model*. Ersteres beschreibt ein Verhalten eines Angreifers nach einer festen Strategie. Dabei quantifizieren sie den Grad des Fehlverhaltens in Prozent als *Percentage of Misbehavior* (PM). Ein PM-Wert x bedeutet, dass eine Station nur $(100 - x)$ Prozent von der tatsächlich zugewiesenen Backoff-Dauer abwartet, bis sie versucht auf den Kanal zuzugreifen. Dieser PM-Wert bleibt im ersten Modell konstant. Der prozentuale Anteil der Backoff-Dauer, die der Angreifer kürzer wartet, bleibt also konstant. Das zweite Modell geht von einer Station mir variierendem PM-Wert aus. Stationen verfahren nach diesem Modell wenn sie vermeiden wollen, dass ein System ihr Fehlverhalten erkennt und sie dafür gegebenenfalls bestraft. Ihre Simulationen zeigen, dass bei Verwendung der Standard DCF ein Angreifer ab einem PM-Wert von 60 Prozent seinen Durchsatz im Vergleich zu legitimen Stationen vervierfachen kann. Ab einem PM-Wert von 80 geht der Durchsatz der anderen Station sogar gegen 0 und kommt somit einem DoS-Angriff gleich. Durch Verwendung ihrer vorgeschlagenen Veränderungen der DCF lässt sich dieser Effekt weitgehend verhindern, führt aber auch zu einem neuen Problem. Ein Angreifer kann einer Sender-Station kurze Backoffs zuweisen, um die eigene Verbindung zu bevorzugen.

   Cárdenas [33] et al. erweitern daher diesen Ansatz um eine korrekte Zuweisung der Backoffs unter der Voraussetzung zu ermöglichen, dass mindestens eine der beiden Stationen ehrlich ist. In ihrem Ansatz basiert die Zuweisung der Backoffs auf auf dem Protokoll von Blum [17] für den sicheren Münzwurf über eine Telefonverbindung und ist für die Verwendung in Ad-hoc-Netzen vorgesehen.

   Auch Guang und Assi [48] beschäftigen sich mit ähnlichen Angriffen innerhalb von Ad-hoc-Netzen. Ihr Fokus liegt auf Verfahren, die gezielt versuchen Erkennungsmechanismen zu umgehen. Dies kann beispielsweise durch die variable Wahl der Backoff-Dauer geschehen, so dass der Durchschnittswert der Backoffs immer über einem gewissen Grenzwert liegt. Auf diese Weise können zumindest Intrusion Detection Systems (IDSs) wie DOMINO [98]

umgangen werden. DOMINO führt zur Erkennung von Angriffen verschiedene Tests durch, die durch derartige Grenzwerte konfiguriert werden können.

Bianchi et al. [15] präsentieren eine weitere interessante Arbeit im Zusammenhang mit der Manipulation von Backoff-Zeiten. Sie untersuchen verschiedene NICs auf die Standardkonformität des Backoff-Verhaltens. Dabei kommen sie zu dem Ergebnis, dass die Fairness der Distributed Coordination Function (DCF) nicht garantiert wird und die Wahrscheinlichkeit eines erfolgreichen Kanalzugriffs vom Hersteller abhängt. In ihren Tests erreichte beispielsweise eine Ralink NIC einen vier mal höheren Durchsatz als eine Atheros NIC. Für letztere existieren bereits Patches[6] für den Linux-Treiber Madwifi, welche den Backoff-Prozess sowie IFSs vollständig ignorieren. Dies zeigt, dass die Manipulation dieser Parameter in der Praxis leicht umsetzbar ist.

Die Manipulation von Protokollparametern scheint auf den ersten Blick speziell für Angreifer interessant zu sein, die einen eigenen Vorteil wie verbesserten Datendurchsatz erreichen möchten (*Greedy Behaviour*). Die Ergebnisse der unterschiedlichen Simulationen haben allerdings gezeigt, dass insbesondere das Ignorieren von IFSs zwar zu einer Verringerung des Durchsatzes anderer Stationen, aber nicht zwangsläufig zur Erhöhung des Durchsatzes beim Angreifer führt [6]. Somit ist dieser Ansatz eher für DoS-Angriffe geeignet. Sowohl der Wirkungsgrad, als auch die Energieeffizienz sind dabei relativ gering. Da es sich bei der Manipulation von IFS um eine harte Verletzung des Standards handelt, ist ein weiterer Nachteil die hohe Entdeckungswahrscheinlichkeit. Die Manipulation der Backoff-Dauer ist hingegen wesentlich schwerer zu entdecken, da niedrige Backoffs aus einem legitimen Auswahlverfahren hervorgehen können. Hier existieren unterschiedliche Simulationsergebnisse, die von einer Verringerung des Durchsatzes anderer Stationen um 60 Prozent bis zur kompletten Unterbrechung reichen. Nur bei letzterem Ergebnis konnte eine Erhöhung des Durchsatzes für den Angreifer erzielt werden und somit wäre dieser Ansatz auch für egoistische Stationen geeignet [74]. Ein Vorteil beider Ansätze bleibt die Unabhängigkeit von der Art des BSS.

3.6.5 Angriffe gegen das Block Acknowledgement

Der Mechanismus des positiven *Block Acknowledgements* ermöglicht das gleichzeitige Bestätigen mehrerer zuvor übertragener Datenpakete, siehe Abschnitt 2.5.1 auf Seite 15. Die Arbeitsgruppe 802.11n greift diesen Mechanismus auf und erweitert ihn insbesondere bei der Behandlung empfangener Pakete. Der Sender spezifiziert innerhalb des ADDBA Requests das Fenster der zu erwartenden Sequenznummern (*Transmission Window*). Dieses wird durch die *Starting Sequence Number* ($WinStart\_B$) und durch die Puffergröße ($WinSize\_B$) festgelegt. Die letzte zu erwartende Sequenznummer berechnet sich als $WinEnd\_B = WinStart\_B + WinSize\_B - 1$. Der Empfänger akzeptiert nach Erhalt eines ADDBA Request nur solche Pakete, dessen Sequenznummern innerhalb des Transmission Windows liegen und speichert diese in den entsprechenden Stellen des Puffers. Pakete außerhalb des Transmission Windows werden verworfen. Während der Entwicklungsphase der Erweiterung 802.11n wurden bereits durch die IEEE verschiedene Schwachstellen in diesem Mechanismus identifiziert, welche für die Durchführung folgender DoS-Angriffe ausgenutzt werden können [32, 95, 96].

- **Fälschen eines Datenpakets mit höherer Sequenznummer:** Ein Angreifer hat die Möglichkeit durch das Senden eines manipulierten Datenpakets mit einer bestimmten Sequenznummer (SN) das Transmission Window zu verschieben. Für die Sequenznummer muss

[6] https://systems.cs.colorado.edu/projects/carp/wiki/WikiStart

gelten: $WinEnd\_B < SN < WinStart\_B + 2^{11}$. Alle Pakete des legitimen Senders werden solange verworfen, bis sie wieder innerhalb des Transmission Windows liegen. Dieser Angriff ist auch möglich wenn Protokolle für Vertraulichkeit und Integrität verwendet werden, da die SN nicht in die Integritätsprüfung mit einbezogen oder verschlüsselt wird. Daher kann ein Angreifer auch alternativ verschlüsselte Pakete abfangen, die SN anpassen und die modifizierten Pakete an den Empfänger weiterleiten. Falls ein Angreifer keine Möglichkeit hat die Pakete anzupassen, oder die Integrität der SN doch geschützt sein sollte, muss ein Angreifer zunächst warten bis er genügend Pakete gesammelt hat. Maximal wären dies 2^{12} Pakete, da dies die maximale Sequenznummer darstellt. Danach kann er, wie zuvor beschrieben, eines der Pakete mit einer höheren SN an den Empfänger senden.

- **Fälschen einer BlockAckReq:** Ähnlich wie bei der Durchführung des obigen Angriffs, kann durch das Fälschen einer BlockAckReq-Nachricht mit höherer SN das Transmission Window verschoben werden. Da es sich bei dieser Nachricht um eine Control-Nachricht handelt besteht hierbei kein Schutz der Vertraulichkeit oder Integrität.

- **Fälschen eines BlockAck:** Ein Angreifer kann nicht korrekt übertragene Datenpakete bestätigen, in dem er eine BlockAck-Nachricht einer anderen Station fälscht. Dies hält den Sender einerseits davon ab diese Pakete erneut zu übertragen und verzögert andererseits die Auslieferung der bereits empfangenen Pakete innerhalb des Puffers an höhere Schichten.

- **Fälschen eines ADDBA Request:** Eine weitere Möglichkeit für einen Angreifer ist das Fälschen des ADDBA Request, welches die erste Nachricht vor dem Beginn des Block Acknowledgement ist [124, 108]. Durch eine falsche Angabe der *Starting Sequence Number* innerhalb dieser Nachricht, wird der Empfänger das Transmission Window falsch positionieren und die Pakete des legitimen Senders verwerfen. Des Weiteren kann ein Angreifer die angegebene Puffergröße beliebig groß wählen und durch mehrfaches Versenden des selben ADDBA Requests versuchen den Speicher des Empfängers zu überlasten. Das Fälschen des ADDBA Requests ist möglich, da es sich hierbei auch um eine Management-Nachricht (*Action Frame*) handelt, die keinen Schutz der Vertraulichkeit oder Integrität bietet.

- **Fälschen einer DELBA-Nachricht:** Die DELBA-Nachricht ist für die Beendigung und die damit verbundene Freigabe der Puffer auf Sender- und Empfängerseite vorgesehen. Da es sich ebenfalls um eine Management-Nachricht handelt, kann auch diese durch einen Angreifer gefälscht und zum vorzeitigen Abbruch des Block Acknowledgement benutzt werden. Diese Möglichkeit ist in der Literatur noch nicht diskutiert worden und würde sich daher für eine genauere Betrachtung eignen.

Alle Angriffe dieser Art können mit nur einem Paket dafür sorgen, dass die Übertragung bis zu 10 Sekunden unterbrochen wird [32]. Dies ist im Vergleich zum vorher beschriebenen Angriff auf die DCF, der durch die maximale Reservierung des NAVs nur eine Unterbrechung von 33 ms erreichen kann, eine deutliche Steigerung. Da es sich bei den gefälschten Nachrichten um standardkonforme Pakete handelt, ist auch die Entdeckungswahrscheinlichkeit sehr gering. Da bereits erste Geräte, die die Draft Version von 802.11n implementieren, auf dem Markt erhältlich sind, stellt sich die Frage, inwieweit diese Sicherheitslücken bei der Implementierung berücksichtigt wurden. Bisher gibt es zu diesen theoretischen Angriffsmöglichkeiten noch keine bekannten Untersuchungen.

3.7 Angriffe gegen 802.11i

Die Erweiterungen der IEEE Arbeitsgruppe 802.11i bieten einen effizienten Schutz der Vertraulichkeit, Integrität und Authentizität von Datenpaketen, siehe Abschnitt 2.6. Allerdings entstehen durch die verwendeten Protokolle und kryptographischen Verfahren neue Schwachstellen, die ein Angreifer für die in den folgenden Abschnitten beschriebenen DoS-Angriffe ausnutzen kann.

3.7.1 Angriff gegen TKIP-Gegenmaßnahmen

Wie in Abschnitt 2.6.1 beschrieben, setzt das *Temporal Key Integrity Protocol* (TKIP) zur Gewährleistung der Integrität auf den *Michael*-Algorithmus [40]. *Michael* gilt allerdings als kryptographisch schwach. Zum einen bietet der Algorithmus für die Berechnung des MIC nur eine Sicherheit von ungefähr 20 Bit [53] und zum anderen ist er invertierbar [123]. Dies bedeutet, dass aus einem Klartext und dem korrespondierenden MIC der Schlüssel abgeleitet werden kann. Ein Angreifer hätte somit theoretisch die Möglichkeit alle zwei Minuten einen erfolgreichen Key-Recovery-Angriff durchzuführen. Um diese Angriffe zu verhindern, führt TKIP entsprechende Gegenmaßnahmen durch. Diese Gegenmaßnahmen sorgen dafür, dass bei einem vermeintlichen Angriff alle TKIP-Operationen für eine Minute pausieren. Optional kann ein Authenticator vorhandene *Pairwise Transient Key* (PTK) verwerfen und Supplicants deauthentisieren. Durch diese Maßnahmen verlängert sich die durchschnittlich Zeit in der ein Key-Recovery-Angriff erfolgreich durchgeführt werden kann auf sechs Monate [53]. Der Mechanismus zur Erkennung von Key-Recovery-Angriffen kann allerdings wiederum für die Durchführung eines DoS-Angriffs ausgenutzt werden. Die Voraussetzung für die Auslösung der Gegenmaßnahmen ist der Empfang zweier Pakete mit einem falschen MIC innerhalb einer Minute. Dabei müssen die *Frame Check Sequence* (FCS), der *Integrity Check Value* (ICV) und der *TKIP Sequence Counter* (TSC) korrekt sein, da diese Werte zuvor in der angegebenen Reihenfolge überprüft werden. Die FCS und der ICV dienen zur Erkennung von Übertragungsfehlern, reichen aber nicht für die Überprüfung der Integrität aus. Die FCS wird über den MAC-Header und den kompletten Frame-Body berechnet. Der ICV wird lediglich über den Datenteil (MSDU) berechnet. Der MIC schützt die Integrität der MSDU, der *Source Address* (SA), der *Destination Address* (DA) und der Priorität innerhalb des Feldes *QoS Control*. Der TSC dient zur Erkennung von Replay-Angriffen. Dieser Zähler ist 48 Bit groß und wird bei der Übertragung jeder MPDU erhöht. Jede Station besitzt einen Replay-Zähler der nach Empfang einer MPDU mit korrekter MIC erhöht wird. Wird ein Paket empfangen, dessen TSC kleiner oder gleich dem aktuellen Replay-Zähler ist, so wird dieses Paket verworfen.

In Abbildung 3.5 auf der nächsten Seite ist der Aufbau einer MPDU bei Verwendung von TKIP dargestellt. Hier ist zu erkennen, dass lediglich der Datenteil, der MIC und der ICV verschlüsselt sind. Der TSC ist in die sechs Felder TSC0 bis TSC5 aufgeteilt. Die Felder TSC2 bis TSC5 dienen dabei als erweiterter Initialization Vector (IV). Da der Paketschlüssel somit von dem aktuellen TSC abhängt, würde eine Veränderung durch einen Angreifer zu einer fehlerhaften MIC und ICV führen. Das Paket würde folglich verworfen werden.

Glass et al. [47] beschreiben in ihrer Arbeit vier Bedingungen, um einen DoS-Angriff basierend auf den TKIP-Gegenmaßnahmen durchführen zu können:

1. Abfangen einer TKIP-verschlüsselten Nachricht vor deren Auslieferung.
2. Modifizieren der Nachricht, so dass der MIC ungültig wird, FCS und ICV aber gültig bleiben.

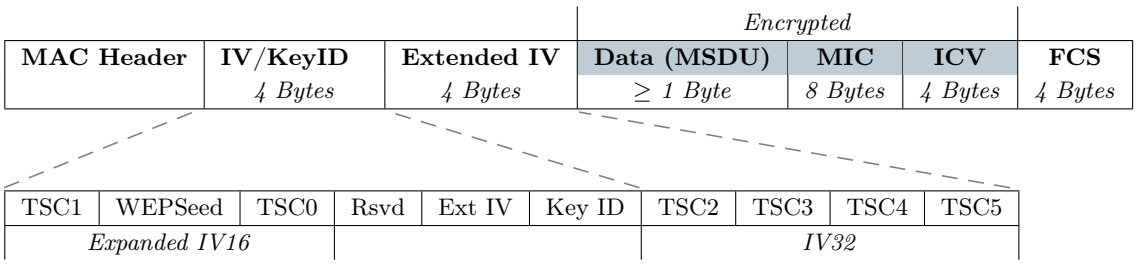


Abbildung 3.5: Aufbau einer MPDU bei Verwendung von TKIP

3. Weiterleiten der modifizierten Nachricht.

4. Sicherstellen, dass zwei modifizierte Pakete innerhalb einer Minute das Ziel erreichen.

Glass et al. untersuchen diesen Angriff innerhalb eines Infrastruktur-BSS. Das Abfangen der Nachrichten erreichen sie durch die Positionierung zwischen Station und Access Point (*Man in the Middle*), siehe Abschnitt 3.2 Punkt 5. Hierbei täuscht der Angreifer die Identität der jeweiligen Kommunikationspartner vor und leitet alle Nachrichten zwischen diesen weiter. Um dies zu erreichen, wenden Glass et al. verschiedene Mechanismen an wie das Fälschen einer *Deauthentication*-Nachricht in Verbindung gefälschter Beacons oder Probe Responses. Ein Problem bei der Weiterleitung der Nachrichten besteht allerdings in der Verzögerung und den Timeouts verschiedener Nachrichten. Insbesondere Bestätigungen (ACKs) haben ein relativ kurzes Timeout-Fenster und müssen somit durch den Angreifer selbst generiert werden. Die Ungültigkeit des MICs erreichen Glass et al. durch Änderungen an der MSDU und entsprechender Anpassung des ICVs, ähnlich wie bei Angriffen auf WEP [18]. Alternativ könnte auch das Prioritätsfeld des Feldes *QoS Control* angepasst werden um den MIC ungültig zu machen. Da der Standard das Erhöhen des Replay-Zählers erst nach Erhalt einer Nachricht mit korrektem MIC vorsieht, bleibt der TSC der modifizierten Nachricht weiterhin gültig. Somit kann die Nachricht innerhalb von einer Minute zwei Mal an den AP weitergeleitet werden um einen DoS-Effekt zu erzielen.

Eine interessante Alternative zum Abfangen der Nachrichten basierend auf MitM wurde in einer Mail von Niels Ferguson diskutiert [41]. Ein Angreifer kann theoretisch jede Nachricht abfangen, indem er nach Empfang des Frame-Bodys ein Störsignal sendet und somit die FCS beschädigt. Der eigentliche Empfänger würde die Nachricht verwerfen und der Angreifer könnte die Nachricht wie oben beschrieben modifizieren und weiterleiten. Ein Nachteil dieser Methode ist allerdings, dass hierfür Änderungen innerhalb der Firmware einer NIC nötig sind. Ohne ein Eingreifen in die Firmware würde eine MPDU erst nach Erhalt des letzten Bits an höhere Schichten übergeben werden. Diese Tatsache erschwert ein solches Vorgehen besonders in Verbindung mit herkömmlichen NICs, da der Firmware-Quellcode meist nicht frei verfügbar ist.

Das Ausnutzen der TKIP-Gegenmaßnahmen ist eine weitere Methode für einen sehr effizienten DoS-Angriff, da mit nur zwei modifizierten Paketen eine Kommunikation für eine Minute unterbrochen werden kann. Die Entdeckungswahrscheinlichkeit ist allerdings sehr groß, weil das Auslösen der Gegenmaßnahmen schon eine Erkennung impliziert. Theoretisch ist dieser Angriff auch in IBSS-Netzen möglich, wenn diese den Aufbau von RSNAs unterstützen.

3.7.2 Angriffe gegen das EAP

He et al. [53] beschreiben verschiedene DoS-Angriffe, die sich gegen das *Extensible Authentication Protocol* (EAP) richten, wie es in Abschnitt 2.6.2 auf Seite 28 im Zusammenhang mit der 802.X Authentication beschrieben wurde. Die Nachrichten *EAPOL-Start*, *EAPOL-Success*, *EAPOL-Logoff* oder *EAPOL-Failure* könnten während der Aufbauphase einer RSNA gefälscht werden und eine erfolgreiche Authentisierung verhindern. Da diese Angriffe allerdings nur in einem sehr kurzen Zeitfenster während des RSNA-Aufbaus durchgeführt werden können, ist hierfür eine hohe Präzision erforderlich. Die Effizienz dieser Angriffe ist abhängig von der Dauer, die benötigt wird, um den Vorgang der Authentisierung durchzuführen. Grundsätzlich kann der Effekt ähnlich dessen eines Deauthentication-Angriffs sein. Die genauen Auswirkungen müssten allerdings noch untersucht werden. Da es sich um standardkonforme EAP-Nachrichten handelt, ist die Wahrscheinlichkeit der Entdeckung relativ gering.

Als weiteren DoS-Angriff gegen EAP beschreiben He et al. [53] das Fluten von *Association Requests*. Das EAP sieht für die eindeutige Zuordnung von Anfrage- und Antwort-Nachrichten zu einer Sitzung ein 8 Bit großes ID-Feld im Header jeder EAP-Nachricht vor. Sendet ein Angreifer nun 265 verschiedene Association Requests, so kann ein AP keine neuen Anfragen mehr entgegennehmen. Der Wirkungsgrad dieses Angriffs ist allerdings von der jeweiligen Treiber-Implementierung innerhalb des APs abhängig. Somit fällt dieser Angriff ebenfalls in die Kategorie *Angriffe auf Treiber und Firmware*, die in Abschnitt 3.8 erläutert wird. Generell kann ein Angriff durch das Fluten von Nachrichten leichter erkannt werden als Angriffe, die mit nur wenigen Paketen auskommen.

