

Universität Stuttgart
Fakultät Informatik,
Elektrotechnik und
Informationstechnik

Diplomarbeit Nr.: 2167

**Entwicklung eines Verfahrens
zur Emulation der
Medienzugriffssteuerung
in Wireless LAN**

Zhenxiang Yang

| | |
|--------------------|-----------------------------|
| Studiengang: | Informatik |
| Prüfer: | Prof. Dr. Kurt Rothermel |
| Betreuer: | Dipl.-Inf. Daniel Herrscher |
| Beginn am: | 01.11.2003 |
| Beendet am: | 14.05.2004 |
| CR-Klassifikation: | C.2.5;D.4.4;I.6.3 |

Kurzfassung

Durch das wachsende Bedürfnis nach mobilen Ad-hoc-Netzwerken (MANETs) wurden mobile drahtlose Netzwerke wie Wireless LAN (Drahtlose Lokale Netzwerke) rasant verbreitet. Die Eigenschaften der drahtlosen Netze unterscheiden sich wesentlich von klassischen kabelgebundenen Netzen. Dafür sind neue Anwendungen und Protokolle zu entwickeln. Für die Leistungsanalyse und Fehlersuche der Anwendungen und Protokolle wird eine leistungsfähige und komfortable Testumgebung benötigt.

Das Network Emulation Testbed (NET) der Abteilung Verteilte Systeme an der Universität Stuttgart ist ein flexibel vernetzter Linux PC-Cluster mit 64 Knoten, in dem beliebige Netztopologien und deren Eigenschaften nachgebildet werden können. Damit bietet das NET eine ideale Analyseumgebung für verteilte Anwendungen und Netzprotokolle. Nun sollen auch Untersuchungen in Wireless-LAN-Umgebungen in NET unterstützt werden. Ziel dieser Diplomarbeit ist deshalb die realistische Nachbildung der MAC-Schicht von IEEE 802.11 (Wireless LAN) in NET.

In dieser Ausarbeitung werden zunächst grundlegende Begriffe der Netzwerktechnik und allgemeine Medienzugriffsverfahren vorgestellt. Danach werden die für IEEE 802.11 spezifischen Zugriffsverfahren detailliert analysiert (z.B. CSMA/CA). Die Probleme für die Emulation, besonders die begrenzte Granularität des Timers, und dementsprechende Vereinfachungen werden erläutert. Das entwickelte Emulationskonzept, die verwendeten Algorithmen, sowie wichtige Implementierungsdetails des entstandenen Emulationwerkzeugs werden ausführlich beschrieben. Performancemessungen runden die Arbeit ab.