3.7.3 Angriff gegen den 4-Way-Handshake

Der 4-Way-Handshake soll den Besitz des gemeinsamen *Pairwise Master Keys* (PMK) bestätigen und aus diesem in Verbindung mit verschiedenen Nonces und der MAC-Adressen einen neuen *Pairwise Transient Key* (PTK) generieren, siehe Abschnitt 2.6.2. In Abbildung 3.6 ist der Ablauf des 4-Way-Handshakes sowie die Durchführung des Angriffs abgebildet. Während Nachrichten 2, 3 und 4 bereits durch den neuen PTK geschützt sind, bleibt Nachricht 1 ungeschützt. Um ein Fälschen der Nachricht 1 und somit die Benutzung eines falschen PTKs zu verhindern, wird der generierte PTK zunächst temporär gespeichert und erst benutzt, nachdem Nachricht 3 durch einen korrekten MIC verifiziert wurde. Ein Angreifer kann allerdings versuchen durch Fluten der Nachricht 1 mit unterschiedlichen Nonces den Speicher des Supplicants zu überlasten, um somit einen DoS-Effekt zu erzielen. Für die Durchführbarkeit dieses Angriffs ist ein hohe Zeitpräzision erforderlich. Auch die Wahrscheinlichkeit der Entdeckung ist durch das Fluten wieder relativ hoch.

3.7.4 RSN IE Poisoning

Das RSN *Information Element* (IE) gibt die verfügbaren Funktionalitäten des Authenticators beziehungsweise des Supplicants zum Aufbau einer sicheren Verbindung an. Ein Authenticator gibt dieses IE innerhalb seiner Beacons oder Probe Responses an. Das IE des Supplicants ist in jedem (Re-)Association Request enthalten. Auf Basis der IEs werden die zu verwendenden Sicherheitsparameter ausgehandelt. Um ein Fälschen dieser Angaben zu vermeiden, sieht der Standard das Bestätigen der IEs während des 4-Way-Handshakes vor. Hierbei sendet der Supplicant sein IE noch-

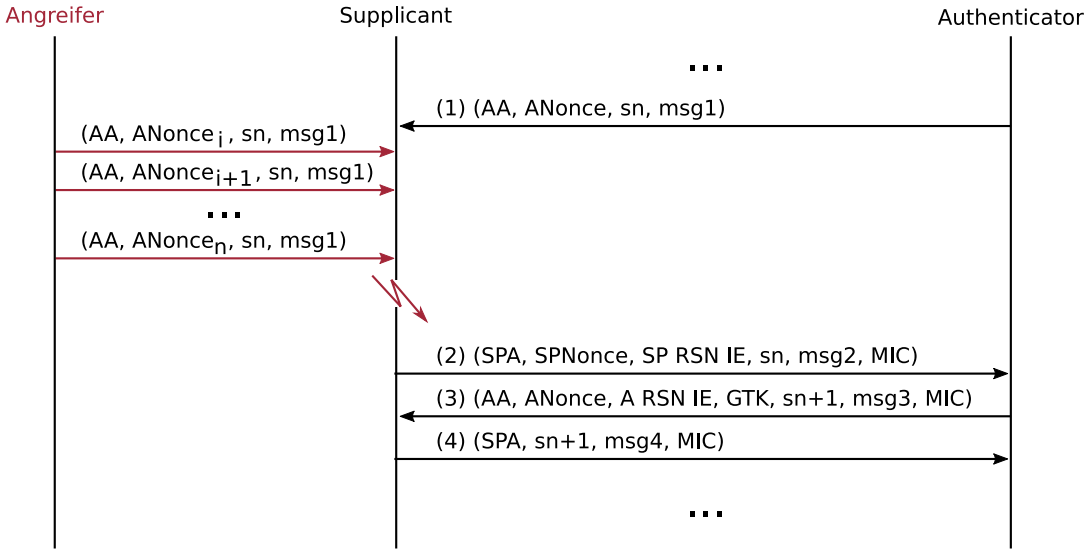


Abbildung 3.6: Ablauf eines Angriffs gegen den 4-Way-Handshake

mals innerhalb der Nachricht 2, der Authenticator innerhalb der Nachricht 3, siehe Abbildung 3.6. Anschließend werden die vorhandenen IEs bitweise verglichen. Sind diese nicht exakt identisch, so wird der Aufbau der RSNA abgebrochen und die Verbindung beendet. Nach He et al. [53] hat ein Angreifer nun die Möglichkeit das Beacon oder die Probe Response eines Authenticators zu speichern, das darin enthaltene IE zu modifizieren und das geänderte Beacon beziehungsweise die geänderte Probe Response wieder zu versenden. Dabei muss die Modifikation so gewählt werden, dass die Aushandlung der Sicherheitsparameter nicht beeinflusst wird. He et al. schlagen beispielsweise eine Modifikation des *Reserved Bits* vor. Nimmt ein Supplicant eine derart gefälschte Nachricht entgegen, kann er sich zunächst authentisieren und den Ablauf zum Aufbau einer RSNA bis zum 4-Way-Handshake durchführen. Der bitweise Vergleich des bereits erhaltenen IEs mit dem korrekten IE aus der Nachricht 3 wird allerdings fehlschlagen und der Aufbau abgebrochen. Der komplette Aufbau der RSNA muss als Folge wiederholt werden. Ein derartiger Angriff wäre somit sehr effizient. Da der Auslöser des Abbruchs das Beacon beziehungsweise die Probe Response war, ist der Angriff sehr schwer zurückzuverfolgen und somit auch die Entdeckungswahrscheinlichkeit sehr gering.

3.8 Angriffe gegen Treiber und Firmware

Gerätetreiber sind aus zwei Gründen attraktive Ziele für Angriffe unterschiedlicher Art. Erstens werden Gerätetreiber meist durch die jeweiligen Hersteller entwickelt, was bei mangelndem Wissen über Funktionen und Schnittstellen des Betriebssystems zu Fehlern und Schwachstellen führen kann. Hersteller von Hardwarekomponenten stehen unter enormen Zeitdruck, um möglichst als Erster, Hardware mit neuen Funktionen oder Unterstützung für neue Standards auf den Markt zu bringen. Aus diesem Grund müssen Gerätetreiber in kürzester Zeit entwickelt und getestet werden. Gerätetreiber enthalten daher wesentlich mehr Fehler als beispielsweise der Kernel eines Betriebssystems und sind somit einer der Hauptgründe für Systemfehler in heutigen Betriebssystemen [35, 114]. Der zweite Grund für die Attraktivität von Gerätetreibern für potentielle Angriffe ist die Tatsache, dass diese durch das Betriebssystem meist mit privilegierten Rechten ausgeführt werden. Ein erfolgreicher Angriff könnte somit den uneingeschränkten Zugriff auf ein System ermöglichen.

Ein weiterer Trend, der besonders bei Herstellern von NICs für 802.11-Netze zu beobachten ist, stellt die Auslagerung von Funktionalitäten der Firmware auf Gerätetreiber dar. Somit werden hardwarenahe Funktionalitäten, die zuvor auf dem Gerät selbst integriert und somit geschützt waren, nun für den Benutzer zugreifbar. Dieses Vorgehen eröffnet ebenfalls neue Angriffspunkte, die in den folgenden Abschnitten beschrieben werden.

Da die Realisierbarkeit und Auswirkung der folgenden Angriffe vom verwendeten Gerätetreiber beziehungsweise der verwendeten Firmware abhängig ist, müssen zuvor Maßnahmen für die Identifizierung von Treibern oder Firmware unternommen werden. Es existieren zahlreiche Forschungsarbeiten, die sich ausschließlich mit diesem Thema befassen. Franklin et al. [44] beschreiben in ihrer Arbeit eine Technik für die passive Identifizierung eines Gerätetreibers basierend auf der Analyse des Verhaltens beim *Active Scanning*, siehe Abschnitt 2.5.3. Diese Technik wird als *Passive Fingerprinting* bezeichnet. Eine Alternative zu dieser Technik stellt das von Bratus et al. [20] beschriebene *Active Fingerprinting* dar. Hierbei werden Informationen über Chipsätze, Firmware oder Treiber gewonnen, indem spezielle Nachrichten versendet und die darauf folgenden Antworten analysiert und ausgewertet werden. Diese speziellen Nachrichten entsprechen meist nicht dem 802.11-Standard und rufen somit bei verschiedenen Gerätetreibern unterschiedliche Reaktionen hervor.

3.8.1 Flooding

Ferreri et al. [42] untersuchen in ihrer Arbeit die Auswirkung von Flooding auf verschiedene Access Points innerhalb eines Infrastruktur-BSS. Flooding (Fluten) bezeichnet das Versenden einer großen Anzahl von Nachrichten innerhalb kürzester Zeit. Sie differenzieren dabei zwischen den drei Ansätzen *Probe Request Flood* (PRF), *Authentication Request Flood* (ARF) und *Association Request Flood* (ASRF).

Das Ziel dieser Angriffe ist die Überlastung und Verschwendung von Speicherkapazitäten des Access Points als zentraler Angriffspunkt (*Single Point of Failure*) und somit die Beeinträchtigung jeglicher Kommunikation innerhalb des BSS. Eine Übersicht über die Auswirkungen dieser drei Angriffsansätze auf verschiedene APs ist in Abbildung 3.7 dargestellt. Die Ergebnisse zeigen, dass die Auswirkungen der einzelnen Angriffe von dem verwendeten AP abhängen. Der erfolgreichs-

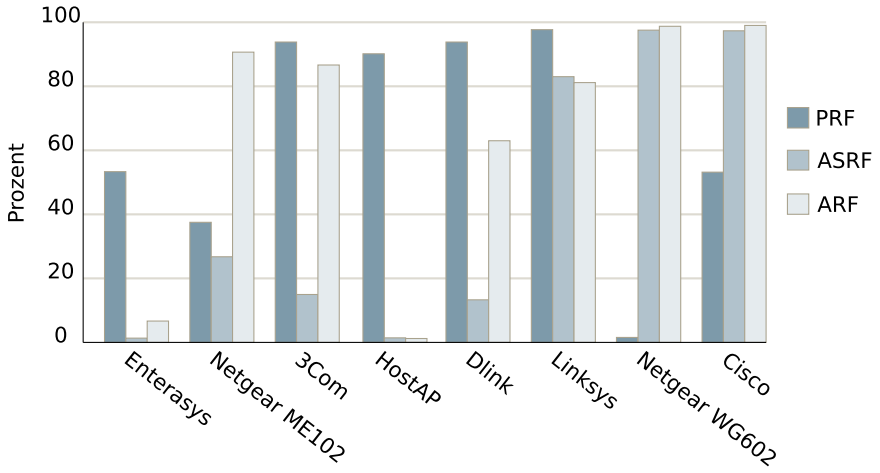


Abbildung 3.7: Paketverlustrate bei verschiedenen Access Points unter Flooding-Angriffen [42]

te Angriff ist der ARF-Ansatz, mit welchem bei den meisten APs ein DoS-Effekt erzielt werden konnte. Eine Voraussetzung für den Erfolg dieses Ansatzes ist allerdings die Änderung der MAC-Adresse für jedes gesendete Paket. Da der Angreifer auf die Probe Responses keine Bestätigung (ACK) sendete, musste der AP jede Probe Response viermal wiederholen, bevor das Versenden abgebrochen wurde. Diese Tatsache und die relativ hohe Injection-Rate von 810 Frames pro Sekunde, bewirkten eine hohe Auslastung der APs und eine ebenfalls hohe Verlustrate der Pakete. Eine weitere interessante Erkenntnis ist der beobachte Effekt bei Verwendung von WEP in Verbindung mit dem Netgear ME102. Hierbei zeigte der PRF-Ansatz keinerlei Wirkung, allerdings verursachte der ARF-Ansatz einen Absturz des APs und der ASRF-Ansatz bewirkte einen DoS-Effekt, der jegliche Kommunikation zwischen legitimen Teilnehmern des BSS verhinderte.

Zusammenfassend kann man festhalten, dass der erreichte Wirkungsgrad der Angriffe von der Implementierung der Firmware in den jeweiligen APs abhängt. Die Energieeffizienz dieser Angriffe ist aufgrund der hohen Senderate von bis zu 810 Paketen pro Sekunde sehr gering und daher für mobile Angreifer eher ungeeignet. Der erzielte DoS-Wirkungsgrad ist im Vergleich zu anderen Angriffen ebenfalls geringer, da auch bei den besten Ergebnissen noch eine Kommunikation stattfinden konnte. Auch die Entdeckungswahrscheinlichkeit ist bei Flooding-Angriffen sehr hoch.

3.8.2 Stack Overflow

Butti und Tinnès [28] untersuchen in ihrer Arbeit Schwachstellen von Treibern verschiedener 802.11-NICs. Um zunächst Schwachstellen zu identifizieren, setzen sie auf eine spezielle Methode für Software Tests, das *Fuzzing*. Durch Fuzzing sollen auf automatisierte Art und Weise Fehler in der Implementierung von Software gefunden werden. Meist basieren Fuzzing-Tests auf randomisierten Eingabedaten. Butti und Tinnès implementieren einen Fuzzer, der die *Information Elements* (IE) innerhalb eines Beacons oder einer Probe Response zufällig verändert. Besonders in der Längenangabe eines IEs sehen sie ein potenzielles Ziel für mögliche Angriffe, die im einfachsten Fall zu einem DoS-Effekt führen können. Dieser DoS-Effekt kann sich bis zum kompletten Absturz des Systems erstrecken. Ein erweiterter und wesentlich gefährlicherer Angriff könnte allerdings durch einen *Stack Overflow* zusätzlichen Schadcode in das System einschleusen. Da Gerätetreiber, wie bereits erwähnt, mit privilegierten Rechten ausgeführt werden, sind durch einen solchen Angriff

beliebige Befehle auf dem System ausführbar. Ihren Fuzzer implementieren sie mit LORCON[7] und Scapy[8], siehe Abschnitt 4.1. Hierbei verfolgen sie zwei verschiedene Test-Strategien:

1. *IE Random Fuzzing*: Bei diesem Test werden Beacons mit zufällig erzeugten IEs gesendet. Die IEs können dabei jede mögliche Form annehmen. Diese Methode testet, inwiefern die Treiber mit ungültigen IEs innerhalb eines gültigen Beacons umgehen können.

2. *SSID IE Random Fuzzing*: Die gesendeten Beacons beinhalten bei diesem Test immer ein SSID Information Element. Das *Type* Feld des IE ist folglich auf den Wert 0 gesetzt. Die Felder *Length* und *Value* werden hingegen zufällig bestimmt. Diese Methode testet, inwiefern Treiber mit ungültigen SSID IEs innerhalb eines gültigen Beacons umgehen können.

Basierend auf den Testergebnissen des Fuzzings wurde eine Liste von Gerätetreibern ermittelt, bei denen durch die falsche Längenangabe des IEs ein *Stack Overflow* erreicht werden konnte. Diese Liste kann durch Einträge in öffentlichen Datenbanken zur Sammlung von Schwachstellen und Sicherheitslücken wie CVE[9], WVE[10] oder MoKB[11] erweitert werden, siehe Tabelle 3.1.

ID	*Treiber*	*Gerät*	*Mgmt Frame*	*IE*
WVE-2008-0008 [24]	Atheros	Linksys WRT50N	Association Request	Atheros Tag
WVE-2008-0010 [27]	Marvell	Netgear WN802T	Association Request	SSID
WVE-2007-0001 [37]	w29n51.sys	Intel 2915ABG	Beacon	SSID
WVE-2007-0013 [23]	6.0.0.18	D-Link DWL-G650	Beacon	TIM
WVE-2006-0072 [30]	a5agu.sys	D-Link DWL-G132	Beacon	Supported Rates
WVE-2006-0071 [29]	bcmwl5.sys	-	Probe Response	SSID
MoKB-22-11-2006 [25]	wg311nd5.sys	NetGear WG311v1	Beacon/Probe Response	SSID
MoKB-18-11-2006 [26]	ma521nd5.sys	NetGear MA521	Beacon/Probe Response	Supported Rates
MoKB-16-11-2006 [86]	wg111v2.sys	NetGear WG111v2	Beacon	$>$ 1100 Bytes
MoKB-01-11-2006 [85]	Apple Airport	Orinoco-based	Probe Response	invalid
CVE-2006-6332 [22]	Madwifi$<$0.9.2.2	-	Beacon/Probe Response	RSN

Tabelle 3.1: Übersicht bekannter Schwachstellen von Gerätetreibern bei der Auswertung verschiedener Information-Elements mit ungültigen Angaben.

Möchte ein Angreifer einen DoS-Effekt erzielen, so ist das Versenden eines einzelnen Beacons oder einer Probe Response mit modifiziertem IE, das zum Absturz eines APs oder einer Station führt, eine sehr effektive Möglichkeit. Die benötigte Energie für einen solchen Angriff ist minimal. Da es sich bei den modifizierten IEs aber um ungültige beziehungsweise nicht standardkonforme Inhalte handelt, ist die Entdeckungswahrscheinlichkeit dieser Angriffe relativ hoch.

[7] http://802.11ninja.net/lorcon/
[8] http://secdev.org/projects/scapy/
[9] http://cve.mitre.org
[10] http://www.wirelessve.org/
[11] http://projects.info-pull.com/mokb/

3.9 Angriffe auf höheren Schichten

Abgesehen von DoS-Angriffen auf die MAC- und PHY-Schicht des 802.11-Standards existieren noch weitere DoS-Angriffe in kabellosen Netzen, die allerdings auf Protokollen höherer Schichten basieren und somit nicht auf Schwachstellen des Standards zurückzuführen sind. Dazu gehören Angriffe gegen Transportprotokolle wie TCP oder UDP und Angriffe gegen Routingprotokolle wie *Ad-hoc On-demand Distance Vector* (AODV) [94] oder *Optimized Link State Routing* (OLSR) [36], die besonders in MANETs oder Mesh-Netzen von Bedeutung sind. Da Routingprotokolle zwar momentan nicht Bestandteil des Standards sind, aber die Arbeitsgruppe 802.11s [63] sich aktiv mit der Erweiterung des Standards zur Unterstützung von Mesh-Routing beschäftigt, könnten diese Angriffe in Zukunft auch für die MAC-Schicht relevant werden. Aus diesem Grund wird im folgenden Abschnitt ein kurzer Überblick über mögliche Angriffe gegen Routingprotokolle gegeben.

3.9.1 Angriffe gegen Routingprotokolle

Kannhavong et al. [71] geben in ihrer Arbeit eine Zusammenfassung über mögliche Angriffe gegen die bekanntesten Routingprotokolle in MANETs, das AODV-Protokoll und das OLSR-Protokoll. Ähnliche Angriffe werden auch von Hu et al. [54] und Aad et al. [1] diskutiert.

- **Flooding:** Das Ziel des *Flooding*-Angriffs ist es, die Netzwerk-Ressourcen wie Bandbreite oder Latenz sowie Ressourcen einzelner Knoten wie Energie oder Rechenleistung zu verschwenden. Beispielsweise kann bei der Verwendung des AODV-Protokolls eine große Anzahl an RREQa in kurzer Zeit an einen nicht existierenden Zielknoten gesendet werden. Da kein Knoten auf diese RREQs antworten wird, werden sie durch das komplette Netz geflutet.
- **Link Withholding:** Hierbei verweigert ein Angreifer die Weitergabe bekannter Verbindungen einzelner oder einer Gruppe von Knoten. Dies kann besonders bei Verwendung des OLSR-Protokolls zu Unterbrechungen kompletter Verbindungen führen.
- **Link Spoofing:** Neben dem Verweigern der Weitergabe von Informationen hat ein Angreifer auch die Möglichkeit gefälschte Informationen zu verbreiten. Diese können das Routing unterbrechen, falls beispielsweise Routen zu nicht existierenden Knoten verbreitet werden.
- **Blackhole:** Basierend auf dem vorherigen Ansatz kann ein Angreifer gefälschte Routinginformationen verbreiten, die andere Knoten in den Glauben versetzen, dass eine bessere Route über den Angreiferknoten existiert. Der Angreifer kann daraufhin alle Pakete, die über ihn geroutet werden, kontrollieren und somit auch verwerfen. Bei Verwendung des AODV-Protokolls kann dies durch das Versenden einer gefälschten RREP-Nachricht mit angepasster Sequenznummer geschehen. Die Sequenznummer muss dabei größer oder gleich der Sequenznummer innerhalb der RREQ-Nachricht sein.
- **Replay:** Besonders in MANETs kann sich die Topologie der Knoten schnell und häufig ändern. Ein Angreifer hat die Möglichkeit empfangene Routing-Nachrichten aufzuzeichnen und zu einem späteren Zeitpunkt erneut zu versenden. Hat sich die Topologie bis dahin geändert, kann es zum Verlust von Nachrichten durch veraltete Angaben von Routen kommen.
- **Wormhole:** Der *Wormhole*-Angriff basiert auf der Kooperation zweier Angreifer, die zusammen über eine private Verbindung unabhängig von anderen Knoten miteinander kommunizieren können, siehe Abbildung 3.8. Dabei kann beispielsweise eine Routinganfrage, die an Angreifer A1 gerichtet ist an Angreifer A2 weitergeleitet werden. Angreifer A2 kann die

Anfrage wiederum an seine Nachbarn weitergeben. Falls A1 und A2 in unterschiedlichen Bereichen eines MANETs liegen, können die weitergeleiteten Informationen zu falschen Routen und zur Unterbrechung von Verbindungen führen.

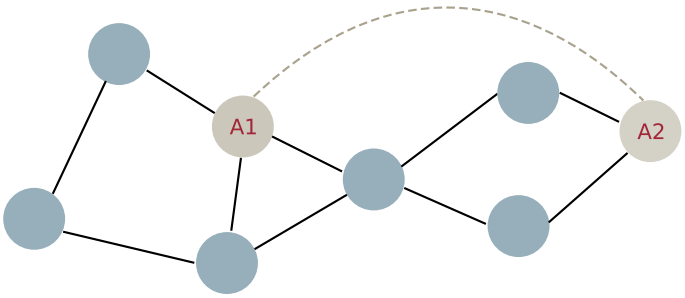


Abbildung 3.8: Beispiel eines Wormholes innerhalb eines MANETs

- **Colluding Misrelay:** Bei diesem Angriff arbeiten ebenfalls zwei Angreifer zusammen. Diesmal befinden sie sich allerdings im gleichen Bereich des MANETs und haben eine direkte Verbindung. Pakete die beispielsweise von Angreifer A1 an Angreifer A2 geroutet werden, können durch Angreifer A2 verändert oder verworfen werden, ohne dass dies durch herkömmliche Methoden zur Erkennung wie *Watchdog* oder *Pathrater* [82] bemerkt werden kann.

3.10 Zusammenfassung

Diese Kapitel hat gezeigt, dass zahlreiche Angriffe gegen die Verfügbarkeit von 802.11-Netzen existieren. Tabelle 3.2 gibt eine Übersicht über alle behandelten Angriffe und in welchen Netzen diese bereits simuliert oder getestet wurden. Ein Großteil dieser Angriffe ist bisher nur in Infrastruktur-BSS-Netzen untersucht worden. Theoretisch sind aber viele der Angriffe ebenfalls in IBSS oder 802.11n-Netzen umsetzbar. WAVE BSS Netze nach der Erweiterung 802.11p sind hingegen durch die Reduzierung an Control- und Management-Nachrichten insbesondere gegen die meisten intelligenten Angriffe resistent. Das *RF Jamming* ist einerseits von allen Angriffen am leichtesten zu entdecken, stellt aber für alle Netzarten den größten Gefahrenfaktor dar, da die physikalische Störung des Funksignals theoretisch immer möglich ist. Für Infrastruktur-BSS-Netze geht eine weitere große Gefahr von dem *Deauthentication*-Angriff oder dem noch fataleren *Autoimmune Disorder* Angriff aus. Diese Angriffe wurden bereits erfolgreich implementiert und sind durch frei verfügbare Tools[12] leicht durchzuführen. Die Implementierung des DoS-Angriffs durch die Reservierung des NAVs hat gezeigt, dass potentiell gefährliche Angriffe in der Praxis ohne Wirkung sein können.

Besonders interessant im Hinblick auf eine genauere Analyse sind daher Angriffe, die theoretisch umsetzbar sind, aber bis jetzt weder implementiert noch simuliert wurden. Zu diesen Angriffen gehört das Fälschen von Management-Informationen, Angriffe gegen Energiesparmechanismen, Angriffe gegen das *Block Acknowledgement* und zwei Angriffe gegen 802.11i, siehe Tabelle 3.2. Zu den Angriffen, die auf dem Fälschen des *Channel Switch Announcement IEs* (vgl. Abschnitt 3.6.2), des *Quiet Elements* (vgl. Abschnitt 3.6.2), der *ATIM-Nachricht* (vgl. Abschnitt 3.6.3) oder der *DELBA-Nachricht* (vgl. Abschnitt 3.6.5) basieren, sind bisher keine Literaturquellen bekannt. Soweit bekannt wurden diese Angriffe zum ersten Mal in dieser Arbeit vorgestellt.

[12] http://aircrack-ng.org

Angriff	802.11 BSS	802.11 IBSS	802.11p WBSS	802.11n
RF Jamming				
Constant Jamming	S,I	I	T	T
Deceptive Jamming	S	T	T	T
Bursty/Busy Jamming	S	T	T	T
Random Jamming	S	T	T	T
Reactive Jamming	S,I	T	T	T
Corruption Jamming	S	T	T	T
Angriffe gegen die MAC-Schicht				
Deauthentication	I	-	-	T
Autoimmune Disorder	I	-	-	T
Fälschen von Management-Informationen				
Fälschen des DS Parameter Sets	T	T	-	T
Fälschen der Channel Switch Announcement*	T	T	-	T
Fälschen des Quiet-Elements*	T	T	-	T
Angriffe gegen Energiesparmechanismen				
Fälschen der TIM/PS-Poll	T	-	-	T
Fälschen der Zeitangaben	T	T	-	T
Fälschen der ATIM*	-	T	-	T
Angriffe gegen die DCF				
NAV-Reservierung	S,I	T	T	T
Capture-Effekte	T	S	-	T
Manipulation von Protokollparametern	I	T	T	T
Angriffe gegen das Block Acknowledgement				
Fälschen des(der) BlockAck(Req)	-	-	-	T
Fälschen der ADDBA	-	-	-	T
Fälschen der DELBA*	-	-	-	T
Angriffe gegen 802.11i				
Angriff gegen TKIP-Gegenmaßnahmen	I	T	-	T
Angriffe gegen das EAP	I	T	-	T
Angriff gegen den 4-Way-Handshake	T	T	-	T
RSN IE Poisoning	T	T	-	T
Angriffe gegen Treiber und Firmware				
Flooding (PRF, ARF, ASRF)	I	T	-	T
Stack Overflow	I	T	T	T

Tabelle 3.2: Übersicht existierender Angriffe, die bereits in Infrastruktur-BSS, IBSS, WBSS oder 802.11n-Netzen simuliert (S), implementiert (I), noch nicht getestet aber theoretisch umsetzbar (T) oder nicht umsetzbar sind (-). Angriffe, die in dieser Arbeit zum ersten Mal vorgestellt wurden, sind mit einem * markiert.

4 Umsetzung ausgewählter Angriffe

Das folgende Kapitel beschreibt die Umsetzung von drei der bereits diskutierten Angriffe gegen die Verfügbarkeit und den Aufbau der Testumgebung, in der die Angriffe durchgeführt und untersucht wurden. Die implementierten Angriffe sind der bereits bekannte *Deauthentication*-Angriff und die beiden neu vorgestellten Angriffe, basierend auf dem Fälschen des *Channel Switch Announcement Information Elements*, beziehungsweise des *Quiet Elements* innerhalb eines Beacons. Da, soweit bekannt, die Auswirkungen durch das Fälschen dieser beiden *Information Elements* bisher noch nicht untersucht wurden und diese Angriffe sowohl in Infrastruktur-BSS, IBSS als auch in zukünftigen 802.11n-Netzen theoretisch durchführbar sind, wurden diese für eine genauere Untersuchung ausgewählt. Einen weiteren Grund für diese Wahl stellt die Attraktivität dieser Ansätze für einen potentiellen Angreifer dar, da sie standardkonform sowie energieeffizient sind und somit eine geringe Entdeckungswahrscheinlichkeit aufweisen. Auch sind die Angriffe im Vergleich zu anderen Angriffen relativ leicht umzusetzen und stellen somit eine umso größere Bedrohung für ein WLAN dar. Beispielsweise setzen Angriffe gegen Energiesparmechanismen eine hohe Zeitpräzision der eingeschleusten Nachrichten voraus und sind für einen Angreifer somit aufwendiger umzusetzen. Die folgenden Abschnitte beschreiben die Umsetzung der drei ausgewählten Angriffe im Detail.

- **Deauthentication-Angriff:** Der Deauthentication-Angriff wurde als Referenzangriff implementiert, um die Effizienz sowie den Wirkungsgrad zwischen diesem und den folgenden Angriffen vergleichen zu können. Der Deauthentication-Angriff ist als DoS-Angriff bereits lange bekannt und lässt sich ohne viel Aufwand mit existierenden Programmen wie aircrack-ng durchführen. Die meisten Implementierungen des Angriffs sind allerdings naiv und fluten das Deauthentication-Paket ohne dabei die konkreten Auswirkungen auf angegriffene Stationen zu berücksichtigen. Als Folge ist die Effizienz dieser Umsetzungen sehr gering und vergleichbar mit naivem *Constant Jamming*. Die Anzahl der Pakete, die pro Minute gesendet werden, kann sich auf mehrere tausend belaufen. Aus diesem Grund wurde für die Umsetzung des Angriffs ein anderer Ansatz gewählt. Die Angreifer-Station verfolgt die laufende Kommunikation und sendet nur dann ein Deauthentication-Paket, falls von der Test-Station ein Datenpaket empfangen wurde. So wird garantiert, dass keine unnötigen Pakete durch die Angreifer-Station versendet werden und somit eine maximale Effizienz erreicht wird.
- **Channel-Switch-Angriff:** Bei diesem Angriff verfolgt die Angreifer-Station solange die Kommunikation, bis ein Beacon empfangen wurde. Beacons können dabei nach SSID gefiltert werden, falls nur ein bestimmtes Netz angegriffen werden soll. Wurde ein Beacon empfangen, wird an dieses das *Channel Switch Announcement Information Element* angehängt. Alle übrigen Informationen des Beacons bleiben unverändert. Aus dem *DS Parameter Set* IE wird der Kanal des BSS ausgelesen. Dies ist notwendig, da durch die Überlappung benachbarter Kanäle der Empfangskanal eines Beacons nicht zwangsläufig auch der Kanal des BSS sein muss. Nach dem der Kanal eingestellt wurde, wird das manipulierte Beacon versendet. Alternativ konnte die *Channel Switch Announcement* auch durch einen Unicast-Action-Frame versendet werden, um beispielsweise eine Station gezielt angreifen zu können. Die Tests wurden mit variierenden Werten für *Channel Switch Mode*, *New Channel Number*

und *Channel Switch Count* durchgeführt. Auch der verwendete Kanal und die Modulationsart (802.11a/b/g) wurden variiert. Hierbei ist zu beachten, dass *Dynamic Frequency Selection* (DFS) und somit die Berücksichtigung des *Channel Switch Announcement* IE*s* nur für 802.11a- und aufkommende 802.11n-Netze in Europa verpflichtend ist.

- **Quiet-Angriff:** Dieser Angriff wurde auf die gleiche Weise umgesetzt wie der zuvor beschriebene Channel-Switch-Angriff. Nur wurde diesmal das *Quiet Element* mit variierenden Werten der Felder *Quiet Count*, *Quiet Period* und *Quiet Duration* an das Beacon angehängt. Auch bei dem *Quiet Element* sei noch einmal anzumerken, dass es sich um einen Bestandteil des DFS-Mechanismus handelt.

Bevor auf den Aufbau der Testumgebung und die Durchführung der Tests eingegangen wird, werden zunächst existierende Softwarelösungen und Bibliotheken für die Umsetzung und Ausführung von Angriffen gegen 802.11-Netze vorgestellt.

4.1 Existierende Software und Bibliotheken

Da schon zahlreiche Softwarelösungen und Bibliotheken für Angriffe oder auch Forschungsarbeiten im Bereich von 802.11-Netzen existieren, sollen diese in den folgenden Abschnitten kurz zusammengefasst werden. Die beiden wichtigsten Funktionen sind in diesem Zusammenhang das *Monitoring* und die *Frame Injection*. Das *Monitoring* beschreibt den Vorgang der kontinuierlichen Beobachtung aller kommunizierten Daten innerhalb eines Netzes. In einem 802.11-Netz kann dies durch jede Station in Reichweite geschehen. Die Funktion des Monitorings besteht einerseits in der Gewinnung von Informationen für die Durchführung der später untersuchten Angriffe und andererseits in der Aufzeichnung der vollständigen Kommunikation für eine anschließende detaillierte Analyse der Auswirkungen. Die *Frame-Injection* beschreibt den Vorgang des Einschleusens beliebiger Pakete in ein vorhandenes Netz. Diese Funktionalität ist neben dem Monitoring die zweite wichtige Voraussetzung für die Umsetzung der späteren Angriffe.

4.1.1 Libpcap

Die Unix-Bibliothek libpcap (*Library for Packet Capture*) besteht aus einer C/C++ API für das Aufzeichnen sowie in neueren Version auch das Versenden von Netzwerkpaketen ab OSI-Schicht 2. Für die Aufzeichnung von Paketen bietet die API die Möglichkeit spezifische Filter zu definieren. Libpcap war ursprünglich Bestandteil des Monitoring-Tools tcpdump[1], wurde dann aber als eigenständige Bibliothek weiterentwickelt. Für Windows existiert ebenfalls ein Port der Bibliothek namens WinPcap und auch für andere Programmier- und Skriptsprachen wie Java oder Python existieren Wrapper, die die Verwendung von libpcap in der jeweiligen Sprache ermöglichen. Zahlreiche Anwendungen machen sich diese Bibliothek zu Nutze, um Netzwerkpakete auszulesen und zu analysieren, darunter Wireshark[2], Snort[3] oder Kismet[4]. Wireshark ist ebenfalls ein Monitoring-Tool, das sich durch eine übersichtliche grafische Oberfläche für die detaillierte Analyse einzelner Netzwerkpakete auszeichnet.

[1] http://www.tcpdump.org/
[2] http://www.wireshark.org/
[3] http://snort.org/
[4] http://kismetwireless.net/

4.1.2 Scapy

Scapy[5] ist ein interaktives Python-Programm zur Manipulation von Netzwerkpaketen. Für das Auslesen von Paketen greift Scapy auf libpcap zurück. Scapy unterstützt das Filtern, Verändern, Erstellen und Versenden von Paketen für zahlreiche Protokolle, darunter auch 802.11. Neue Protokolle lassen sich außerdem leicht hinzufügen. Neben der interaktiven Verwendung, kann Scapy auch als Modul in Python-Skripte eingebunden werden, um die Funktionalität für eigene Anwendungen zu verwenden.

4.1.3 Aircrack-ng

Aircrack-ng[6] ist eine Sammlung von kleinen Hilfsprogrammen, die speziell für das Eindringen in WEP und WPA PSK geschützte 802.11-Netze entwickelt wurden. Die Aufgaben der einzelnen Programme sind vielfältig und reichen vom einfachen Einstellen des *Monitor Modes* (airmon-ng), über das Aufzeichnen empfangener Pakete (airodump-ng), das Einschleusen von Paketen (aireplay-ng), bis hin zur Rekonstruktion des WEP-Schlüssels (aircrack-ng). Mit Aireplay-ng ist es möglich beliebige MAC-Pakete zu versenden. Für den Deauthentication-Angriff bietet das Programm bereits eine fertige Lösung an, die das Fluten der Deauthentication-Nachricht ermöglicht.

4.1.4 LORCON

LORCON[7] (*Loss Of Radio CONnectivity*) ist eine Open-Source-Bibliothek für C, die es erlaubt treiberunabhängig eigene 802.11-Pakete zu erstellen und zu versenden. Hiermit lassen sich auf einfache Weise Anwendungen implementieren, die beliebige Pakete in einem 802.11-Netz versenden können. Ein Beispiel für eine solche Anwendung ist File2air[8], welches aufgezeichnete 802.11-Pakete aus einer Datei liest und diese versendet.

4.1.5 FreeMAC

FreeMAC ist ein Projekt der University of California Santa Barbara, das das Ziel verfolgt ein Framework für die Entwicklung und Evaluierung von Multi-Channel-MAC-Protokollen auf herkömmlicher 802.11-Hardware zu ermöglichen [103]. Ähnliche Ziele verfolgen auch die Projekte SoftMac [88], MadMAC [104], FlexMAC [79] und CARP[9]. Als Grundlage für die Implementierung setzen alle fünf Projekte auf die Erweiterung des Linux-Kerneltreibers Madwifi[10] für Atheros basierte Chipsätze. Neben dem Abschalten bestimmter MAC-Funktionen wie automatisches Acknowledgement, dem virtuellen Carrier-Sense-Mechanismus oder dem Backoff-Prozess, bietet FreeMAC eine API für zeitkritische Funktionen an. Diese werden durch einen angepassten Handler implementiert, welcher an Stelle des Beacon-Handlers im Treiber eingebunden wird. Zur Zeit ist FreeMAC noch nicht verfügbar, soll zukünftig aber als Open-Source-Projekt bereitgestellt werden.

[5] http://secdev.org/projects/scapy/
[6] http://aircrack-ng.org
[7] http://802.11ninja.net/lorcon/
[8] http://www.willhackforsushi.com/File2air.html
[9] https://systems.cs.colorado.edu/projects/carp
[10] http://madwifi-project.org

4.1.6 Software Defined Radio

Software Defined Radio (SDR) beschreibt ein Verfahren der Funkübertragung, mit dem eine hohe Flexibilität im Bereich der Signalverarbeitung erreicht werden kann. Der Grundgedanke ist dabei die Signalverarbeitung durch Software mit Hilfe anpassbarer Hardwarekomponenten zu realisieren. Da durch diese Technik insbesondere auf der physikalischen Schicht beliebige Protokolle implementiert werden können, setzen auch Forscher vermehrt auf SDR [19, 70, 76, 84, 106]. Nachteile dieser Technik sind der relativ hohe Einarbeitungsaufwand, die hohen Kosten von bis zu mehreren 1000 Euro und insbesondere im Zusammenhang mit 802.11, die Verzögerung durch Prozessverarbeitung und Laufzeit der Datenbusanbindung [106]. Da bei 802.11 die einzuhaltenden Zeitabstände im Bereich von nur wenigen Mikrosekunden liegen, können zu große Latenzen eine Implementierung durch SDR erschweren. Dies ist momentan auch bei GNUradio[11] der Fall. GNUradio ist eine Open-Source SDR-Platform basierend auf einem frei programmierbaren Mainboard, dem *Universal Software Radio Peripheral* (USRP). Die zentrale Einheit des USRP ist ein Altera FPGA, das über eine USB-2.0-Schnittstelle mit einem PC verbunden werden kann. Der PC fungiert als zentrale Recheneinheit, die die Modulation, Demodulation aber insbesondere auch die höheren Schichten eines Netzwerkstacks implementiert. Das USRP übernimmt nur noch die Aufgabe, das digitale Signal auf die entsprechende Trägerfrequenz zu schiften und in ein analoges Funksignal zu transformieren. Bisher wurde lediglich versucht eine 802.11b-Station mit einer Senderate von 1 Mbps durch GNUradio zu implementieren. Allerdings ist die Latenz der USB-2.0-Schnittstelle zu groß um die Zeitvorgaben des Standards einhalten zu können[12]. Eine Interaktion mit herkömmlichen 802.11-NICs ist daher noch nicht möglich. Eine neue Version des USRP soll durch eine schnellere Datenbusanbindung dieses Problem in Zukunft beheben.

4.2 Aufbau der Testumgebung

Der folgenden Abschnitt erläutert den Aufbau der Testumgebung und die Realisierung der einzelnen Komponenten. Für die Durchführung der Angriffe werden zwei verschiedene Topologien verwendet. Die erste Topologie ist ein Infrastruktur-BSS bestehend aus einem AP, einer Test-Station und einer Angreifer-Station, siehe Abbildung 4.1 a). Als AP stand für die Tests wahlweise ein Cisco Aironet 1130AG für 802.11a/g und ein D-Link DWL-G730 für 802.11b/g zur Verfügung. Die zweite Topologie ist ein IBSS bestehend aus einer Test-Stationen und einer Angreifer-Station, siehe Abbildung 4.1 b). In beiden Topologien sind außerdem eine Ping-Station und eine Monitor-Station vorhanden. Die Ping-Station besitzt die Aufgabe per ICMP-Pings einen konstanten Datenverkehr für die Messungen der Monitor-Station zu erzeugen.

4.2.1 Realisierung der Ping-Station

In der ersten Topologie ist die Ping-Station, ein Samsung X20 Laptop mit Linux-Kernel 2.6.24, per Ethernet mit dem AP verbunden. Diese sendet einen konstanten Datenstrom per ICMP-Pings an die Test-Station. In der IBSS-Topologie ist die Ping-Station gleichzeitig der Ersteller des IBSS und somit direkter Teilnehmer. Der Vorteil bei der Verwendung von ICMP-Pings ist die einfache Erzeugung von Datenpaketen bei der Test-Station, da jeder empfangene Ping mit dem selben Inhalt

[11] http://gnuradio.org
[12] http://gnuradio.org/trac/wiki/FAQ#CanIrun802.11a/b/g/n/withaUSRP

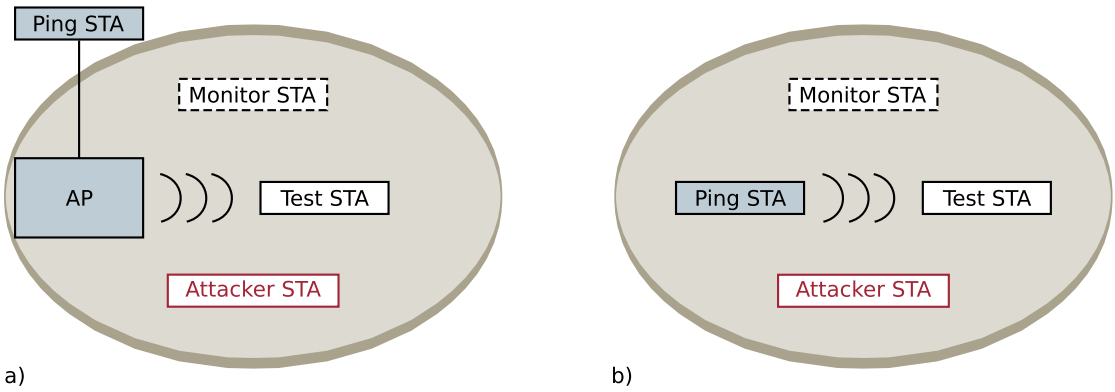


Abbildung 4.1: Topologien der Testumgebung

zurückgesendet werden muss. Während der Tests wurden ICMP-Pings in einem Zeitabstand von 0,1 Sekunden und mit einer Payload-Größe von 5000 Bytes durch den folgenden Befehl versendet:

```
ping -i 0.1 <Test_STA_IP> -s 5000
```

4.2.2 Realisierung des Monitors

Um die Auswirkungen der Angriffe messen und analysieren zu können, wurde während eines Testdurchlaufs der komplette Datenverkehr aufgezeichnet. Für diesen Zweck kam ein IBM ThinkPad T43 mit einer Atheros AR5212 WLAN-NIC und Linux-Kernel 2.6.24 zum Einsatz. Der für Atheros Chipsätze verwendete Linux-Treiber Madwifi[13] hat den Vorteil, dass dieser das Erstellen sogenannter *Virtual Access Points* (VAP) unterstützt. Ein VAP ist ein virtuelles Netzwerk-Device. Diese Methode erlaubt es eine existierende WLAN-NIC parallel in verschiedenen Modi zu betreiben. Ein VAP im Monitor-Modus lässt sich unter Unix-Systemen durch die folgende Abfolge von Befehlen erzeugen:

```
ifconfig ath0 down
wlanconfig ath0 destroy
wlanconfig ath0 create wlandev wifi0 wlanmode monitor
ifconfig ath0 up
```

Alternativ kann das Kernel-Modul des Madwifi Treibers auch direkt mit einem Parameter geladen werden, der automatisch ein Netzwerk-Device im Monitor-Modus erstellt:

```
modprobe ath_pci autocreate=monitor
```

Der Monitor-Modus erlaubt es, alle Pakete der OSI-Schicht 2 und somit der 802.11-MAC-Schicht zu empfangen, ohne dass die Station Teilnehmer eines bestimmten BSS sein muss. Dies kann zu jedem Zeitpunkt allerdings nur auf einem Kanal geschehen. Um den aktuellen Kanal für ein Netzwerk-Device beispielsweise auf den Kanal 13 einzustellen, kann das Linux-Tool iwconfig benutzt werden:

```
iwconfig ath0 channel 13
```

[13] http://madwifi.org

Nachdem der gewünschte Kanal eingestellt ist, kann das Aufzeichnen des Verkehrs beginnen. Hierfür wurde während der Tests tcpdump verwendet. Der Parameter -s spezifiziert die Anzahl an Bytes eines jeden Pakets, die gespeichert werden sollen. Der Wert 0 sorgt in diesem Fall dafür, dass jedes Paket vollständig gespeichert wird.

```
tcpdump -i ath0 -s 0 -w test.cap
```

Für eine genauere Analyse der aufgezeichneten Pakete kann die Capture-Datei später in Wireshark geöffnet werden. Um die Auswirkungen der Angriffe während und nach dem Testdurchlauf zu veranschaulichen, kam das Programm tcpstat zum Einsatz. Mit tcpstat können entweder direkt die empfangenen Pakete eines Netzwerk-Devices oder alternativ mit tcpdump aufgezeichnete Capture-Dateien ausgewertet werden. Für die Veranschaulichung wurde der Durchsatz jeder Station anhand der übertragenen 802.11-MAC-Datenpakete pro Sekunde gemessen. Dies kann durch tcpstat in Verbindung eines entsprechenden Filters (-f) und eines geeigneten Ausgabeformats (-o) erzielt werden:

```
tcpstat -i ath0 -f "wlan[0] == 0x08 && wlan src <Test_STA_MAC>" 1 -o "%R\t%n\t%N\n"
```

Das Ausgabeformat wurde so gewählt, dass es als Eingabe für gnuplot[14] und somit zur direkten Visualisierung verwendet werden kann. Hierbei ist %R die Zeit in Sekunden, %n die Anzahl der Pakete und %N die Anzahl an Bytes.