Inhaltsverzeichnis

| | |
|--|-----------|
| Kapitel 1 Einführung | 1 |
| 1.1 Motivation | 1 |
| 1.2 Aufgabenstellung | 1 |
| 1.3 Struktur der Arbeit | 2 |
| Kapitel 2 Grundlagen..... | 3 |
| 2.1 Das ISO/OSI Referenzmodell | 3 |
| 2.2 LAN-Standards - IEEE 802.x Familie | 4 |
| 2.3 Wired LAN (Kabelgebundene Lokale Netzwerke)..... | 5 |
| 2.4 Medienzugriffsverfahren in LAN..... | 6 |
| 2.4.1 Kollisionsfreies Verfahren | 7 |
| 2.4.1.1 Polling | 7 |
| 2.4.1.2 TokenRing | 7 |
| 2.4.2 CSMA..... | 7 |
| 2.4.2.1 CSMA - Überblick | 7 |
| 2.4.2.2 CSMA/CD | 7 |
| 2.4.2.3 Binär-Exponential-Back-Off-Funktion für CSMA/CD | 9 |
| 2.5 Wireless LAN (Drahtlose Lokale Netzwerke)..... | 10 |
| 2.5.1 Wireless LAN Standard | 10 |
| 2.5.2 Bitübertragungsschicht in IEEE 802.11 - Überblick..... | 10 |
| 2.5.3 Netzwerk-Topologien..... | 12 |
| 2.5.3.1 Infrastrukturnetzwerk | 13 |
| 2.5.3.2 Ad-hoc-Netzwerk | 14 |
| 2.5.4 Netzwerkverwaltungen..... | 14 |
| 2.5.5 Verschlüsselte Datenübertragung..... | 15 |
| 2.6 Drahtlose- vs. Kabelgebundene LANs..... | 16 |
| 2.7 Das NET (Network Emulation Testbed)..... | 16 |
| 2.7.1 Hardwareumgebung | 16 |
| 2.7.2 Emulationsstrategie | 17 |
| Kapitel 3 Analyse des Medienzugriffsverfahrens in IEEE 802.11..... | 19 |
| 3.1 Elementare Medienzugriffsmethoden | 19 |
| 3.2 CSMA/CA..... | 20 |
| 3.2.1 Basismechanismus | 21 |
| 3.2.2 CSMA/CA mit ACK | 24 |
| 3.2.3 CSMA/CA – mit RTS/CTS | 26 |
| 3.3 PCF..... | 31 |
| 3.4 Fragmentierung | 33 |
| Kapitel 4 Konzeption und Entwurf | 35 |
| 4.1 Architektur | 35 |
| 4.2 Emulation des Funkkanals | 35 |
| 4.2.1 Tunnelung..... | 35 |
| 4.2.2 Reichweite..... | 36 |
| 4.2.3 Signalausbreitungsverzögerung | 37 |
| 4.3 Emulation der MAC-Schicht..... | 37 |
| 4.3.1 Netzwerktopologie, Medienzugriffsverfahren und optionale Funktionen in MAC | 37 |

| | |
|---|-----------|
| 4.3.2 Rahmentypen und Rahmenformat | 38 |
| 4.3.3 BackOff-Funktion | 42 |
| 4.3.4 Trägererkennung | 44 |
| 4.3.5 Granularität des Timers | 45 |
| 4.3.6 Interframe Space (IFS) | 45 |
| 4.4 Entwurf der WILEMU | 47 |
| 4.4.1 Definitionen von WILEMU-Zuständen und Lokalen Variablen | 48 |
| 4.4.2 Zustandsdiagramme | 48 |
| 4.4.2.1 „IDLE“ | 48 |
| 4.4.2.2 „Anschluss 1“ | 52 |
| 4.4.2.3 „CONTENTION“ | 55 |
| 4.4.2.4 „LISTENING“ | 56 |
| 4.4.2.5 „WAITING_CTS“ | 57 |
| 4.4.2.6 „WAITING_ACK“ | 59 |
| 4.4.2.7 „WAITING_DATA“ | 60 |
| 4.4.2.8 „RECEIVING_DATA“ | 61 |
| 4.4.2.9 „BROADCASTING“ | 62 |
| 4.4.2.10 „RECEIVING_BC“ | 63 |
| Kapitel 5 Implementierung | 65 |
| 5.1 Architektur | 65 |
| 5.2 Funktionsweise | 66 |
| 5.2.1 Senden und Empfang | 66 |
| 5.2.2 Modulaktivierung und -deaktivierung | 67 |
| 5.2.3 Steuerung von WILEMU und Änderung von Parametern | 67 |
| 5.3 Algorithmus | 68 |
| Kapitel 6 Testmessung | 71 |
| 6.1 Test-Werkzeug | 71 |
| 6.2 Test-Szenarien und Messergebnisse | 71 |
| 6.2.1 Test 1 | 71 |
| 6.2.2 Test 2 | 72 |
| 6.2.3 Test 3 | 73 |
| 6.3 Diskussion der Testergebnisse | 74 |
| Kapitel 7 Verwandte Arbeiten | 77 |
| 7.1 Zentralisierte Methoden | 77 |
| 7.1.1 NS – E | 77 |
| 7.1.2 JEmu | 77 |
| 7.1.3 Bewertung | 78 |
| 7.2 Verteilte Methode | 78 |
| 7.2.1 EthEmu | 78 |
| Kapitel 8 Zusammenfassung und Ausblick | 81 |
| 8.1 Zusammenfassung | 81 |
| 8.2 Ausblick | 81 |