4.2.3 Realisierung des Angreifers

Ebenso wie die Monitor-Station wird auch die Angreifer-Station durch ein IBM ThinkPad T43 mit einer Atheros AR5212 WLAN-NIC für 802.11a/b/g und Linux-Kernel 2.6.24 realisiert. Diese Wahl wurde getroffen, da wie schon oben beschrieben der Open-Source-Treiber Madwifi für Atheros Chipsätze unter Linux ein hohe Flexibilität bietet und leicht zu manipulieren ist. Der Treiber wurde in der Version 0.9.4 verwendet. Um die für die Angriffe notwendige Frame-Injection zu ermöglichen, muss ebenfalls ein Netzwerk-Device im Monitor-Modus erstellt werden. Die Implementierung der Angriffe wurde durch Scapy realisiert. Obwohl die Umsetzung der Angriffe in einer Skriptsprache wie Python den Nachteil einer größeren Zeitverzögerung mit sich bringt, bleiben im Vergleich zu komplexeren Programmiersprachen wie C
C++, die geringe Einarbeitungszeit sowie die Reduzierung auf wesentliche Code-Bestandteile als große Vorteile bestehen.

4.3 Durchführung der Tests

Die Angriffe wurden in Büroräumen der Universität Ulm getestet. Da die Universität Ulm über ein flächendeckendes WLAN verfügt, wurde für die Tests ein Kanal mit möglichst geringer oder vorzugsweise keiner Aktivität gewählt. Für die Modulation von 802.11b/g im 2,4-GHz-Bereich war dies der Kanal 10. Da der für die 802.11a-Modulation im 5-GHz-Bereich verfügbare Cisco AP für den Betrieb in Europa gefertigt wurde, war bei diesem DFS standardmäßig aktiviert. Der AP suchte sich bei jedem Startvorgang selbst einen freien Kanal für den 802.11a-Betrieb aus. Ein

[14] http://www.gnuplot.info/

expliziter Kanal ließ sich daher nur für den 802.11g-Betrieb einstellen. Da der gewählte 802.11a-Kanal des APs aber stets im Frequenzbereich zwischen 5,25 GHz bis 5,35 GHz lag, für den DFS in Europa vorgeschrieben ist, war eine explizite Konfiguration des Kanals nicht erforderlich. Soweit in den folgenden Abschnitten nicht anders beschrieben, sind die Tests immer in Verbindung mit dem Cisco AP durchgeführt worden. Der grundlegende Ablauf jedes Testdurchlaufs war wie folgt:

1. Starten der Übertragung von ICMP-Pings durch die Ping-Station an die Test-Station.
2. Starten der Aufzeichnung durch die Monitor-Station.
3. Warten für 10 Sekunden.
4. Durchführung des Angriffs durch die Angreifer-Station.
5. Warten für x Sekunden.
6. Wiederholen von Punkt 4 und 5 (optional).
7. Beenden der Aufzeichnung und der Ping-Übertragung.

Die Durchführung eines Angriffs (Punkt 4) wird in der Python-Implementierung[15] mit Scapy durch die beiden Parameter `INJECT_COUNT` und `INJECT_DELAY` beeinflusst. Der Parameter `INJECT_COUNT` bestimmt die Anzahl an Paketen, die pro Angriff gesendet werden. Der Standardwert für diesen Parameter wurde auf 1 festgelegt. Für Werte größer 1 bestimmt der Parameter `INJECT_DELAY` die Verzögerung nach jedem gesendeten Paket. Dieser Wert wurde standardmäßig auf 0 festgelegt. Soll ein Angriff wiederholt durchgeführt werden, kann dies durch den Parameter `ATTACK_COUNT` angegeben werden. In diesem Falle wird nach jedem Angriff für die Dauer `ATTACK_DELAY` gewartet, bevor er erneut durchgeführt wird. Ist der Parameter `ATTACK_DELAY` auf 0 gesetzt, so wird für die Dauer gewartet, für die ein DoS-Effekt durch den jeweiligen Angriff theoretisch erzielt werden kann. Soweit in den folgenden Abschnitten nicht anders angegeben, werden für die Parameter der Angriffe immer die Standardwerte verwendet.

Parameter	Standardwert
INJECT_COUNT	1
INJECT_DELAY	0
ATTACK_COUNT	1
ATTACK_DELAY	theoretisch maximale Dauer des DoS-Effekts

Tabelle 4.1: Beeinflussende Parameter der Angriffsimplementierung und deren Standardwerte

Die Angriffe wurden bei 15 Geräten unter verschiedenen Betriebssystemen mit teilweise variierenden Treibern getestet. Tabelle 4.2 zeigt eine Übersicht aller getesteten Geräte und der untersuchten Treiber in Abhängigkeit des verwendeten Betriebssystems.

Größtenteils wurde der Linux-Kernel in der Version 2.6.24 verwendet. Lediglich die Treiber iwlagn für die Intel 4965AG und 5100AGN NICs, sowie die Nokia-Modelle 770 und N810 wurden mit anderen Kernelversionen getestet. Der Treiber iwlagn wurde mit einem Kernel der Version 2.6.27 getestet. Bei den Nokia-Modellen kamen die Maemo[16] Versionen 2.2 und 4.1 mit jeweiliger Kernelversion 2.6.16 und 2.6.21 zum Einsatz. Unter den getesteten Geräten konnten fünf in einem 802.11a-Kanal operieren. Die Airport Extreme NIC sowie die Intel NICs 4965AGN und 5100AGN unterstützten außerdem eine Draft-Version der 802.11n-Erweiterung.

[15] Der Quellcode ist auf der beigefügten DVD enthalten.
[16] http://maemo.org

	802.11				Treiber			
Gerät/NIC	a	b	g	n	Linux	Windows	Mac OS	Symbian
Intel 2100B		•			ipw2100 v0.56			
Intel 2200BG		•	•		ipw2200 v1.2.2	XP v9.0.4.39		
Intel 3945ABG	•	•	•		iwl3945 v1.2.0	Vista v10.6.0.46		
Intel 4965AGN	•	•	•	•	iwlagn v1.3.27	Vista v11.1.0.86		
Intel 5100AGN	•	•	•	•	iwlagn v1.3.27	XP v12.0.0.82		
Ubiquiti SRC	•	•	•		madwifi v0.9.4.5	XP v7.7.0.0		
Airport Extreme	•	•	•	•			v1.4.8.0	
Airport Extreme	•	•	•	•			v5.10.38.9	
Intersil ISL3890		•	•		Prism54 v1.2			
Lucent Wavelan		•			Host AP v0.5.7	XP v7.43.0.9		
iPhone 3G		•	•				?	
iPod Touch 2G		•	•				?	
Nokia 770		•	•		cx3110x v0.8.1			
Nokia N810		•	•		cx3110x v2.0.15			
Nokia E51		•	•					?
Nokia E71		•	•					?

Tabelle 4.2: Übersicht der getesteten Geräte, deren unterstützen Modulationsarten und der verwendeten Treiber in Abhängigkeit des Betriebssystems. Ein ? zeigt an, dass bei diesen Geräten keine Möglichkeit bestand den verwendeten Treiber auszulesen.

Im nächsten Kapitel werden die Ergebnisse der verschiedenen Tests vorgestellt und ausführlich analysiert.

5 Analyse und Bewertung der Ergebnisse

Dieses Kapitel gibt eine umfassende Darstellung der Ergebnisse, die bei den Tests der drei im letzten Kapitel beschriebenen Angriffe erzielt wurden. Auf Basis dieser Ergebnisse werden die Auswirkungen der jeweiligen Angriffe auf die verschiedenen Geräte im Detail analysiert. Eine abschließende Zusammenfassung und Bewertung der Ergebnisse ermöglicht die Einschätzung der Risiken, die von diesen Angriffen ausgehen.

5.1 Ergebnisse der Tests

Der folgende Abschnitt stellt die Ergebnisse der getesteten Angriffe vor. Als Grundlage für die Auswertung diente der Bruttodurchsatz der jeweiligen Test-Station. Dieser wurde durch die Monitor-Station anhand der gesendeten MAC-Datenpakete pro Sekunde gemessen.

5.1.1 Deauthentication-Angriff

In Abbildung 5.1 ist die Anzahl der Deauthentication-Nachrichten dargestellt, die pro Minute gesendet werden müssen, um bei den getesteten Geräten einen andauernden DoS-Effekt zu erzielen. Die Ergebnisse variieren sehr stark von nur 11 Nachrichten pro Minute bis 668 Nachrichten pro Minute. Die durchschnittliche Rate von 162 Nachrichten pro Minute und somit 3 Nachrichten pro Sekunde, ist aber immer noch relativ hoch. Eine genauere Analyse der Kommunikation zeigte, dass die Verbindung bei den meisten Geräten nicht nach jeder Deauthentication-Nachricht unterbrochen wurde. In vielen Fällen führte die Nachricht nur zu einer kurzen Sendepause, nach der die Kommunikation fortgesetzt wurde. Erst nach mehrfachen Wiederholungen der Deauthentication-Nachricht wurde die Verbindung unterbrochen und in wenigen Millisekunden durch eine erneute Anmeldung am AP wieder aufgenommen. Dieses Verhalten erklärt die teilweise hohe Anzahl an gesendeten Deauthentication-Nachrichten. Die relativ niedrigen Werte, wie bei der Intel NIC 2100B in Verbindung mit dem Linux-Treiber ipw2200 oder bei der Airport Extreme unter Mac OS X mit Treiberversion 5.10.38.9, sind einerseits durch die relativ hohe Verzögerung einer erneuten Anmeldung von mehreren Sekunden und andererseits durch das frühzeitige Abbrechen nach mehrfachen wiederholten Anmeldeversuchen begründet. Im letzteren Fall blieben die Geräte somit dauerhaft vom AP getrennt. Der Großteil der untersuchten Geräte versuchte aber fortlaufend eine neue Verbindung aufzubauen.

Der Deauthentication-Angriff bewirkte bei den meisten Geräten eine vollständige Unterbrechung der Kommunikation und somit einen DoS-Effekt für die Dauer von einer Minute. Trotz der teilweise sehr hohen Senderate von Deauthentication-Nachrichten wurde dieser Effekt aber nicht bei allen Geräte erzielt. In Abbildung 5.2 sind die Auswirkungen auf die drei Geräte dargestellt, die wäh-

rend des Deauthentication-Angriffs noch einen signifikanten Durchsatz aufweisen konnten. Hier ist zu erkennen, dass besonders die Intel 2200BG NIC unter Windows noch einen konstanten Durchsatz von ungefähr 150 Kbps aufweist. Bei dieser NIC konnte somit, trotz der hohen Senderate von 668 Deauthentication-Nachrichten pro Minute, kein DoS-Effekt im Sinne einer vollständigen Unterbrechung der Kommunikation erreicht werden. Der vorhandene Durchsatz ist bei allen drei Geräten auf die kurze Verzögerung zwischen erneuten Anmeldeversuchen, beziehungsweise der Missachtung mancher Deauthentication-Nachrichten zurückzuführen. Während die Verzögerung zwischen erneuten Anmeldeversuchen nicht durch den Standard vorgegeben ist und somit auch beliebig kurz gewählt werden kann, so stellt die Missachtung von Deauthentication-Nachrichten eine Verletzung des Standards dar.

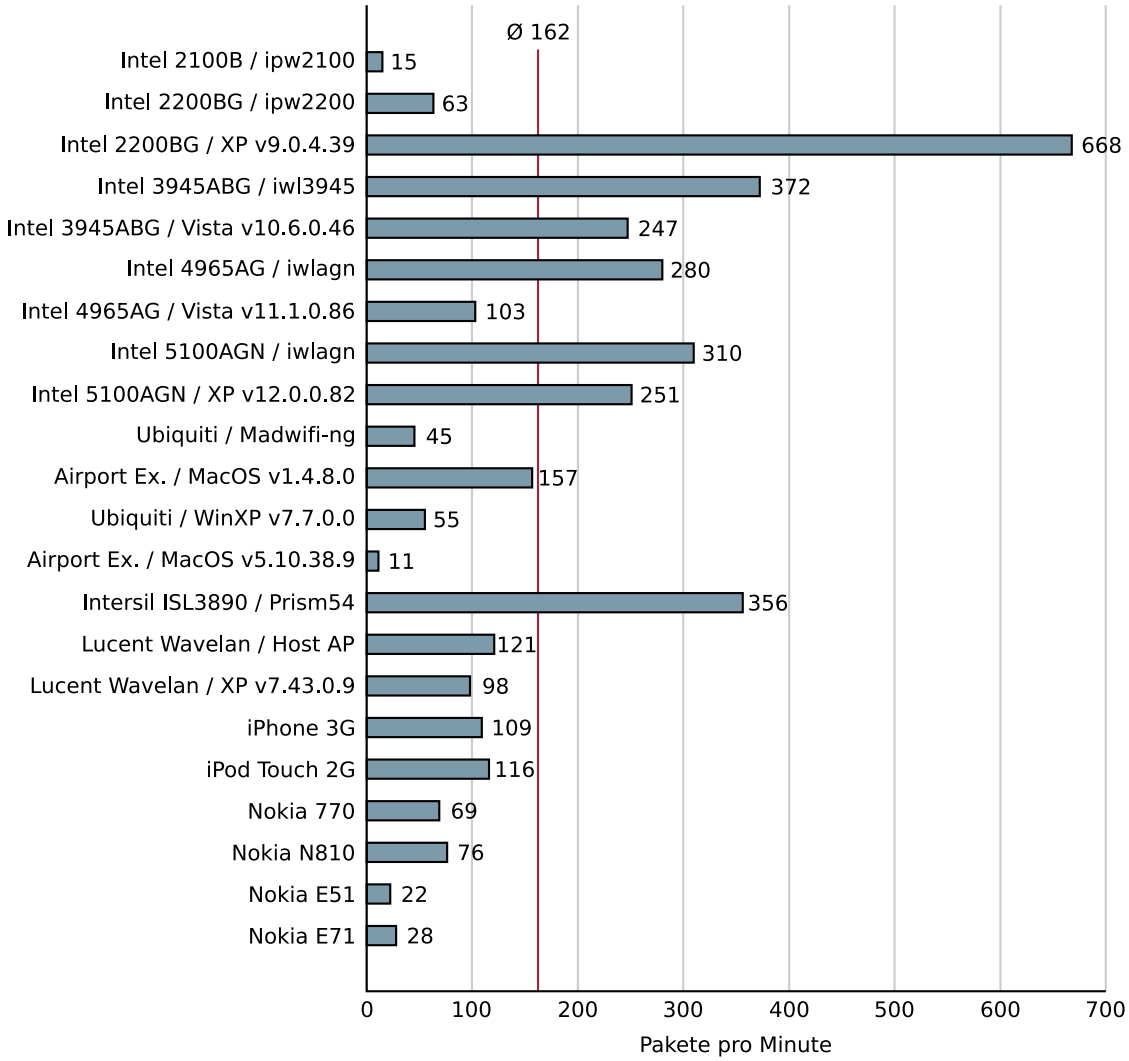


Abbildung 5.1: Anzahl der benötigten Deauthentication-Nachrichten pro Minute, um bei verschiedenen NICs einen andauernden DoS-Effekt aus Anwendersicht zu erzielen

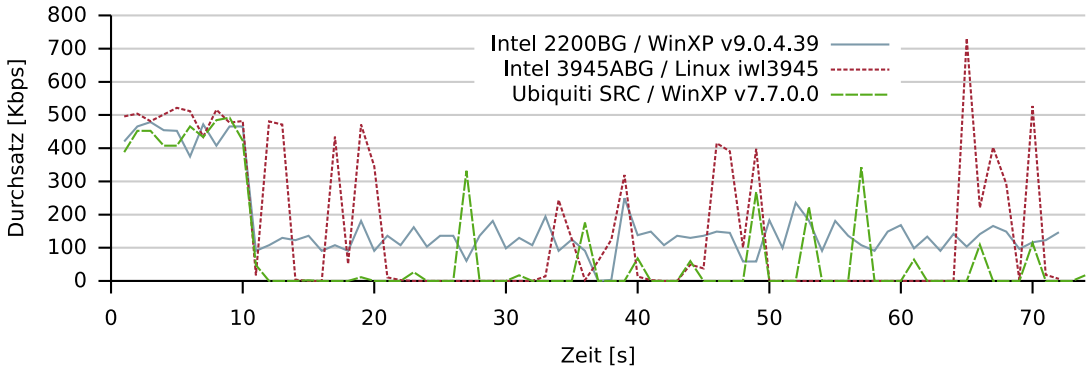


Abbildung 5.2: Auswirkungen des Deauthentication-Angriffs bei drei verschiedenen NICs mit signifikantem Durchsatz

5.1.2 Channel-Switch-Angriff

Die Auswirkungen des Channel-Switch-Angriffs waren sehr unterschiedlich. Die Spezifikation des Standards wurde bei vielen getesteten Geräten nur teilweise oder auch gar nicht implementiert. Bei neun der untersuchten Geräte zeigte der Angriff keine Wirkung. In Anbetracht der Tatsache, dass diese Geräte aber alle nur eine Modulation nach den Erweiterungen 802.11b und 802.11g unterstützen, ist dieses Ergebnis nicht überraschend. Das Verhalten der Geräte ist standardkonform, da die DFS-Mechanismen nur für 5-GHz-Kanäle in Europa vorgeschrieben sind und für alle übrigen Regionen sowie für Frequenzbänder außerhalb des 5-GHz-Bereiches optional bleiben. Bei den restlichen sechs getesteten Geräten war der Angriff erfolgreich. Die folgenden Abschnitte werden die verschiedenen Ergebnisse bei den einzelnen Geräte, die erfolgreich angegriffen werden konnten, im Detail darstellen.

Intel 2200BG

Besonders erstaunlich war das Ergebnis der Intel 2200BG NIC in Verbindung mit dem Linux-Treiber ipw2200. Obwohl dieses Gerät die Modulation nach 802.11a nicht unterstützt und somit die DFS-Mechanismen nicht unterstützen müsste, implementiert der Treiber als Einziger der getesteten den *Channel Switch Mode* 1. Somit war es möglich die Übertragung für die maximale Angabe von 255 Beacon-Intervallen zu unterbrechen. Da die verwendeten APs ein Beacon-Intervall von 100 TUs konfiguriert hatten, betrug die Dauer der Unterbrechung 26,1 Sekunden[1], siehe Abbildung 5.3. Der Verlauf des Durchsatzes zeigt einen deutlichen Ausschlag unmittelbar nach der Unterbrechung. Dies zeigt, dass einige der zu sendenden Pakete gespeichert und anschließend als Burst übertragen wurden. Nach der Unterbrechung wurde die Übertragung ohne eine erneute Anmeldung fortgesetzt. Für einen andauernden DoS-Effekt sind nur drei gefälschte Beacons pro Minute nötig. Bei einem größeren Beacon-Intervall verringert sich die Anzahl benötigter Beacons, da sich die Dauer der Unterbrechung entsprechend erhöht. Würde ein AP ein maximales Beacon-Intervall von 65535 TUs verwenden, so könnte theoretisch eine maximale Unterbrechung von 4,75 Stunden mit einem einzelnen gefälschten Beacon erreicht werden.

[1] $(100 * 1024\ \mu s * 255)/1000000 = 26,1\ s$

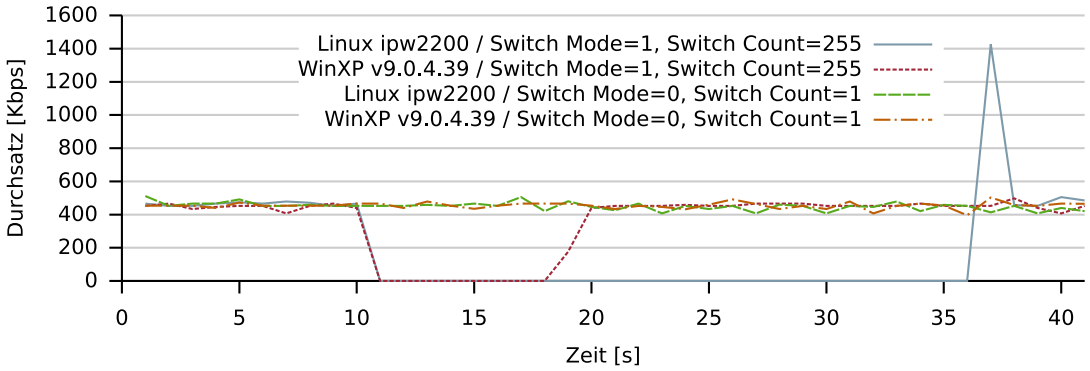


Abbildung 5.3: Auswirkungen von Channel-Switch-Angriffen bei der Intel 2200BG NIC unter Linux und Windows XP

Bei der Angabe eines *Channel Switch Count* von 0 oder 1 wurde kein Effekt erzielt. Dies zeigt, dass der eigentliche Kanalwechsel nicht durchgeführt wird und ein Einsatz des DFS-Mechanismus wie ihn der Standard spezifiziert nicht möglich ist. Bei der Angabe eines ungültigen Kanals war folglich ebenfalls keine Wirkung zu beobachten. Die Nichtdurchführung des Kanalwechsels ist insbesondere deswegen erstaunlich, da der *Switch Mode* von 1 beachtet und somit die Übertragung bis zum Wechsel standardgemäß unterbrochen wurde. Ohne den anschließenden Kanalwechsel macht dieses Verhalten für die Praxis allerdings wenig Sinn.

Unter Windows XP zeigte der Intel-Treiber der Version 9.0.4.39 ein ähnliches Verhalten. Die Unterbrechung der Verbindung beschränkte sich allerdings unabhängig von der Angabe des *Channel Switch Counts* auf maximal sieben Sekunden. Um einen andauernden Angriff durchzuführen, sind daher neun gefälschte Beacons pro Minute nötig. Nach jeder Unterbrechung erfolgte eine erneute Anmeldung mittels Authentication und Association.

Intel 3945ABG

Der unter Linux getestete Treiber iwl3945 war besonders anfällig für den Channel-Switch-Angriff. Die Verbindung wurde bei nahezu jeder Variante des Angriffs vollständig unterbrochen und musste manuell neu aufgebaut werden. Eine Ausnahme war die Angabe eines *Switch Count* von 0 bei einem *Switch Mode* von ebenfalls 0. In diesem Falle wurde die *Channel Switch Announcement* ignoriert, siehe Abbildung 5.4. Ein Blick in die System-Logdatei zeigte, dass die Test-Station bei jedem Angriff versuchte mehrere Probe Requests zu versenden. Bei der Analyse der aufgezeichneten Daten durch die Monitor-Station waren diese Nachrichten allerdings nicht vorhanden. Auch auf dem neuen Kanal der *Switch Announcement* wurde keine Übertragung der Test-Station festgestellt. Der Treiber war somit in einer Art Deadlock-Zustand, welche nur durch die manuelle Neuanmeldung am AP gelöst werden konnte.

Unter Windows Vista zeigte die Intel 3945ABG in Verbindung mit der Treiberversion 10.6.0.46 ein stabileres Verhalten. Alle *Channel Switch Announcements* mit einem *Switch Mode* von 1 wurden komplett ignoriert und bewirkten keinen Effekt. Bei einem *Switch Mode* von 0 und der Angabe eines gültigen Kanals wurde hingegen eine Unterbrechung von fünf Sekunden erzielt. Hierbei berücksichtigte der Treiber auch die Angabe des *Switch Counts* bis zu einem maximalen Wert von 254 Beacon-Intervallen. Eine Angabe von 255 bewirkte überraschenderweise einen sofortigen

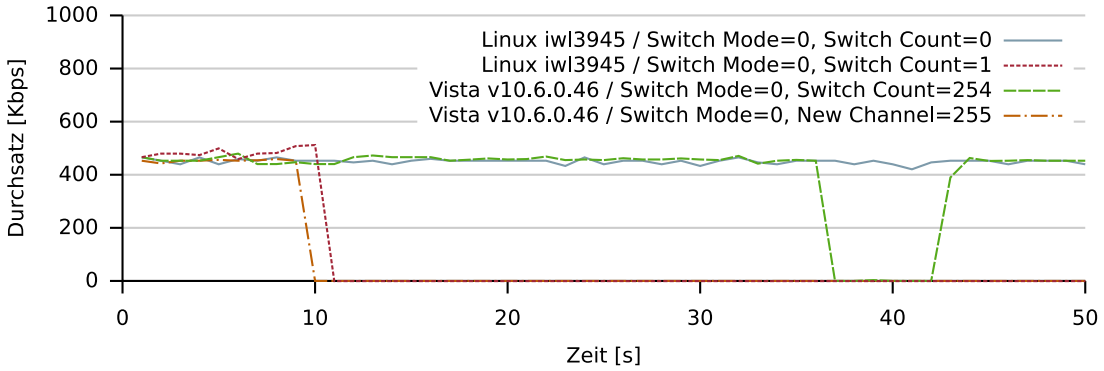


Abbildung 5.4: Auswirkungen von Channel-Switch-Angriffen bei der Intel 3945ABG NIC unter Linux und Windows Vista

Wechsel. Eine vollständige Unterbrechung wurde durch die Angabe eines ungültigen Kanals von 0 oder 255 erzielt. Ähnlich wie bei dem unter Linux getesteten Treiber konnte die Kommunikation hier nur durch eine manuelle Neuanmeldung wieder fortgesetzt werden.

Intel 4965AG/5100AGN

Der Channel-Switch-Angriff erzielte bei den Intel NICs 4965AGN und 5100AGN unter Linux die gleichen Ergebnisse. Der verwendete Treiber iwlagn versuchte bei einem angegebenen *Switch Count* von 0 oder 1 auf den neuen Kanal zu wechseln. Bei den ersten Tests wurde die Verbindung vollständig unterbrochen. Erst durch eine erneute manuelle Anmeldung am AP konnte die Kommunikation fortgesetzt werden. Wurde allerdings auf der Test-Station selbst ein ICMP-Ping initiiert, so dauerte die Unterbrechung nur drei bis fünf Sekunden, siehe Abbildung 5.5. Anschließend wurde die Verbindung auf dem alten Kanal nach einer erneuten Authentication und Reassociation fortgesetzt. Die Angabe eines ungültigen Kanals führte zum selben Ergebnis und bewirkte keine längere Unterbrechung. War der angegebene *Switch Count* allerdings größer als 1, so wurde die Verbindung wiederum vollständig unterbrochen. Der *Switch Mode* hatte auf dieses Verhalten keinen Einfluss.

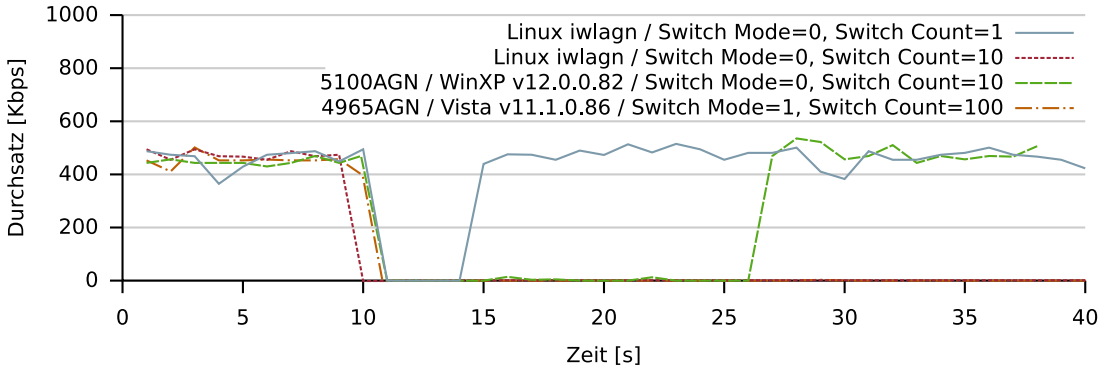


Abbildung 5.5: Auswirkungen von Channel-Switch-Angriffen bei den Intel NICs 4965AG und 5100AGN unter Linux und Windows XP

Die erzielte Unterbrechung betrug bei der Intel 4965AGN NIC unter Windows Vista und der Treiberversion 11.1.0.86 bei jedem Testdurchlauf mindestens 10 Sekunden. Die Angabe eines *Switch*

Count wurde berücksichtigt und bewirkte eine längere Unterbrechung von 15 bis 30 Sekunden. Auch die Angabe des *Switch Modes* wurde berücksichtigt, führte aber bei einem Wert von 1 und einem *Switch Count* größer als 1 in den meisten Fällen zu einer vollständigen Unterbrechung der Verbindung, sodass eine manuelle Neuanmeldung erforderlich war. Die Angabe eines ungültigen Kanals hatte keinen Einfluss auf die Dauer der Unterbrechung.

Unter Windows XP mit der Intel-Treiberversion 12.0.0.82 war das Verhalten der Intel 5100AGN NIC relativ konstant. Jede beliebige Angabe von *Channel Switch Count*, *Switch Mode* und *Channel Number* bewirkte eine Unterbrechung der Verbindung von 10 bis 15 Sekunden und eine anschließend erneute Anmeldung per Authentication und Association.

Ubiquiti SRC

Die Ubiquiti SRC NIC basiert auf einem Atheros Chipsatz und wird daher unter Linux durch den Treiber Madwifi unterstützt. Wie in Abschnitt 3.6.2 schon angesprochen wurde, haben die Entwickler des Madwifi-Treibers eine potentielle Gefahr durch das Fälschen einer *Channel Switch Announcement* erkannt und dies bei der Implementierung berücksichtigt. Die Ergebnisse der Tests belegen dies, da ein *Switch Count* größer als 1 keine Auswirkungen aufzeigt, siehe Abbildung 5.6. Die *Switch Announcement* wird komplett ignoriert. Auch bei der Angabe eines ungültigen Kanals zeigt der Angriff keine Wirkung. Enthält die *Switch Announcement* allerdings einen gültigen Kanal und ein *Switch Count* von 0 oder 1, so wird die Verbindung für acht Sekunden unterbrochen. Die Analyse der aufgezeichneten Daten zeigt, dass die Test-Station versucht auf dem neuen Kanal die Verbindung fortzusetzen. Nach mehrfachen Retransmissions sendet sie einen expliziten Probe Request an den AP und setzt die Verbindung nach erneuter Authentication und Association auf dem alten Kanal fort. Für einen andauernden Angriff sind acht gefälschte Beacons pro Minute nötig. Interessanterweise reagiert der Madwifi-Treiber auf einen wiederholten Angriff inkonsistent. In einem Testdurchlauf wurde acht Sekunden nach dem ersten gefälschten Beacon ein zweites versendet. Das zweite Beacon bewirkte eine Unterbrechung der Verbindung von 56 Sekunden.

Unter Windows XP ist das Ergebnis für den Atheros-Treiber mit der Version 7.7.0.0 ähnlich dem des vorher beschriebenen 5100AGN Treibers. Jede beliebige und somit auch ungültige Angabe der einzelnen Werte führte zu einer Unterbrechung der Verbindung von vier bis fünf Sekunden. Die Verbindung wurde ebenfalls erst nach erneuter Authentication und Association fortgesetzt.

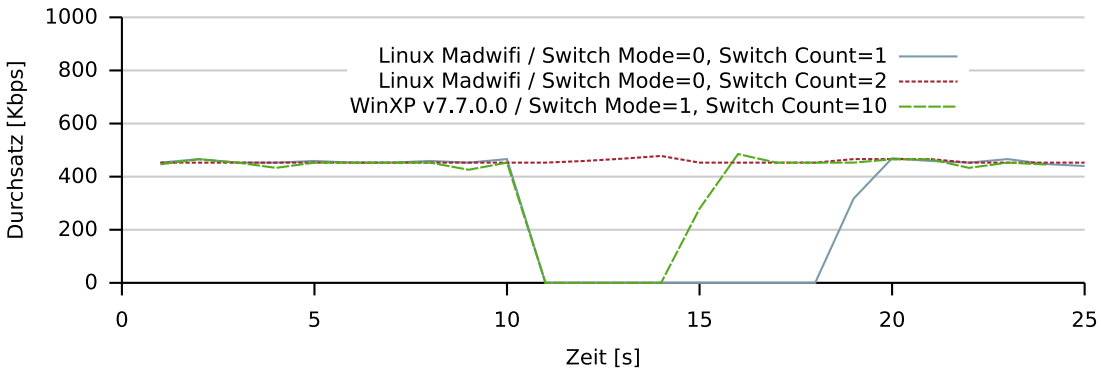


Abbildung 5.6: Auswirkungen von Channel-Switch-Angriffen bei der Ubiquiti SRC NIC unter Linux und Windows XP

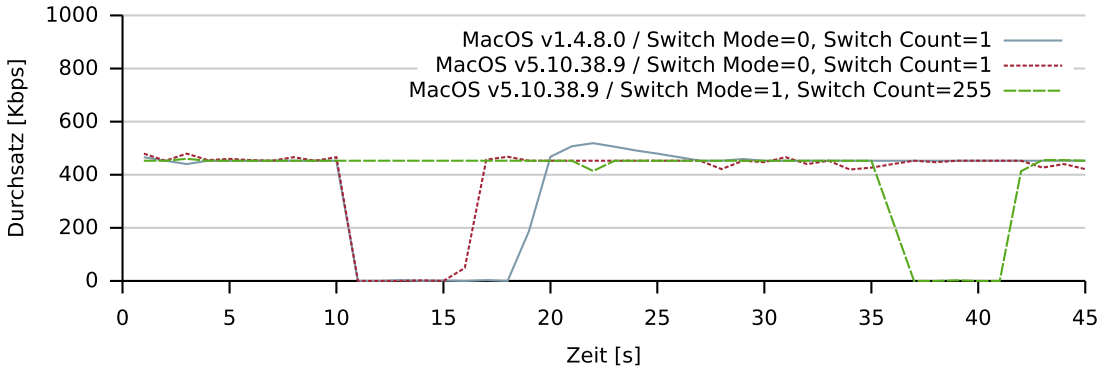


Abbildung 5.7: Auswirkungen von Channel-Switch-Angriffen bei der Airport Extreme NIC unter Mac OS X mit verschiedenen Treiberversionen

Airport Extreme

Die unter Mac OS getestete Airport Extreme NIC mit Treiberversion 1.4.8.0 verhielt sich ähnlich wie die Ubiquiti SRC in Verbindung mit dem Madwifi-Treiber. *Switch Announcements* mit ungültigen Kanalangaben oder einem *Switch Count* größer als 1 wurden ignoriert. Andernfalls betrug die Dauer der Unterbrechung zwischen sechs und neun Sekunden, siehe Abbildung 5.7. Die Analyse der ausgetauschten Nachrichten zeigte, dass die längere Unterbrechung von neun Sekunden durch das Ausbleiben einer Deauthentication-Nachricht des APs begründet war. Sendete der AP hingegen eine Deauthentication-Nachricht, so dauerte es nur sechs Sekunden bis sich die Test-Station durch eine Authentication- und Reassociation-Nachricht erneut mit dem AP verbunden hatte.

In Verbindung mit der Treiberversion 5.10.38.9 wurden abweichende Ergebnisse erzielt. Der angegebene *Switch Mode* wurde zwar ignoriert, allerdings wurde der *Switch Count* berücksichtigt. Auch die Angabe eines ungültigen Kanals war möglich, führte aber zu keiner längeren Unterbrechung. Die erreichte Dauer der Unterbrechung betrug ebenfalls sechs bis neun Sekunden in Abhängigkeit einer gesendeten Deauthentication-Nachricht durch den AP.

Beeinflussende Faktoren

Die zuvor beschriebenen Ergebnisse der Testdurchläufe wurden jeweils unter Verwendung des Cisco AP in Verbindung mit variierenden Kanälen erzielt. Interessanterweise machte es für die Ergebnisse keinen Unterschied ob es sich bei dem ausgewählten Kanal um einen 802.11g-Kanal im 2,4-GHz-Bereich oder um einen 802.11a-Kanal im 5-GHz-Bereich handelte. Dies zeigt, dass die verwendete Modulationsart keinen Einfluss auf die Durchführbarkeit des Channel-Switch-Angriffs hat. Ein Grund hierfür könnte die Vermeidung einer erhöhten Komplexität des Gerätetreibers sein, die durch die Berücksichtigung der Modulationsart bei der Implementierung eintreten würde.

Bei der Verwendung des D-Link DWL-G730 APs wurden bei manchen NICs teilweise stark abweichende Ergebnisse erzielt. Dies war der Fall bei der Ubiquiti NIC in Verbindung mit dem Madwifi-Treiber, bei der Intel 3945ABG NIC in Verbindung mit dem iwl3945 Treiber und bei der Airport Extreme NIC in Verbindung des Mac OS Treibers mit Version 5.10.38.9. In Abbildung 5.8 ist deutlich zu erkennen, dass die Dauer der Unterbrechung von 40 über 70 Sekunden bis hin zur

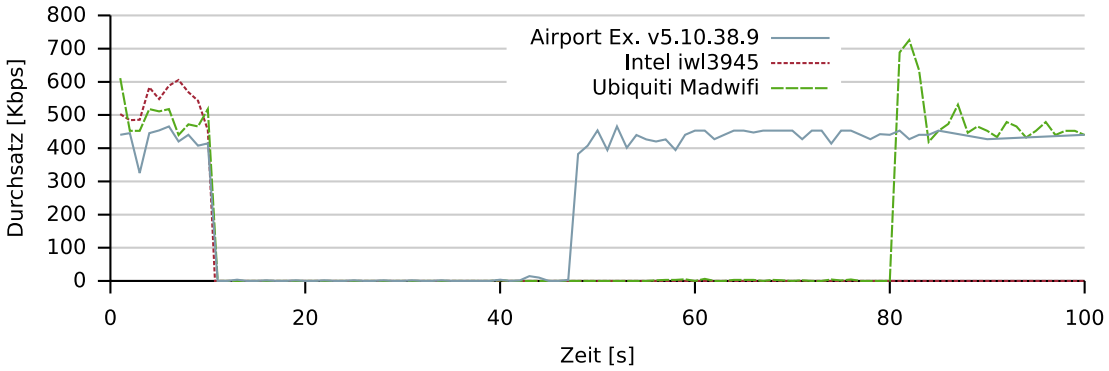


Abbildung 5.8: Unterschiedliche Auswirkungen eines Channel-Switch-Angriffs mit *Switch Mode*=0 und *Switch Count*=1 in Verbindung mit dem D-Link DWL-G730 AP

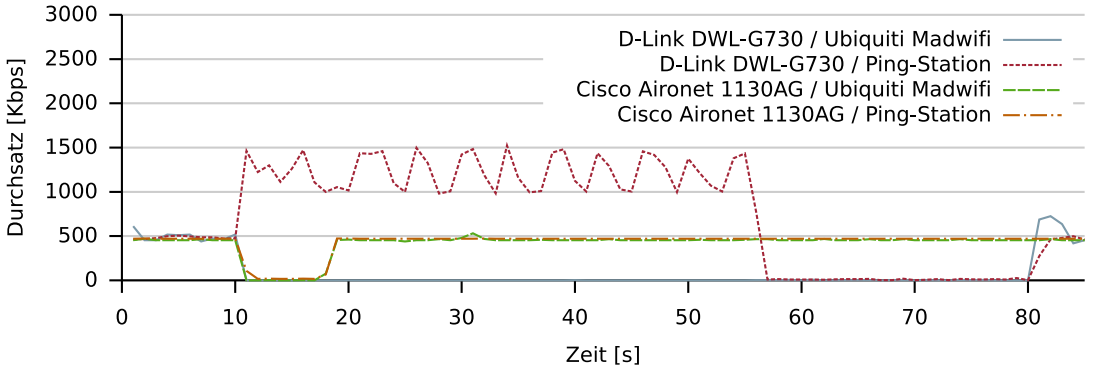


Abbildung 5.9: Auswirkungen eines Channel-Switch-Angriffs mit *Switch Mode*=0 und *Switch Count*=1 auf den Durchsatz der Test- und Ping-Station in Abhängigkeit des verwendeten APs

vollständigen Unterbrechung reicht. Die Unterbrechung variierte von Testdurchlauf zu Testdurchlauf bei allen drei getesteten Geräten zwischen diesen Ergebnissen. Das häufigste Ergebnis war allerdings die vollständige Unterbrechung der Verbindung bis zu einer manuellen Neuanmeldung.