Abbildungsverzeichnis

| | |
|--|----|
| Abbildung 2-1 ISO/OSI Referenzmodell [tane97]..... | 3 |
| Abbildung 2-2 LAN Standards - IEEE 802.x Familie [IEEE 811]..... | 4 |
| Abbildung 2-3 Transportpfad..... | 5 |
| Abbildung 2-4 Typische Netzwerktopologien von LAN [tane97]..... | 5 |
| Abbildung 2-5 Wichtige Bestandteile des MAC-Kopfs in Ethernet..... | 6 |
| Abbildung 2-6 Medienzugriffsverfahren [roth01] | 6 |
| Abbildung 2-7 Polling-Verfahren..... | 7 |
| Abbildung 2-8 Grundprinzip des CSMA/CD..... | 8 |
| Abbildung 2-9 Minimale Rahmensgröße in CSMA/CD..... | 9 |
| Abbildung 2-10 Frequenzbänder im elektromagnetischen Spektrum [nemg01] | 11 |
| Abbildung 2-11 PHY-Architektur in IEEE 802.11 [IEEE811]..... | 11 |
| Abbildung 2-12 Europäische Kanalselektion- nicht überlappend [IEEE811b] | 12 |
| Abbildung 2-13 Europäische Kanalselektion – überlappend [IEEE811b]..... | 12 |
| Abbildung 2-14 Übertragungsraten und Reichweiten in IEEE 802.11b [nemg01] | 12 |
| Abbildung 2-15 Infrastrukturnetzwerk mit Ethernet..... | 13 |
| Abbildung 2-16 Infrastrukturnetzwerk mit Richtfunkssystem | 14 |
| Abbildung 2-17 Ad-hoc-Netzwerke..... | 14 |
| Abbildung 2-18 Grundprinzip des WEP in IEEE 802.11 | 15 |
| Abbildung 2-19 Hardwareumgebung des NETs [hero02] | 17 |
| Abbildung 2-20 Protokollstruktur der Emulation für MAC-Protokoll | 18 |
| Abbildung 3-1 MAC-Architektur in IEEE 802.11 [IEEE811] | 19 |
| Abbildung 3-2 Sendestationen senden ihre Daten immer weiter, obwohl Kollision auftritt... .. | 20 |
| Abbildung 3-3 Zugriffsmechanismus in CSMA/CA [IEEE811] | 21 |
| Abbildung 3-4 die ideale Umgebung | 22 |
| Abbildung 3-5 Mobilstationen liegen innerhalb eines überlappenden Bereiches..... | 22 |
| Abbildung 3-6 Zeitdiagramm des Basismechanismus von CSMA/CA | 23 |
| Abbildung 3-7 Zeitdiagramm von CSMA/CA mit ACK..... | 25 |
| Abbildung 3-8 Standorte der Mobilstationen..... | 26 |
| Abbildung 3-9 „Hidden Terminal“ Problem..... | 27 |
| Abbildung 3-10 „Exposed Terminal“ Problem..... | 28 |
| Abbildung 3-11 Rahmenkommunikationen bei CSMA/CA mit RTS/CTS | 29 |
| Abbildung 3-12 Standorte der Mobilstationen..... | 29 |
| Abbildung 3-13 Zeitdiagramm des CSMA/CA mit RTS/CTS | 30 |
| Abbildung 3-14 Zeitdiagramm von PCF..... | 32 |
| Abbildung 3-15 Nachrichtenübertragung mit Fragmentierung..... | 34 |
| Abbildung 4-1 WILEMU-Rahmen muss getunnelt gesendet werden | 35 |
| Abbildung 4-2 Mobilstationen haben verschiedene Reichweiten..... | 36 |
| Abbildung 4-3 Allgemeines Rahmenformat in IEEE 802.11 MAC [IEEE811],[nemg01] | 39 |
| Abbildung 4-4 Allgemeines Rahmenformat in WILEMU..... | 41 |
| Abbildung 4-5 Rahmensformate von ACK, RTS und CTS nach IEEE 802.11 [IEEE811] | 41 |
| Abbildung 4-6 Format des Steuerungsrahmens in WILEMU..... | 42 |
| Abbildung 4-7 CW in CSMA/CA [IEEE811]..... | 42 |
| Abbildung 4-8 Vergleich der Übertragungsdauer | 44 |
| Abbildung 4-9 IFS-Intervalle in IEEE 802.11 DCF [IEEE811] | 46 |
| Abbildung 4-10 Zeitberechnung in IEEE 802.11 DCF [IEEE811] | 47 |
| Abbildung 4-11 Zustandsdiagramm - „IDLE“ | 49 |