Die genauere Analyse der ausgetauschten Nachrichten zeigte ein unterschiedliches Verhalten der beiden APs. Nachdem die Verbindung unterbrochen wurde und somit die ACKs der Test-Station ausblieben, wurden die Datenpakete der Ping-Station durch den Cisco AP nicht mehr weitergeleitet. Dies ist in Abbildung 5.9 zu erkennen, in der der Durchsatz der Ubiquiti SRC NIC und der Ping-Station unter Verwendung der beiden APs dargestellt ist. Da das Ausbleiben von Antworten durch das Hidden-Station-Problem begründet sein könnte, sendete der Cisco AP mehrfach eine RTS-Nachricht bevor er letztlich mit einer Deauthentication-Nachricht die Test-Station abmeldete. Der D-Link AP sendete hingegen auch bei ausbleibenden ACKs kontinuierlich alle weiteren Datenpakete der Ping-Station. Erst nach einer erfolglosen ARP-Anfrage der Ping-Station wurde die Übertragung durch die Ping-Station selbst eingestellt. Der Zeitpunkt dieser Anfrage variierte zwischen 30 und 60 Sekunden. Eine Deauthentication-Nachricht wurde durch den D-Link AP allerdings während keinem Testdurchlauf gesendet. Diese Tatsache könnte die unterschiedlichen Testergebnisse erklären, da das Ausbleiben der Deauthentication-Nachricht in manchen Treibern den Zustandswechsel zwischen *Verbunden* und *Getrennt* verhindert. Nur im getrennten Verbindungszustand wird durch die meisten Treiber ein neuer Anmeldeversuch unternommen.

5.1.3 Quiet-Angriff

Insgesamt berücksichtigten 5 der 15 getesteten Geräte die Angabe des Quiet-Elements innerhalb des Beacons und konnten somit erfolgreich angegriffen werden. Zu diesen Geräten gehörten die Intel NICs 2200BG, 3945ABG und 4965AGN, sowie die Nokia Internet-Tablets 770 und N810. Alle übrigen Geräte in Verbindung mit den verschiedenen Treibern ignorierten das Quiet-Element.

In Abbildung 5.10 sind die Auswirkungen des Quiet-Angriffs mit einer maximalen Angabe der *Quiet Duration* von 65536 TUs dargestellt. Bei der Intel 2200BG NIC wurde in Verbindung mit dem Linux-Treiber ipw2200 ebenso wie bei der Intel 4965AGN NIC in Verbindung mit dem Windows-Vista-Treiber der Version 11.1.0.86 eine vollständige Unterbrechung von 67 Sekunden erreicht. Dieses Ergebnis entspricht dem zu erwartenden standardkonformen Verhalten für die Verwendung des DFS-Mechanismus. Unter Windows XP wurde bei der Intel 2200BG mit dem Treiber der Version 9.0.4.39 allerdings nur eine Unterbrechung von acht Sekunden erreicht. Der Treiber scheint auch hier eine maximale Dauer festzulegen, wie es schon bei der Behandlung des *Channel Switch Counts* festgestellt wurde. Der *Quiet Count* muss sowohl unter Linux als auch unter Windows jeweils 1 sein, ansonsten wird das Quiet-Element vollständig ignoriert.

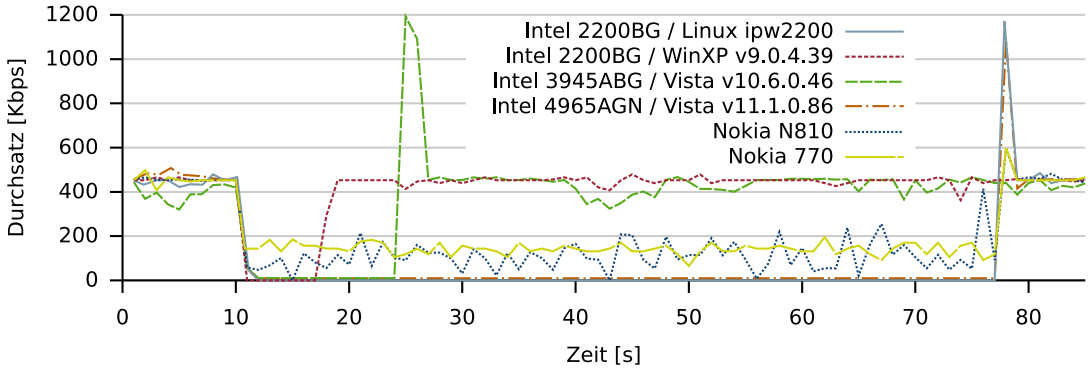


Abbildung 5.10: Auswirkungen des Quiet-Angriffs mit einer maximalen Quiet Duration von 65535 TUs bei angreifbaren Geräten

Auch bei der Intel 3945ABG NIC wurde in Verbindung mit dem Windows-Vista-Treiber der Version 10.6.0.46 nur eine Unterbrechung von maximal 15 Sekunden erreicht. Somit scheint auch dieser Treiber die Dauer des Quiet-Intervalls zu beschränken. Eine weitere Beschränkung durch den Treiber ist das Ignorieren von Quiet-Elementen mit enthaltener *Quiet Period* größer als 0.

Die Nokia Internet-Tablets erlauben eine maximale Angabe der *Quiet Duration* und auch eine beliebige Angabe des *Quiet Counts*. Der Startzeitpunkt des Quiet-Intervalls kann somit bis zu 255 Beacon-Intervalle nach der Übertragung des gefälschten Beacons liegen. Der Durchsatz ging allerdings während des Quiet-Intervalls nicht vollständig zurück. Eine genauere Betrachtung der Nachrichten zeigt, dass nach jedem ICMP-Ping-Request, das erste Fragment der ICMP-Ping-Response noch versendet wurde. Alle nachfolgenden Fragmente wurden nicht versendet. Dieses Verhalten macht eine Kommunikation unmöglich und würde auch die Messungen zum Erkennen von Radar-Aktivitäten während des Quiet-Intervalls erschweren. Eine Ursache kann daher eine fehlerhafte Implementierung des Treibers oder der Firmware der Nokia Internet-Tablets sein.

Die Angaben einer *Quiet Period* oder ein mehrfaches Vorkommen von Quiet-Elementen innerhalb eines Beacons wird von keinem der untersuchten Geräte unterstützt. Da die Intel NIC 2200BG

sowie die Nokia Internet-Tablets ausschließlich die Modulation nach 802.11b und 802.11g unterstützen und somit die DFS-Mechanismen laut 802.11h nicht berücksichtigen müssten, sind die Ergebnisse insgesamt überraschend. Drei der getesteten Geräte, die hingegen auch die Modulation nach 802.11a unterstützen und somit laut Standard die Angabe von Quiet-Elementen berücksichtigen müssten, waren gegen jede Variante des Quiet-Angriffs resistent. Diese waren die drei NICs Intel 5100AGN, Ubiquiti SRC und Airport Extreme. Das beobachtete Verhalten dieser Geräte ist somit zum einen nicht standardkonform und zum anderen ein Verstoß gegen die Vorschriften des ETSI nach Norm EN 301 893 [38]. Als Konsequenz dürften die drei Geräte keine Zulassung für den Betrieb in Europa erhalten. Allerdings ist auch die Zulassung der Geräte fraglich, die die angegebene Dauer des Quiet-Elements begrenzen, da hierdurch Messungen eventuell verfälscht werden könnten. Somit wiesen nur die Intel NIC 2200BG in Verbindung mit dem Linux-Treiber ipw2200 sowie die Intel NIC 4965AGN in Verbindung mit dem Treiber für Windows Vista ein standardkonformes Verhalten auf, das eine Zulassung nach ETSI-Norm erlauben sollte.

5.1.4 Ad-hoc-Modus

Laut Standard sind die DFS-Mechanismen sowohl für Infrastruktur-BSS als auch für IBSS-Netze, also Netze im Ad-hoc-Modus, zu implementieren. Um die Durchführbarkeit und Auswirkungen der Angriffe in IBSS-Netzen einschätzen zu können, wurde eine Auswahl der angreifbaren Stationen exemplarisch im Ad-hoc-Modus untersucht. Diese Geräte waren die Ubiquiti SRC NIC, die Intel 2200BG NIC, die Airport Extreme v.1.4.8.0 NIC und das Nokia Internet-Tablet 770. Bis auf die Ergebnisse der Linux-Treiber in Verbindung mit der Intel 2200BG NIC und der Ubiquiti SRC NIC, waren die Ergebnisse identisch zu denen im vorher untersuchten Infrastruktur-Modus.

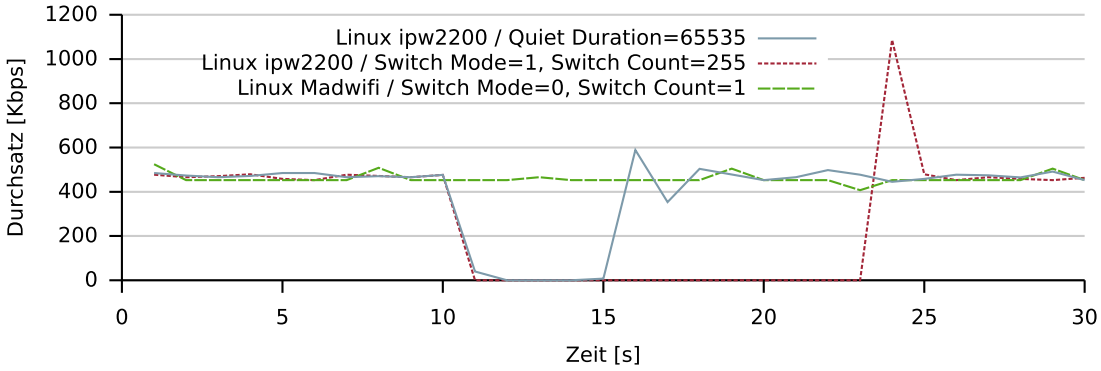


Abbildung 5.11: Abweichende Ergebnisse der Angriffe im Ad-hoc-Modus bei der Intel 2200BG NIC und Ubiquiti SRC NIC

In Abbildung 5.11 sind die abweichenden Ergebnisse im Ad-hoc-Modus dargestellt. Sowohl der Channel-Switch-Angriff als auch der Quiet-Angriff waren im Ad-hoc-Modus bei der Intel 2200BG NIC in Verbindung mit dem Linux-Treiber ipw2200 weniger erfolgreich. Die maximale Unterbrechung betrug bei dem Channel-Switch-Angriff nur 13 Sekunden im Vergleich zu 26 Sekunden im Infrastruktur-Modus. Da das Beacon-Interval ebenfalls 100 ms betrug, hätte die maximale Angabe des *Switch Counts* auch eine Unterbrechung von 26 Sekunden bewirken müssen. Der Treiber scheint somit im Ad-hoc-Modus die maximale Dauer bis zum Kanalwechsel zu beschränken. Ein ähnliches Verhalten zeigte der Treiber beim Quiet-Angriff, bei dem die maximale Angabe des Quiet-Intervalls von 67 Sekunden nur eine Unterbrechung von 7 Sekunden bewirkte.

Die Ubiquiti NIC zeigte sich in Verbindung mit dem Linux-Treiber Madwifi gegen beide Angriffe resistent und ignorierte somit auch die *Channel Switch Announcement*, die im Infrastruktur-Modus zu einer Unterbrechung von acht Sekunden geführt hatte.

5.2 Zusammenfassung und Bewertung

Die Ergebnisse der Tests zeigen, dass viele Varianten der untersuchten Angriffe mit nur einer gefälschten Nachricht eine Verbindung erfolgreich unterbrechen und somit einen DoS-Effekt erreichen können. Die Dauer der Unterbrechung ist allerdings stark von dem verwendeten Gerät, Treiber sowie teilweise auch von dem verwendeten AP abhängig. Tabelle 5.1 zeigt eine Übersicht der benötigten Nachrichten je Angriff bei den getesteten Geräten in Verbindung verschiedener Treiber. Angegeben ist jeweils die Nachrichtenanzahl, die ausreicht, um einen einminütigen DoS-Effekt zu erzielen. Besonders hervorgehoben werden die Ergebnisse, bei denen mit nur einer Nachricht ein DoS-Effekt für eine Minute oder sogar länger erzielt werden konnte.

Die Dauer der Unterbrechung reichte bei den verschiedenen Varianten des Channel-Switch-Angriffs von 5 Sekunden über 26 Sekunden bis hin zur vollständigen Unterbrechung der Verbindung. Vollständig bedeutet in diesem Fall, dass die Geräte die Verbindung nicht eigenständig neu aufgebaut haben. In den meisten Fällen wurde aber nur eine vorübergehende Unterbrechung von 5 bis 10 Sekunden erreicht. Vergleicht man dieses Ergebnis mit dem des Deauthentication-Angriffs, bei dem durchschnittlich 162 Nachrichten pro Minute gesendet werden müssen, so ist der Channel-Switch-Angriff mit nur 1 bis 12 benötigten Nachrichten pro Minute wesentlich effizienter. Ein Nachteil aus Sicht des Angreifers bleibt allerdings die Resistenz mancher Geräte. Insgesamt waren 9 der 15 getesteten Geräte gegen einen Channel-Switch-Angriff resistent. Da aber alle diese Geräte keine Modulation nach 802.11a unterstützen, war dieses Ergebnis zu erwarten. Fünf der erfolgreich angegriffenen Geräte unterstützen die Modulation nach 802.11a. Interessanterweise war der Channel-Switch-Angriff bei diesen Geräten auch auf 802.11b/g-Kanälen wirksam.

Da WLAN-Chipsätze der neusten Generation fast immer eine Unterstützung von 802.11a aufweisen, ist zu erwarten, dass neu auf den Markt kommende Geräte ebenfalls durch einen Channel-Switch-Angriff verwundbar sind. Die Gefahr, die durch diesen Angriff ausgeht, kann somit als hoch eingestuft werden. Der Angriff kann auch für die Durchführung eines Man in the Middle Angriffs benutzt werden, um eine Station dazu zu bringen, sich mit dem AP des Angreifers auf einem anderen Kanal zu verbinden. Auch aus dieser Sicht stellt der Channel-Switch-Angriff eine potentielle Gefahr dar, die aber in dieser Arbeit nicht weiter betrachtet wird.

Die höchste Effizienz bei der Erzielung eines DoS-Effekts wies der Quiet-Angriff mit nur einer benötigten Nachricht pro Minute auf. Hierdurch bleibt sowohl die zur Durchführung notwendige Energie, als auch die Entdeckungswahrscheinlichkeit sehr gering. Der Wirkungsgrad des Angriffs war relativ hoch, auch wenn bei den Nokia-Modellen die Kommunikation nicht vollständig unterbunden werden konnte. Der Quiet-Angriff war bei 5 der insgesamt 15 getesteten Geräte erfolgreich, allerdings bei den Intel NICs 3945ABG und 4965AG nur in Verbindung mit den jeweiligen Treibern für Windows Vista. Da es sich bei den beiden NICs um neuere Modelle handelt und nur diese unter Windows Vista getestet wurden, ist die Wahrscheinlichkeit hoch, dass auch viele der zukünftig erhältlichen WLAN-Geräte insbesondere unter Windows Vista durch den Quiet-Angriff verwundbar sind. Bei zwei weiteren angreifbaren Geräten handelt es sich um aktuelle Nokia Internet-Tablets mit dem auf Linux basierenden Betriebssystem Maemo. Falls der dabei verwendete Treiber cx3110x

nicht aktualisiert wird, ist es auch hier wahrscheinlich, dass zukünftige Modelle, die unter Maemo arbeiten, auf die gleiche Weise angegriffen werden können. Insgesamt geht also auch von dem Quiet-Angriff insbesondere für neue WLAN-Geräte eine hohe Gefahr aus. Interessanterweise unterstützen drei der fünf angreifbaren Geräte keine Modulation nach 802.11a und müssten somit laut Standard das Quiet-Element nicht berücksichtigen.

Da die Angabe eines Quiet-Elements zu den DFS-Mechanismen nach 802.11h gehört, hätten zumindest alle Geräte mit einer Unterstützung von 802.11a, die das *Channel Switch Announcement* IE beachtet haben, auch das Quiet-Element beachten müssen. Da dies bei den drei NICs Intel 5100AGN, Ubiquiti SRC und Airport Extreme aber nicht der Fall war, sind diese Geräte zum einen nicht standardkonform und dürften zum anderen nicht innerhalb der EU betrieben werden [38]. Auch die Beschränkung des Quiet-Intervalls durch die Intel 3945ABG unter Windows Vista ist nicht standardkonform und könnte die Messungen des Kanalzustands verfälschen.

Ein Grund für das Nichtbeachten des Quiet-Elements beziehungsweise Beschränken des Quiet-Intervalls könnte sein, dass manche Hersteller einerseits die potentielle Gefahr des Quiet-Angriffs bei der Implementierung der Treiber berücksichtigen. Andererseits wird bei manchen Implementierungen aber die Management-Funktionalität teilweise losgelöst von den zugrundeliegenden Modulationsarten behandelt. Durch dieses Vorgehen können Code-Bestandteile der Treiber für verschiedene Geräte wiederverwendet, gleichzeitig aber auch mögliche Schwachstellen übertragen werden. Dies ist eine mögliche Erklärung sowohl für den Erfolg des Channel-Switch-Angriffs auf 802.11b/g-Kanälen, als auch für den Erfolg des Quiet-Angriffs bei der Verwendung von 802.11b/g-Geräten. Um diese Vermutung zu verifizieren, müsste ein Einblick in den Quellcode der Firmware und Treiber von den verschiedenen Geräten möglich sein. Insbesondere bei den Windows-Treibern handelte es sich allerdings um proprietäre Closed-Source-Treiber.

Die Untersuchung der Angriffe im Ad-hoc-Modus hat gezeigt, dass die Durchführbarkeit sowohl des Channel-Switch-Angriffs als auch des Quiet-Angriffs von der Netzart unabhängig ist. Neben der hohen Energieeffizienz der Angriffe durch die geringe Anzahl benötigter Nachrichten, ist dies aus Sicht des Angreifers ein weiterer Vorteil gegenüber dem ebenfalls untersuchten Deauthentication-Angriff, der nur in Infrastruktur-Netzen erfolgreich eingesetzt werden kann. Da Channel-Switch-Announcements nicht nur in Beacons sondern auch in Action-Frames mit Unicast-Adressen enthalten sein können, sind außerdem gezielte Angriffe gegen einzelne Stationen eines Netzes problemlos möglich.

Gerät/Treiber	802.11 a	b	g	n	Anzahl benötigter Nachrichten pro Minute Deauthentication	Channel-Switch	Quiet
Intel 2100B		•					
Linux ipw2100 v0.56					15	-	-
Intel 2200BG		•	•				
Linux ipw2200 v1.2.2					63	3	**1**
WinXP v9.0.4.39					668	9	8
Intel 3945ABG	•	•	•				
Linux iwl3945 v1.2.0					372	**1**	-
Vista v10.6.0.46					247	**1-12**	4
Intel 4965AGN	•	•	•	•			
Linux iwlagn v1.3.27					280	**1-12**	-
Vista v11.1.0.86					103	**1-6**	**1**
Intel 5100AGN	•	•	•	•			
Linux iwlagn v1.3.27					310	**1-12**	-
WinXP v12.0.0.82					251	4-6	-
Ubiquiti SRC	•	•	•				
Linux Madwifi v0.9.4.5					45	2-8	-
WinXP v7.7.0.0					157	12	-
Airport Extreme	•	•	•	•			
Mac OS X v1.4.8.0					55	7-10	-
Mac OS X v5.10.38.9					11	7-10	-
Intersil ISL3890		•	•				
Linux Prism54 v1.2					356	-	-
Lucent Wavelan		•					
Linux Host AP v0.5.7					121	-	-
WinXP v7.43.0.9					98	-	-
iPhone 3G		•	•				
Mac OS X ?					109	-	-
iPod Touch 2G		•	•				
Mac OS X ?					116	-	-
Nokia 770		•	•				
Linux cx3110x v0.8.1					69	-	**1**
Nokia N810		•	•				
Linux cx3110x v2.0.15					76	-	**1**
Nokia E51		•	•				
Symbian OS ?					22	-	-
Nokia E71		•	•				
Symbian OS ?					28	-	-

Tabelle 5.1: Vergleich der benötigten Nachrichten für einen andauernden DoS-Effekt mit den drei Angriffen auf verschiedene Geräte in Abhängigkeit der verwendeten Treiber. Erfolglose Angriffe sind mit einem - gekennzeichnet.

6 Zusammenfassung und Ausblick

Wireless LANs, basierend auf dem IEEE 802.11-Standard, erfreuen sich einer stetig steigenden Verbreitung und sind schon heute in zahlreichen Anwendungsbereichen eine bevorzugte Variante der drahtlosen Datenkommunikation. Stark wachsende Anwendungsfelder sind beispielsweise die Telefonie oder die Anbindung von Multimedia-Geräten wie Spielekonsolen oder Digitalkameras. Auch in kritischen Anwendungsgebieten, wie beispielsweise dem Gesundheitswesen, haben WLANs bereits Einzug gefunden [39].

Um einen reibungslosen und sicheren Ablauf drahtloser Kommunikation zu ermöglichen, spezifiziert der IEEE 802.11-Standard zahlreiche Mechanismen und Protokolle auf Ebene der PHY- und MAC-Schicht, von denen die wichtigsten in dieser Arbeit vorgestellt wurden. Die Sicherheitsanforderungen Vertraulichkeit und Integrität können zumindest für Datenpakete durch die Verwendung von CCMP (WPA2) innerhalb einer RSNA gewährleistet werden. Die Protokolle WEP und TKIP besitzen hingegen bekannte Schwachstellen und sollten wenn möglich durch CCMP ersetzt werden. Für Kontroll- und Management-Nachrichten bietet der aktuelle Standard jedoch bisher keinen Schutz. Es existieren zwar Bestrebungen durch die Erweiterung 802.11w in Zukunft auch Management-Nachrichten durch CCMP zu schützen, allerdings kann die Verabschiedung durch die IEEE noch einige Zeit dauern, und selbst dann wird der Schutz nur schwer mit vorhandener Hardware zu realisieren sein.