| | |
|---|----|
| Abbildung 4-12 Zeitdiagramm – „CONTENTION“ | 51 |
| Abbildung 4-13 Zeitdiagramm - „BROADCASTING“ | 51 |
| Abbildung 4-14 Zeitdiagramm - Senden des RTS und Warten auf CTS | 52 |
| Abbildung 4-15 Zustandsdiagramm – Anschluss 1 (Empfang eines gültigen Rahmens)..... | 52 |
| Abbildung 4-16 Zeitdiagramm – „RECEIVING_BC“ (Empfang eines Broadcast-Rahmens) 53 | |
| Abbildung 4-17 Zeitdiagramm – „LISTENING“ | 53 |
| Abbildung 4-18 Zeitdiagramm - Empfang des RTS und Absenden von CTS..... | 54 |
| Abbildung 4-19 Zustandsdiagramm – „CONTENTION“ | 55 |
| Abbildung 4-20 Zustandsdiagramm – „LISTENING“ | 56 |
| Abbildung 4-21 Zustandsdiagramm – „WAITING_CTS“ | 57 |
| Abbildung 4-22 Zeitdiagramm - Empfang des CTS und Absenden von DATA..... | 58 |
| Abbildung 4-23 Zustandsdiagramm – „WAITING_ACK“ | 59 |
| Abbildung 4-24 Zustandsdiagramm – „WAITING_DATA“ | 60 |
| Abbildung 4-25 Zustandsdiagramm – „RECEIVING_DATA“ | 61 |
| Abbildung 4-26 Zustandsdiagramm – „BROADCASTING“ | 62 |
| Abbildung 4-27 Zustandsdiagramm – „RECEIVING_BC“ | 63 |
| Abbildung 5-1 WILEMU im Kommunikationsstapel..... | 65 |
| Abbildung 5-2 Schnittstelle-Funktionen der WILEMU..... | 66 |
| Abbildung 5-3 Vereinfachter Algorithmus von WILEMU..... | 69 |
| Abbildung 6-1 Test 1..... | 71 |
| Abbildung 6-2 Durchsatz bei einem Sender/Empfänger..... | 72 |
| Abbildung 6-3 Test 2..... | 72 |
| Abbildung 6-4 Durchsatz bei zwei Sendern und einem Empfänger | 73 |
| Abbildung 6-5 Test 3..... | 74 |
| Abbildung 6-6 „Round-Trip-Time“ mit „Ping“ | 74 |