Neben den Sicherheitsanforderungen Vertraulichkeit und Integrität spielt auch die Verfügbarkeit eines WLANs, insbesondere für kritische Anwendungsbereiche, eine immer wichtigere Rolle. Diese zu gewährleisten stellt aber durch die Beschaffenheit eines kabellosen Netzes eine große Herausforderung dar. Aus diesem Grund existieren derzeit zahlreiche Angriffe gegen die Verfügbarkeit, die im Rahmen dieser Arbeit ausführlich diskutiert wurden. Sie können sich sowohl gegen physikalische als auch gegen Mechanismen der MAC-Schicht richten und von einem Angreifer zur Unterbrechung der Kommunikation (*Denial of Service*) oder zum Erlangen eines eigenen Vorteils (*Greedy Behaviour*) genutzt werden. Ein naiver DoS-Angriff gegen die PHY-Schicht ist das *Constant Jamming*, das aber für einen Angreifer die Nachteile eines hohen Energieverbrauchs sowie einer hohen Entdeckungswahrscheinlichkeit mit sich bringt. Ein Vorteil von Jamming-Angriffen aus Angreifersicht bleibt die Durchführbarkeit unabhängig von der Netzart, und somit stellen diese Angriffe auch für zukünftige 802.11n- und 802.11p-Netze eine potentielle Gefahr dar. Vorstellbare Schutzmaßnahmen gegen Jamming-Angriffe sind die Verwendung von Frequenzspreizverfahren wie FHSS und DSSS oder der Wechsel zwischen verschiedenen Modulationsverfahren auf PHY-Ebene. Dies könnte zukünftig durch den Einsatz von *Software Defined Radio* ermöglicht werden.

Intelligente Angriffe richten sich zum großen Teil gegen die Mechanismen der MAC-Schicht. Sie sind dadurch meist schwerer zu entdecken und benötigen oft auch wesentlich weniger Energie, weil sie teilweise mit nur einzelnen gefälschten Nachrichten auskommen, um die Kommunikation für längere Zeit zu unterbrechen. Drei solcher Angriffe wurden im Rahmen dieser Arbeit implementiert, getestet und ausführlich analysiert. Diese sind der optimierte Deauthentication-Angriff, der

Channel-Switch-Angriff und der Quiet-Angriff. Soweit bekannt, wurden die letzten beiden Angriffe in dieser Arbeit zum ersten Mal vorgestellt. Sie richten sich gegen die DFS-Mechanismen nach 802.11h und basieren auf gefälschten *Channel Switch Announcements* beziehungsweise *Quiet Elements* innerhalb gesendeter Beacons. Die Angabe einer *Channel Switch Announcement* veranlasst einen Kanalwechsel, die Angabe eines *Quiet Elements* die Einstellung der Übertragung für eine spezifizierte Zeitspanne.

Während der klassische Ansatz des Deauthentication-Angriffs, wie bei aircrack-ng, mit dem Fluten von Nachrichten noch mehrere hundert bis tausend gefälschte Nachrichten pro Minute benötigt um einen andauernden DoS-Effekt zu erzielen, so waren bei den Tests des optimierten Deauthentication-Angriffs durchschnittlich nur 162 Nachrichten pro Minute nötig. Dies zeigt, dass die Energieeffizienz schon allein durch das reaktive Verhalten eines Angreifers um ein Vielfaches gesteigert werden kann.

Die Untersuchungen der neu vorgestellten Angriffe, basierend auf gefälschten *Channel Switch Announcements* beziehungsweise *Quiet Elements*, haben gezeigt, dass mit diesen noch einmal eine Steigerung der Energieeffizienz erzielt werden kann. Sowohl mit dem Channel-Switch-Angriff als auch dem Quiet-Angriff genügte bei vielen getesteten Geräten bereits das Versenden einer gefälschten Nachricht, um die Kommunikation für eine Minute oder sogar länger zu unterbinden. Eine vergleichbare Effizienz kann nur mit dem vorgestellten Angriff gegen die TKIP-Gegenmaßnahmen oder mit Angriffen gegen Treiber und Firmware erreicht werden, die einen Geräteabsturz verursachen. Beide Ansätze besitzen allerdings für einen potentiellen Angreifer den Nachteil eines hohen Aufwands zur Umsetzung, da bei ersterem Nachrichten abgefangen werden müssen und bei letzteren bekannt sein muss, welche Treiber bei den anzugreifenden Stationen eingesetzt werden. Die hohe Entdeckungswahrscheinlichkeit durch die nicht standardkonformen Nachrichten ist ein weiterer Nachteil dieser Angriffe. Im Gegensatz dazu sind der Channel-Switch-Angriff als auch der Quiet-Angriff relativ leicht umzusetzen und sind sowohl durch die minimale Anzahl benötigter Nachrichten, als auch durch die Standardkonformität der Nachrichten sehr schwer zu entdecken. Obwohl sich diese beiden Angriffe gegen Geräte richten, die eine Modulation nach 802.11a oder 802.11n unterstützen, waren in den Tests interessanterweise auch reine 802.11b/g-Geräte angreifbar. Diese Tatsachen erhöhen die Attraktivität der Ansätze für potentielle Angreifer und gleichzeitig auch die damit verbundene Gefahr für die Benutzer eines WLANs. Da bei den Tests insbesondere Geräte neuerer Generationen erfolgreich angegriffen werden konnten, stellen die Angriffe mit deren steigender Verbreitung ein sich erhöhendes Risiko dar. Solange in der Praxis eine Verschlüsselung von Management-Nachrichten noch nicht existiert, sind diese Angriffe problemlos durchführbar und somit eine große Gefahr für die Verfügbarkeit eines WLANs.

Da die Erweiterung 802.11w noch nicht durch das IEEE verabschiedet wurde, ist derzeit noch kein einheitliches Verfahren für die Verschlüsselung von Management-Nachrichten verfügbar. Somit kann das Fälschen dieser Nachrichten momentan nicht verhindert werden. Eine theoretische Möglichkeit, um zumindest gefälschte Nachrichten erkennen zu können, ist der Vergleich der Signalstärken, mit denen die verschiedenen Nachrichten empfangen wurden. Dieser Ansatz wird in der Arbeit von Sheng et al. [105] diskutiert und ermöglicht beispielsweise das Erkennen gefälschter Beacons oder Deauthentication-Nachrichten eines Access Points. Die Zuverlässigkeit der Erkennung ist dabei allerdings sehr begrenzt und kann durch einen Angreifer durch das Anpassen seiner Sendeleistung ausgehebelt werden. Soweit bekannt, sind derzeit noch keine Systeme verfügbar, die eine zuverlässige Erkennung in der Praxis ermöglichen. Daher ist es umso wichtiger, dass die Gefahren von Angriffen gegen die Verfügbarkeit erkannt und hierfür zukünftig praktikable Sicherheitsmechanismen gefunden werden.

Neben den im Zentrum stehenden Angriffen gegen die Verfügbarkeit eines WLANs hat diese Arbeit des Weiteren Probleme hinsichtlich der Standardkonformität der getesteten Geräte aufgedeckt. Die Analyse des Deauthentication-Angriffs hat gezeigt, dass manche Deauthentication-Nachrichten von verschiedenen Geräten ignoriert wurden. Ebenso wurden *Channel Switch Announcements* sowie *Quiet Elements* von einigen Geräten ignoriert oder zumindest nicht standardkonform ausgewertet. Diese Nachrichten sind aber für die Durchführung der DFS-Mechanismen im 5-GHz-Bereich und somit für die Vermeidung von Störeinflüssen auf militärische Radarsignale im europäischem Raum unerlässlich. Da die Tests auf Kanal 60 (5,3 GHz) und mit Geräten durchgeführt wurden, die für den europäischen Raum zugelassen sind, ist das Nichtbeachten dieser Nachrichten umso erstaunlicher. Das konkrete Verhalten war stets von dem verwendeten Treiber und teilweise auch von dem verwendeten *Access Point* abhängig. Insbesondere das Nichtbeachten des *Quiet Elements* bei den drei NICs Intel 5100AGN, Ubiquiti SRC und Airport Extreme in Verbindung mit jedem getesteten Treiber ist zum einen nicht standardkonform und zum anderen ein Verstoß gegen die Vorschriften des ETSI nach EN 301 893 [38]. Diese Geräte dürften zumindest in Verbindung mit den untersuchten Treibern für Linux, Windows XP und Mac OS X nicht in Europa verwendet werden.

Aufgrund der steigenden Anforderungen an heutige WLANs, wie hohe Bandbreite, Reichweite oder die Unterstützung von QoS-Diensten, nimmt auch die Komplexität des IEEE 802.11-Standards stetig zu. Dies führt unvermeidlich zu einer vermehrten Anzahl an potentiellen Angriffspunkten, wie die Angriffe gegen das *Block Acknowledgement* von 802.11n oder das Ausnutzen der AIFSs von 802.11e verdeutlichen. Selbst die Protokolle zur Sicherung der Vertraulichkeit und Integrität bieten Angriffsmöglichkeiten, wie beispielsweise die TKIP-Gegenmaßnahmen, durch die die Verfügbarkeit eines WLANs kompromittiert werden kann. Die Sicherung der Verfügbarkeit stellt daher eine große Herausforderung dar, die es in Zukunft insbesondere mit dem Aufkommen neuer Einsatzgebiete von WLANs zu bewältigen gilt und somit bei der Weiterentwicklung des Standards verstärkt berücksichtigt werden sollte. Die momentane Situation im Standardisierungsprozess des IEEEs lässt allerdings eine andere Entwicklung erwarten. Durch den Druck vieler Industriepartner, die möglichst schnell neue Erweiterungen auf den Markt bringen möchten und daher schon unvollständige Draft-Versionen in ihren Geräten implementieren, werden Erweiterungen teilweise frühzeitig verabschiedet. Am Beispiel von 802.11n zeigt sich, dass bekannte Angriffsmöglichkeiten gegen die Verfügbarkeit, wie die gegen das *Block Acknowledgement*, teilweise hingenommen werden, um den Standardisierungsprozess nicht noch weiter hinauszuzögern [96]. Allerdings stellt der Einsatz von WLANs ohne die Gewährleistung der Verfügbarkeit, insbesondere in kritischen Anwendungsgebieten, ein hohes Risiko dar. Aber auch in weniger kritischen Anwendungsbereichen spielt die Verfügbarkeit eine immer wichtigere Rolle, die gewährleistet werden muss, um die Akzeptanz von WLANs auf Dauer zu sichern. Niemand wird sich ein neues WLAN-fähiges Telefon zulegen, wenn jemand in der Nachbarschaft mit einem handelsüblichen Laptop und wenig Aufwand den Begriff des Telefonstreichs neu definieren kann.

Literaturverzeichnis

[1] Aad, I. ; Hubaux, J. P. ; Knightly, E. W.: Denial of Service Resilience in Ad Hoc Networks. In: *Proceedings of the 10th annual international conference on Mobile computing and networking*, ACM New York, NY, USA, 2004, S. 202–215

[2] ABI Research: *Wi-Fi Hotspot Forecasts*. Marktforschungsstudie, 2008. `http://www.abiresearch.com/products/market_data/Wi-Fi_Hotspot_Forecasts` Code: MD-WLHS

[3] Aboba, B. ; Blunk, L. ; Vollbrecht, J. ; Carlson, J. ; Levkowetz, H.: *RFC 3748: Extensible Authentication Protocol*. IETF, 2004. `http://www.ietf.org/rfc/rfc3748.txt`

[4] Aboba, B. ; Simon, D.: *RFC 2716: PPP EAP TLS Authentication Protocol*. IETF, 1999. `http://www.ietf.org/rfc/rfc2716.txt`

[5] Acharya, Mithun ; Sharma, Tanu ; Thuente, David ; Sizemore, David: Intelligent Jamming in 802.11b Wireless Networks. In: *Proceedings of OPNETWORK*. Washington D.C., USA : OPNET, 2004

[6] Acharya, Mithun ; Thuente, David: Intelligent Jamming Attacks, Counterattacks and (Counter)^2 Attacks in 802.11b Wireless Networks. In: *Proceedings of OPNETWORK*. Washington D.C., USA : OPNET, 2005

[7] Ahmad, Sohail ; Murthy, J V R. ; Vartak, Amit: *Autoimmunity Disorder in Wireless LANs*. DEFCON, 2008. `http://www.defcon.org/images/defcon-16/dc16-presentations/defcon-16-ahmad.pdf`

[8] Arbaugh, W.A.: *An Inductive Chosen Plaintext Attack against WEP/WEP2*. IEEE, 2001. `http://www.cs.umd.edu/~waa/attack/v3dcmnt.htm`

[9] Arbaugh, W.A. ; Shankar, N. ; Wan, Y.C.J. ; Zhang, Kan: Your 802.11 Wireless Network has No Clothes. In: *Wireless Communications, IEEE* 9 (2002), Nr. 6, S. 44–51

[10] Barker, R. H.: Group Synchronizing of Binary Digital Sequences. In: *Communication Theory*. London, Butterworth, 1953, S. 273–287

[11] Bayraktaroglu, E. ; King, C. ; Liu, X. ; Noubir, G. ; Rajaraman, R. ; Thapa, B.: On the Performance of IEEE 802.11 under Jamming. In: *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, 2008. – ISBN 0743–166X, S. 1265–1273

[12] Beck, Martin ; Tews, Erik: *Practical attacks against WEP and WPA*. Whitepaper, November 2008. `http://dl.aircrack-ng.org/breakingwepandwpa.pdf`

[13] Bellardo, John ; Savage, Stefan: 802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions. In: *Proceedings of the 12th conference on USENIX Security Symposium* Bd. 12, USENIX Association, 2003, 15–28

[14] Bharghavan, Vaduvur ; Demers, Alan ; Shenker, Scott ; Zhang, Lixia: MACAW: a media access protocol for wireless LAN's. In: *Proceedings of the conference on Communications architectures, protocols and applications*. London, United Kingdom : ACM, 1994. – ISBN 0–89791–682–4, 212–225

[15] Bianchi, G. ; Stefano, A. D. ; Giaconia, C. ; Scalia, L. ; Terrazzino, G. ; Tinnirello, I.: Experimental Assessment of the Backoff Behavior of Commercial IEEE 802.11b Network Cards. In: *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, 2007. – ISBN 0743–166X, S. 1181–1189

[16] Bishop, Matt: *Introduction to Computer Security.* 2005. Addison-Wesley Longman, Amsterdam, 2004. – 784 S. – ISBN 0321247442

[17] Blum, Manuel: Coin flipping by telephone a protocol for solving impossible problems. In: *SIGACT News* 15 (1983), Nr. 1, S. 23–27

[18] Borisov, Nikita ; Goldberg, Ian ; Wagner, David: Intercepting mobile communications: the insecurity of 802.11. In: *Proceedings of the 7th annual international conference on Mobile computing and networking.* Rome, Italy : ACM, 2001. – ISBN 1–58113–422–3, 180–189

[19] Borre, K. ; Akos, D. M. ; Bertelsen, N. ; Rinder, P. ; Jensen, S. H.: *A Software-defined GPS and Galileo Receiver: A Single-frequency Approach.* Birkhäuser, 2007

[20] Bratus, Sergey ; Cornelius, Cory ; Kotz, David ; Peebles, Daniel: Active Behavioral Fingerprinting of Wireless Devices. In: *WiSec '08: Proceedings of the first ACM conference on Wireless network security.* NY, USA : ACM, 2008. – ISBN 978–1–59593–814–5, S. 56–61

[21] Buennemeyer, T.K. ; Gora, M. ; Marchany, R.C. ; Tront, J.G.: Battery Exhaustion Attack Detection with Small Handheld Mobile Computers. In: *IEEE International Conference on Portable Information Devices*, 2007, S. 1–5

[22] Butti, Laurent: *CVE-2006-6332: Stack-based buffer overflow in MadWifi before 0.9.2.1.* CVE, 2006. `http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-6332`

[23] Butti, Laurent: *WVE-2007-0013: D-Link DWL-G650+ Wireless Driver Long TIM Overflow.* WVE, 2007. `http://www.wirelessve.org/entries/show/WVE-2007-0013`

[24] Butti, Laurent: *WVE-2008-0008: Atheros IE Tag Overflow.* WVE, 2008. `http://www.wve.org/entries/show/WVE-2008-0008`

[25] Butti, Laurent ; LMH: *MOKB-22-11-2006: NetGear WG311v1 Wireless Driver Long SSID Overflow.* MOKB, 2006. `http://projects.info-pull.com/mokb/MOKB-22-11-2006.html`

[26] Butti, Laurent ; Moore, H.D. ; LMH: *MOKB-18-11-2006: NetGear MA521 Wireless Driver Long Rates Overflow.* MOKB, 2006. `http://projects.info-pull.com/mokb/MOKB-18-11-2006.html`

[27] Butti, Laurent ; Tinnés, Julien: *WVE-2008-0010: Marvell Null SSID Association Request.* WVE, 2008. `http://www.wve.org/entries/show/WVE-2008-0010`

[28] Butti, Laurent ; Tinnés, Julien: Discovering and Exploiting 802.11 Wireless Driver Vulnerabilities. In: *Journal in Computer Virology* 4 (2008), Nr. 1, S. 25–37

[29] Cache, Johnny ; Eagle, Chris: *WVE-2006-0071: Broadcom Driver Probe Response SSID Overflow.* WVE, 2006. `http://www.wirelessve.org/entries/show/WVE-2006-0071`

[30] Cache, Johnny ; Moore, H.D. ; LMH ; Miller, Matt: *WVE-2006-0072: D-Link DWL-G132 Wireless Driver Beacon Rates Overflow.* WVE, 2006. `http://www.wirelessve.org/entries/show/WVE-2006-0072`

[31] Cam-Winget, Nancy ; Housley, Russ ; Wagner, David ; Walker, Jesse: Security Flaws in 802.11 Data Link Protocols. In: *Communications of the ACM* 46 (2003), Nr. 5, S. 35–39

[32] Cam-Winget, Nancy ; Smith, Doug ; Walker, Jesse: *IEEE 802.11-07/2163r0 – A-MPDU Security Issues.* IEEE, 2007. `https://mentor.ieee.org/802.11/file/07/11-07-2163-01-000n-a-mpdu-security-issues.ppt`

[33] Cárdenas, Alvaro A. ; Radosavac, Svetlana ; Baras, John S.: Detection and Prevention of MAC Layer Misbehavior in Ad Hoc Networks. In: *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*. Washington DC, USA : ACM, 2004. – ISBN 1–58113–972–1, 17–22

[34] Chen, B. ; Muthukkumarasamy, V. ; Guimaraes, N. ; Isaias, P. ; Goikoetxea, A.: Denial of Service Attacks Against 802.11 DCF. In: *Proceedings of the IADIS International Conference: Applied Computing*, 2006

[35] Chou, Andy ; Yang, Junfeng ; Chelf, Benjamin ; Hallem, Seth ; Engler, Dawson: An empirical study of operating systems errors. In: *Proceedings of the eighteenth ACM symposium on Operating systems principles*. Banff, Alberta, Canada : ACM, 2001. – ISBN 1–58113–389–8, 73–88

[36] Clausen, T. ; Jacquet, P.: *RFC 3626: Optimized Link State Routing Protocol (OLSR)*. IETF, 2003. `http://www.ietf.org/rfc/rfc3626.txt`

[37] Ellch, Jon ; Pinto, Breno S.: *WVE-2007-0001: Intel Centrino Wireless Driver Malformed Beacon SSID IE*. WVE, 2007. `http://www.wirelessve.org/entries/show/WVE-2007-0001`

[38] ETSI: *EN 301 893 v1.5.1: Broadband Radio Access Networks (BRAN); 5 GHz High Performance RLAN; Harmonized EN covering essential requirements of article*. 2008

[39] F.A.Z.-Institut ; Intel: *E-Health – Aktuelle Entscheiderbefragung zur IT in Krankenhäusern: Anforderungen, Potenziale, Investitionen*. 2006 (Best of IT-Solutions). – 58 S.