Tabellenverzeichnis

| | |
|--|----|
| Tabelle 2-1 Beschreibung der 7 Netzwerkschichten [roth01]..... | 3 |
| Tabelle 2-2 Wartezeit nach Kollisionen (Binär-Exponential-Back-Off) | 9 |
| Tabelle 2-3 Übertragungsrate in IEEE 802.11 [IEEE811], [IEEE811b], [nemg01]..... | 10 |
| Tabelle 2-4 Unterschiede zwischen IEEE 802.11b und IEEE 802.3 | 16 |
| Tabelle 4-1 Verbindungsfehlerrate..... | 36 |
| Tabelle 4-2 Fehlerrateliste bei Station B..... | 37 |
| Tabelle 4-3 Signalausbreitungsverzögerung in IEEE 802.11 | 37 |
| Tabelle 4-4 Netzwerktopologie und Medienzugriffsverfahren..... | 37 |
| Tabelle 4-5 Wichtige Datentypen und Subtypen in IEEE 802.11 [IEEE811] | 38 |
| Tabelle 4-6 Rahmenstyp in der Emulation..... | 39 |
| Tabelle 4-7 Adressenfelder in MAC-Kopf nach IEEE 802.11 [siko01]..... | 40 |
| Tabelle 4-8 Werte der BackOff-Wartezeit in CSMA/CA | 42 |
| Tabelle 4-9 Werte der Zeitschlitz in CSMA/CA [IEEE811], [IEEE811b]..... | 43 |
| Tabelle 4-10 IFS in IEEE 802.11 [IEEE811], IEEE811b]..... | 46 |
| Tabelle 4-11 IFS in WILEMU | 47 |
| Tabelle 4-12 Definition der WILEMU-Zuständen..... | 48 |
| Tabelle 4-13 Lokale Variablen in WILEMU | 48 |
| Tabelle 5-1 Entwicklungsumgebung..... | 65 |
| Tabelle 5-2 Bestandteile von WILEMU | 65 |
| Tabelle 5-3 Definitionen von „cmd“ | 68 |
| Tabelle 6-1 Messergebnis bei einem Sender/Empfänger | 72 |
| Tabelle 6-2 Messergebnis bei zwei Sendern und einem Empfänger | 73 |
| Tabelle 6-3 Messergebnis „Ping“ | 74 |

Kapitel 1 Einführung

1.1 Motivation

Dank der modernen Internet- und Halbleitertechnologien nimmt die Vernetzung von Computern und Mobilgeräten immer mehr zu. Nach den rapid wachsenden Bedürfnissen an flexiblerer Arbeitsumgebung, besserer Erreichbarkeit, effizienter Verwaltung, digitaler Unterhaltung und Heimnetzwerken, insbesondere an den mobilen Ad-hoc-Netzwerke (MANET) wurden mobile und drahtlose Netzwerke wie Wireless LAN (Drahtlose Lokale Netzwerke) rasant verbreitet.

Im Bürogebäude, wo Verkabelung ungünstig oder sogar unmöglich ist, können Rechner mittels Wireless LAN Technik vernetzt werden. Durch Zugangspunkte können Wireless LAN auch mit dem Unternehmens Ethernet verbunden werden. Innerhalb der Firma können sich mobile Teilnehmer frei bewegen. Während Reisen wollen die Geschäftsleute einen Zugang zum öffentlichen Diensten, wie z.B. Email, WWW. Wireless LAN kann in z.B. Flughäfen oder Bahnhöfen eingesetzt werden. Damit wird ein „mobiler Schreibtisch“ ermöglicht. Die Aufgaben können schneller erledigt werden. Der Bedarf von Heimvernetzung und SOHO-Vernetzung (Small Office, Home Office) wächst durch Unterhaltungselektronik und Internettechnik. Mit einem einzigen Internet-Zugang können Leute mit Laptop oder Desktop irgendwo im Haus Internet surfen, Email versenden. Kinder und Eltern spielen im Netzwerk, ohne Computer durch Kabel anzubinden. Wireless LAN ist eine gut passende Variante dazu. In Hörsale oder Bibliothek durch Einsatz von Wireless LAN haben die Studenten die Möglichkeit, auf Sitzplätze die Vorlesungsmaterialien oder Literaturen zu lesen.

Wireless LAN ist nicht nur ein Ersatz für kabelgebundene LAN entwickelt worden. Drahtlose Ad-hoc Netzwerke bieten sogar völlig neue Einsatzfelder.

Die Eigenschaften der drahtlosen Netze unterscheiden sich wesentlich von klassischen kabelgebundenen Netzen. Wegen der Unzuverlässigkeit des Funkkanals ist die Verlust- und Fehlerrate der Datenübertragung höher als in kabelgebundenen Netze. Die Mobilität der Netzwerkteilnehmer verursacht eine dynamische Änderung von Netzwerktopologien und -Partitionen.