[40] Ferguson, Niels: *IEEE 802.11-02/020r0 – Michael: An improved MIC for 802.11 WEP*. IEEE, 2002. `https://mentor.ieee.org/802.11/file/02/11-02-0020-00-000i-michael-an-improved-mic-for-802-11-wep.doc`

[41] Ferguson, Niels ; Reinhold, Arnold: *DoS Attack on WPA 802.11*. Mail-Archive, 2002. `http://www.mail-archive.com/cryptography@wasabisystems.com/msg03078.html`

[42] Ferreri, F. ; Bernaschi, M. ; Valcamonici, L.: Access Points Vulnerabilities to DoS Attacks in 802.11 Networks. In: *IEEE Wireless Communications and Networking Conference (WCNC)* Bd. 1, 2004. – ISBN 1525–3511, S. 634–638 Vol.1

[43] Fluhrer, Scott R. ; Mantin, Itsik ; Shamir, Adi: Weaknesses in the Key Scheduling Algorithm of RC4. In: *Revised Papers from the 8th Annual International Workshop on Selected Areas in Cryptography*, Springer-Verlag, 2001. – ISBN 3–540–43066–0, 1–24

[44] Franklin, Jason ; McCoy, Damon ; Tabriz, Parisa ; Neagoe, Vicentiu ; Randwyk, Jamie V. ; Sicker, Douglas: Passive Data Link Layer 802.11 Wireless Device Driver Fingerprinting. In: *Proceedings of the 15th conference on USENIX Security Symposium - Volume 15*. Vancouver, B.C., Canada : USENIX Association, 2006, 12–12

[45] Gast, Matthew: *802.11 Wireless Networks: The Definitive Guide*. 1. O'Reilly, 2002. – ISBN 0596001835

[46] Glass, Steve ; Muthukkumarasamy, Vallipuram: 802.11 DCF Denial of Service Vulnerabilities. In: *Australian Computer, Network & Information Forensics Conference*, School of Computer and Information Science, Edith Cowan University, Western Australia, 2005. – ISBN 0–7298–0612–X, S. 8–14

[47] Glass, Steve ; Muthukkumarasamy, Vallipuram: A Study of the TKIP Cryptographic DoS Attack. In: *15th IEEE International Conference on Networks (ICON)*, 2007. – ISBN 1556–6463, S. 59–65

[48] Guang, L. ; Assi, C.: Mitigating Smart Selfish MAC Layer Misbehavior in Ad Hoc Networks. In: *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2006, S. 116–123

[49] Guang, L. ; Assi, C.: Vulnerability Assessment of Ad Hoc Networks to MAC Layer Misbehavior. In: *Wireless Communications and Mobile Computing* 7 (2007), Nr. 6, S. 703–715

[50] Guang, L. ; Assi, C. ; Benslimane, A.: Enhancing IEEE 802.11 Random Backoff in Selfish Environments. In: *IEEE Transactions on Vehicular Technology* 57 (2008), Nr. 3, S. 1806–1822. – ISSN 0018–9545

[51] Gummadi, Ramakrishna ; Wetherall, David ; Greenstein, Ben ; Seshan, Srinivasan: Understanding and Mitigating the Impact of RF Interference on 802.11 networks. In: *SIGCOMM Comput. Commun. Rev.* 37 (2007), Nr. 4, S. 385–396

[52] Gupta, V. ; Krishnamurthy, S. ; Faloutsos, M.: Denial of Service Attacks at the MAC Layer in Wireless Ad Hoc Networks. In: *MILCOM* Bd. 2, 2002, S. 1118–1123

[53] He, Changhua ; Mitchell, John C.: Security Analysis and Improvements for IEEE 802.11i. In: *12th Annual Network and Distributed System Security Symposium*, 2005, S. 90–110

[54] Hu, Yih-Chun ; Perrig, Adrian ; Johnson, David B.: Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. In: *Wireless Networks* 11 (2005), Nr. 1-2, S. 21–38

[55] IEEE: *Std 802.11-1997 – IEEE Standard for LAN/MAN – Specific requirements Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications.* 1997

[56] IEEE: *Std 802.11h™ – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications – Amendment 5: Spectrum and Transmit Power Management Extensions in the 5 GHz band in Europe.* 2003

[57] IEEE: *Std 802.11i™ – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications – Amendment 6: Medium Access Control (MAC) Security Enhancements.* 2004

[58] IEEE: *Std 802.1X™-2004 – IEEE Standard for LAN/MAN – Port-Based Network Access Control.* 2004

[59] IEEE: *Std 802.11e™ – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications – Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements.* 2005

[60] IEEE: *Std 802.11™-2007 – IEEE Standard for LAN/MAN – Specific requirements Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications.* 2007

[61] IEEE: *P802.11n™/D5.0 - Draft Amendment to STANDARD for Information Technology-Telecommunications and Information Exchange Between Systems — Local and Metropolitan networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY). Amendment 4: Enhancements for Higher Throughput.* 2008

[62] IEEE: *P802.11p™ /D4.0 - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Amendment 8: Wireless Access in Vehicular Environments.* 2008

[63] IEEE: *P802.11s/D2.0 - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Amendment x: Mesh Networking.* 2008

[64] IEEE: *P802.11w™/D6.0 - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Amendment 4: Protected Management Frames.* 2008

[65] Jain, R. ; Chiu, D. ; Hawe, W.: A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems / Digital Equipment Corporation. Version: 1984. `ftp://ftp.netlab.ohio-state.edu/pub/jain/papers/fairness.htm`. Maynard, MA, USA, 1984 (TR-301). – DEC Research Report. – 38 S.

[66] Jakubiak, J. ; Koucheryavy, Y.: State of the Art and Research Challenges for VANETs. In: *5th IEEE Consumer Communications and Networking Conference (CCNC)*, 2008. – ISBN 0197–2618, S. 912–916

[67] Jiang, Daniel ; Delgrossi, Luca: IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments. In: *IEEE Vehicular Technology Conference (VTC)*, 2008. – ISBN 1550–2252, S. 2036–2040

[68] Jiang, Li B. ; Liew, Soung C.: Proportional fairness in wireless LANs and ad hoc networks. In: *IEEE Wireless Communications and Networking Conference* Bd. 3, 2005. – ISBN 1525–3511, S. 1551–1556 Vol. 3

[69] Jones, K. ; Liu, Ling: What Where Wi: An Analysis of Millions of Wi-Fi Access Points. In: *IEEE International Conference on Portable Information Devices*, 2007, S. 1–4

[70] Jow, A. ; Schurgers, C. ; Palmer, D.: CalRadio: a portable, flexible 802.11 wireless research platform. In: *Proceedings of the 1st international workshop on System evaluation for mobile platforms*. San Juan, Puerto Rico : ACM, 2007, S. 49–54

[71] Kannhavong, B. ; Nakayama, H. ; Nemoto, Y. ; Kato, N. ; Jamalipour, A.: A survey of routing attacks in mobile ad hoc networks. In: *Wireless Communications, IEEE* 14 (2007), Nr. 5, S. 85–91. – ISSN 1536–1284

[72] Kargl, Frank ; Klenk, Andreas ; Schlott, Stefan ; Weber, Michael: Advanced Detection of Selfish or Malicious Nodes in Ad Hoc Networks. In: *Security in Ad-hoc and Sensor Networks (ESAS)* Bd. 3313, Springer, 2004. – ISBN 3–540–24396–8, 152–165

[73] Khan, M.A. ; Hasan, A.: Pseudo Random Number Based authentication to counter denial of service attacks on 802.11. In: *Wireless and Optical Communications Networks, 2008. WOCN '08. 5th IFIP International Conference on*, 2008, S. 1–5

[74] Kyasanur, P. ; Vaidya, N.H.: Selfish MAC Layer Misbehavior in Wireless Networks. In: *Mobile Computing, IEEE Transactions on* 4 (2005), Nr. 5, S. 502–516. – ISSN 1536–1233

[75] In: Li, Hongjian ; Xu, Ming ; Li, Yi: *Lecture Notes in Computer Science*. Bd. 4847: *Selfish MAC Layer Misbehavior Detection Model for the IEEE 802.11-Based Wireless Mesh Networks*. Guangzhou, China : Springer, 2007, S. 382–391

[76] Lichte, H. S. ; Valentin, S. ; Eitzen, F. ; Stege, M. ; Unger, C. ; Karl, H.: Integrating Multiuser dynamic OFDMA into IEEE 802.11a and Prototyping it on a Real-Time Software-Defined Radio Testbed. In: *3rd International Conferenceon Testbeds and Research Infrastructure for the Development of Networks and Communities*, 2007, S. 1–9

[77] LittleW0lf: *WVE-2006-0050: IEEE 802.11 invalid channel beacon DoS*. WVE, 2006. `http://www.wirelessve.org/entries/show/WVE-2006-0050`

[78] Lolla, V.N. ; Law, Lap K. ; Krishnamurthy, S.V. ; Ravishankar, C. ; Manjunath, D.: Detecting MAC Layer Back-off Timer Violations in Mobile Ad Hoc Networks. In: *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2006. – ISBN 1063–6927, S. 63

[79] Lu, M. ; Steenkiste, P. ; Chen, T.: Using Commodity Hardware Platform to Develop and Evaluate CSMA Protocols. In: *International Workshop on Wireless Network Testbeds, experimental Evaluation and Characterization*. San Francisco, USA : ACM, 2008, 73–80

[80] MacMichael, John L.: Auditing Wi-Fi Protected Access (WPA) Pre-Shared Key Mode. In: *Linux Journal* 2005 (2005), Nr. 137, S. 2. – ISSN 1075–3583

[81] Marshall, A. W. ; Olkin, I.: *Inequalities: theory of majorization and its applications.* Academic Press, 1979. – ISBN 978–0124737501

[82] Marti, Sergio ; Giuli, T. J. ; Lai, Kevin ; Baker, Mary: Mitigating routing misbehavior in mobile ad hoc networks. In: *Proceedings of the 6th annual international conference on Mobile computing and networking.* Boston, Massachusetts, United States : ACM, 2000. – ISBN 1–58113–197–6, 255–265

[83] Martin, T. ; Hsiao, M. ; Ha, Dong ; Krishnaswami, J.: Denial-of-service attacks on battery-powered mobile computers. In: *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications (PerCom)*, 2004, S. 309–318

[84] Minden, G. J. ; Evans, J. B. ; Searl, L. ; DePardo, D. ; Petty, V. R. ; Rajbanshi, R. ; Newman, T. ; Chen, Q. ; Weidling, F. ; Guffey, J.: KUAR: A Flexible Software-Defined Radio Development Platform. In: *Proceedings of the 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, 2007, S. 428–439

[85] Moore, H.D.: *MOKB-01-11-2006: Apple Airport Probe Response Kernel Memory Corruption.* MOKB, 2006. `http://projects.info-pull.com/mokb/MOKB-01-11-2006.html`

[86] Moore, H.D. ; LMH: *MOKB-16-11-2006: NetGear WG111v2 Wireless Driver Long Beacon Overflow.* MOKB, 2006. `http://projects.info-pull.com/mokb/MOKB-16-11-2006.html`

[87] Negi, R. ; Rajeswaran, A.: DoS analysis of reservation based MAC protocols. In: *IEEE International Conference on Communications (ICC)* Bd. 5, 2005, S. 3632–3636 Vol. 5

[88] Neufeld, Michael ; Fifield, Jeff ; Doerr, Christian ; Sheth, Anmol ; Grunwald, Dirk: SoftMAC - Flexible Wireless Research Platform. In: *Fourth Workshop on Hot Topics in Networks (HotNets-IV)*, 2005

[89] Noubir, Guevara: On Connectivity in Ad Hoc Networks under Jamming Using Directional Antennas and Mobility. In: *Wired/Wireless Internet Communications (WWIC)* Bd. 2957, Springer, 2004 (Lecture Notes in Computer Science). – ISBN 3–540–20954–9, 186–200

[90] Papadimitratos, P. ; Kung, A. ; Hubaux, J. P. ; Kargl, F.: Privacy and Identity Management for Vehicular Communication Systems: a Position Paper. In: *Workshop on Standards for Privacy in User-Centric Identity Management* (2006)

[91] Paul, T.K. ; Ogunfunmi, T.: Wireless LAN Comes of Age: Understanding the IEEE 802.11n Amendment. In: *Circuits and Systems Magazine, IEEE* 8 (2008), Nr. 1, S. 28–54. – ISSN 1531–636X

[92] Pelechrinis, Konstantinos ; Iliofotou, Marios: *Denial of Service Attacks in Wireless Networks: The case of Jammers.* Department of Computer Science & Engineering, University of California Riverside, 2006. `http://www.cs.ucr.edu/~kpele/Jamming.pdf`

[93] Perahia, E.: IEEE 802.11n Development: History, Process, and Technology. In: *Communications Magazine, IEEE* 46 (2008), Nr. 7, S. 48–55. – ISSN 0163–6804

[94] Perkins, C ; Belding-Royer, E. ; Das, S.: *RFC 3561: Ad hoc On-Demand Distance Vector (AODV) Routing.* IETF, 2003. `http://www.ietf.org/rfc/rfc3561.txt`

[95] Qian, Luke ; Cam-Winget, Nancy ; Smith, Doug: *IEEE 802.11-08/0703r0 – Issues and Solutions to IEEE 802.11n.* IEEE, 2008. `https://mentor.ieee.org/802.11/file/08/11-08-0703-00-000n-11n-a-mpdu-dos-issues-and-solutions.ppt`

[96] Qian, Luke ; Cam-Winget, Nancy ; Smith, Doug: *IEEE 802.11-08/0755r1 – Review of 802.11n A-MPDU DoS Issues*. IEEE, 2008. https://mentor.ieee.org/802.11/file/08/11-08-0755-01-000n-review-of-a-mpdu-dos-issues.ppt

[97] Racic, Radmilo ; Ma, Denys ; Chen, Hao: Exploiting MMS Vulnerabilities to Stealthily Exhaust Mobile Phone's Battery. In: *Securecomm and Workshops*, 2006. – ISBN 1–4244–0423–1, S. 1–10

[98] Raya, M. ; Aad, I. ; Hubaux, J.-P. ; Fawal, A. E.: DOMINO: Detecting MAC Layer Greedy Behavior in IEEE 802.11 Hotspots. In: *IEEE Transactions on Mobile Computing* 5 (2006), Nr. 12, S. 1691–1705. – ISSN 1536–1233

[99] Raymond, D. ; Marchany, R. ; Brownfield, M. ; Midkiff, S.: Effects of Denial of Sleep Attacks on Wireless Sensor Network MAC Protocols. In: *Information Assurance Workshop, IEEE*, 2006, S. 297–304

[100] Rigney, C. ; Willens, S. ; Rubens, A. ; Simpson, W.: *RFC 2865: Remote Authentication Dial In User Service (RADIUS)*. IETF, 2000. http://www.ietf.org/rfc/rfc2865.txt

[101] Schleher, D. Curtis: *Electronic Warfare in the Information Age*. Bk & diskette. Artech House, 1999. – 614 S. – ISBN 0890065268

[102] Schoch, Elmar ; Kargl, Frank ; Leinmuller, Tim ; Weber, Michael: Vulnerabilities of Geocast Message Distribution. In: *Globecom Workshops, IEEE*, 2007. – ISBN 978–1–4244–2024–7, S. 1–8

[103] Sharma, Ashish ; Belding, Elizabeth M.: FreeMAC: Framework for Multi-Channel Mac Development on 802.11 Hardware. In: *Proceedings of the ACM workshop on Programmable routers for extensible services of tomorrow*. Seattle, WA, USA : ACM, 2008. – ISBN 978–1–60558–181–1, 69–74

[104] Sharma, Ashish ; Tiwari, Mohit ; Zheng, Haitao: MadMAC: Building a Reconfiguration Radio Testbed using Commodity 802.11 Hardware. In: *1st IEEE Workshop on Networking Technologies for Software Defined Radio Networks*, 2006, S. 78–83

[105] Sheng, Yong ; Tan, Keren ; Chen, Guanling ; Kotz, David ; Campbell, Andrew: Detecting 802.11 MAC Layer Spoofing Using Received Signal Strength. In: *The 27th Conference on Computer Communications INFOCOM*, 2008. – ISBN 0743–166X, S. 1768–1776

[106] Shono, T. ; Shirato, Y. ; Shiba, H. ; Uehara, K. ; Araki, K. ; Umehira, M.: IEEE 802.11 wireless LAN implemented on software defined radio with hybrid programmable architecture. In: *IEEE Transactions on Wireless Communications* 4 (2005), Nr. 5, S. 2299–2308. – ISSN 1536–1276

[107] Sikora, Axel: *Wireless LAN - Protokolle und Anwendungen*. 1. Aufl. Addison-Wesley, 2001. – 224 S. – ISBN 382731917X

[108] Smith, Doug ; Walker, Jesse ; Cam-Winget, Nancy: *WVE-2008-0006: Block ACK DoS*. WVE, 2008. http://www.wirelessve.org/entries/show/WVE-2008-0006

[109] Smith, Jason: Denial of Service Vulnerabilities in IEEE 802.11i. In: *Recent advances in security technology*, 2007. – ISBN 978–0–9757873–0–7, S. 212–213

[110] Stajano, Frank ; Anderson, Ross J.: The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In: *Proceedings of the 7th International Workshop on Security Protocols*, Springer, 2000. – ISBN 3–540–67381–4, 172–194

[111] Stallings, William: The Advanced Encryption Standard. In: *Cryptologia* XXVI (2002), Nr. 3, S. 165–188. – ISSN 0161–1194

[112] Stubblefield, Adam ; Ioannidis, John ; Rubin, Aviel D.: Using the Fluhrer, Mantin, and Shamir Attack to Break WEP. (2002). ISBN 1–891562–14–2

[113] Ståhlberg, Mika: *Radio Jamming Attacks Against Two Popular Mobile Networks*. Helsinki University of Technology, 2000. `http://www.tml.tkk.fi/Opinnot/Tik-110.501/2000/papers/stahlberg.pdf`

[114] Swift, Michael M. ; Bershad, Brian N. ; Levy, Henry M.: Improving the reliability of commodity operating systems. In: *Proceedings of the nineteenth ACM symposium on Operating systems principles*. Bolton Landing, NY, USA : ACM, 2003. – ISBN 1–58113–757–5, 207–222

[115] Taher, T.M. ; Misurac, M.J. ; LoCicero, J.L. ; Ucci, D.R.: Microwave Oven Signal Interference Mitigation For Wi-Fi Communication Systems. In: *5th IEEE Consumer Communications and Networking Conference (CCNC)*, 2008. – ISBN 0197–2618, S. 67–68

[116] Thuente, David J. ; Acharya, Mithun: Intelligent Jamming in Wireless Networks with Applications to 802.11b and Other Networks. In: *Proceedings of the 25th IEEE Communications Society Military Communications Conference (MILCOM)*, 2006

[117] Thuente, David J. ; Newlin, Benjamin ; Acharya, Mithun: Jamming Vulnerabilities of IEEE 802.11e. In: *Proceedings of the 26th IEEE Communications Society Military Communications Conference (MILCOM)*, 2007, S. 1–7

[118] Walker, Jesse R.: *IEEE 802.11-00/362 – Wireless LANs Unsafe at any key size; An analysis of the WEP encapsulation*. IEEE, 2000, . – https://mentor.ieee.org/802.11/file/00/11-00-0362-00-000e-unsafe-at-any-key-size-an-analysis-of-the-wep-encapsulation.doc

[119] Wätjen, Dietmar: *Kryptographie. Grundlagen, Algorithmen, Protokolle.* 1. Spektrum Akademischer Verlag, 2003. – 306 S. – ISBN 3827414318

[120] Weiser, Mark: Some Computer Science Issues in Ubiquitous Computing. In: *Communications ACM* 36 (1993), Nr. 7, S. 75–84. – ISSN 0001–0782

[121] West, William ; Agu, Emmanuel: Experimental Evaluation of Energy-Based Denial-of Service Attacks in Wireless Networks. In: *International Journal of Computer Science and Network Security* Bd. 7, 2007 (6), S. 222–236

[122] Whiting, D. ; Housley, R. ; Ferguson, N.: *RFC 3610: Counter with CBC-MAC (CCM)*. IETF, 2003. `http://www.ietf.org/rfc/rfc3610.txt`

[123] Wool, Avishai: A Note on the Fragility of the Michael Message Integrity Code. In: *IEEE Transactions on Wireless Communications* 3 (2004), Nr. 5, S. 1459–1462. – ISSN 1536–1276

[124] Wright, Joshua: High Speed Risks in 802.11n Networks. In: *RSA Conference*, ARUBA Networks, 2008

[125] Wullems, Chris ; Tham, Kevin ; Smith, Jason ; Looi, Mark: A Trivial Denial of Service Attack on IEEE 802.11 Direct Sequence Spread Spectrum Wireless LANs. In: *Wireless Telecommunications Symposium*, 2004, S. 129–136

[126] Xu, Kaixin ; Gerla, Mario ; Bae, Sang: How effective is the IEEE 802.11 RTS/CTS Handshake in Ad Hoc Networks. In: *IEEE Global Telecommunications Conference (GLOBECOM)* Bd. 1, 2002, S. 72–76 vol.1

[127] Xu, Wenyuan ; Trappe, Wade ; Zhang, Yanyong ; Wood, Timothy: The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks. In: *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*. Urbana-Champaign, IL, USA : ACM, 2005. – ISBN 1–59593–004–3, 46–57

[128] Yousefi, Saleh ; Fathy, Mahmood ; Benslimane, Abderrahim: Performance of Beacon Safety Message Dissemination in Vehicular Ad hoc NETworks (VANETs). In: *Journal of Zhejiang University - Science A* 8 (2007), November, Nr. 12, S. 1990–2004

[129] Zhou, Bosheng ; Marshall, A. ; Zhou, Wenzhe ; Yang, Kun: A Random Packet Destruction DoS Attack for Wireless Networks. In: *IEEE International Conference on Communications (ICC)*, 2008, S. 1658–1662

[130] Zhou, Yihong ; Wu, Dapeng ; Nettles, Scott M.: Analyzing and Preventing MAC-layer Denial of Service Attacks for Stock 802.11 Systems. In: *Workshop on Broadband Wireless Services and Applications (BROADNETS)*, 2004

[131] Zimmermann, Hubert: OSI Reference Model–The ISO Model of Architecture for Open Systems Interconnection. In: *IEEE Transactions on Communications* 28 (1980), Nr. 4, S. 425–432. – ISSN 0096–2244

Erklärung

Ich erkläre, dass ich die Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den .....

Bastian Könings

Inhalte der beigefügten DVD-ROM:

- Diese Arbeit als PDF-Version
- Quellcode der Angriffsimplementierungen
  `/attacks/attack.py`
- Scapy Version 2.0.0.10
  `/attacks/scapy_2.0.0.10`
- Alle Grafiken aus Kapitel 5
  `/analysis/results/...`
- Capture-Dateien der Angriffe im tcpdump/libpcap-Format
  `/analysis/capture/...`
- Grafiken des Durchsatzes der Ping- und Test-Stationen während der Angriffe
  `/analysis/capture/...`
- Restliche Grafiken aus Kapitel 1 bis 4
  `/graphics/...`