Dafür sind neue Strategien für Routing, Fehlerbeseitigung, Datenverwaltung usw. zu definieren und eine Menge von neuen Anwendungen und Protokollen zu entwickeln. Sie nehmen in der aktuellen Forschung einen immer größeren Raum ein.

Während der Entwicklung müssen die Leistungen von Anwendungen und Protokollen analysiert werden. Für die Leistungsanalyse und Fehlersuche wird eine leistungsfähige und komfortable Testumgebung benötigt.

Das Network Emulation Testbed (NET) der Abteilung Verteilte Systeme an der Universität Stuttgart ist ein flexibel vernetzter Linux PC-Cluster mit 64 Knoten, in dem beliebige Netztopologien und deren Eigenschaften nachbildet werden können. Damit bietet das NET eine ideale Analyseumgebung für verteilte Anwendungen und Netzprotokolle. Nun sollen auch Untersuchungen auf der Grundlage eines emulierten Wireless LAN in NET unterstützt werden.

1.2 Aufgabenstellung

Ziel dieser Diplomarbeit ist die realistische Nachbildung von IEEE 802.11b (Wireless LAN) im NET an der Abteilung Verteilte Systeme. Im Rahmen des NET-Projekts ist bereits Software für die Emulation der Bitübertragungsschicht von Funknetzen entstanden. Aufbauend auf diesem „Emulierten Funkkanal“ muss die Medienzugriffssteuerung (MAC) von IEEE 802.11b emuliert werden.

Im Rahmen dieser Diplomarbeit ist zunächst das in IEEE 802.11b beschriebene MAC-Protokoll eingehend zu analysieren, welche Eigenschaften überhaupt in Software nachbildet werden können und wo die Vereinfachungen nötig sind. Dann ist ein Verfahren zu entwickeln, wie diese Nachbildung auf der Grundlage eines emulierten Funkkanals realisiert werden kann.

1.3 Struktur der Arbeit

In Kapitel 2 werden die grundlegende Netzwerktechnik sowie die allgemeine Medienzugriffsverfahren eingeführt. Die Grundlagen der LANs werden auch in diesem Kapitel vorgestellt. Eine kurze Vorstellung über NET (Network Emulation Testbed) der Universität Stuttgart befindet sich am Ende dieses Kapitels.

Die Medienzugriffsverfahren in Wireless LAN werden in Kapitel 3 detailliert analysiert.

In Kapitel 4 wird zunächst das Emulationskonzept des Funkkanals beschrieben. Zur Emulation der in IEEE 802.11 spezifizierten MAC-Schicht werden die Probleme sowie daraus resultierende Vereinfachungen diskutiert. Der Entwurf der Emulationssoftware „WILEMU“ folgt am Ende des Kapitels.

Eine Beschreibung der Implementierung sowie der grundlegende Algorithmus werden in Kapitel 5 dargestellt. Die Testmessungen der Implementierung und Evaluation werden in Kapitel 6 beschrieben.

In Kapitel 7 wird der aktuelle Entwicklungsstand der Netzwerke emulation, sowohl zentralisierte als auch verteilte Lösungen vorgestellt. Anschließend wird eine Zusammenfassung und ein Ausblick zu weiteren Arbeiten in Kapitel 8 gegeben.

Kapitel 2 Grundlagen

2.1 Das ISO/OSI Referenzmodell

Um die Entwicklungskomplexität zu verringern und eine bessere Erweiterbarkeit zu gewährleisten, werden die modernen Netzwerke schichtweise organisiert. Untere Schicht bietet Dienste an obere Schicht und obere Schicht ist Dienstverbraucher der unteren Schicht. Sie kommunizieren sich durch Dienstzugangspunkt. Die n-te Schicht zwischen einer Station und einer fernen Station, die im gleichen Netz steht, kommunizieren miteinander nach Kommunikationsprotokoll. In solchen Kommunikationsprotokollen werden z.B. Funktionsweise und Datenformat definiert.

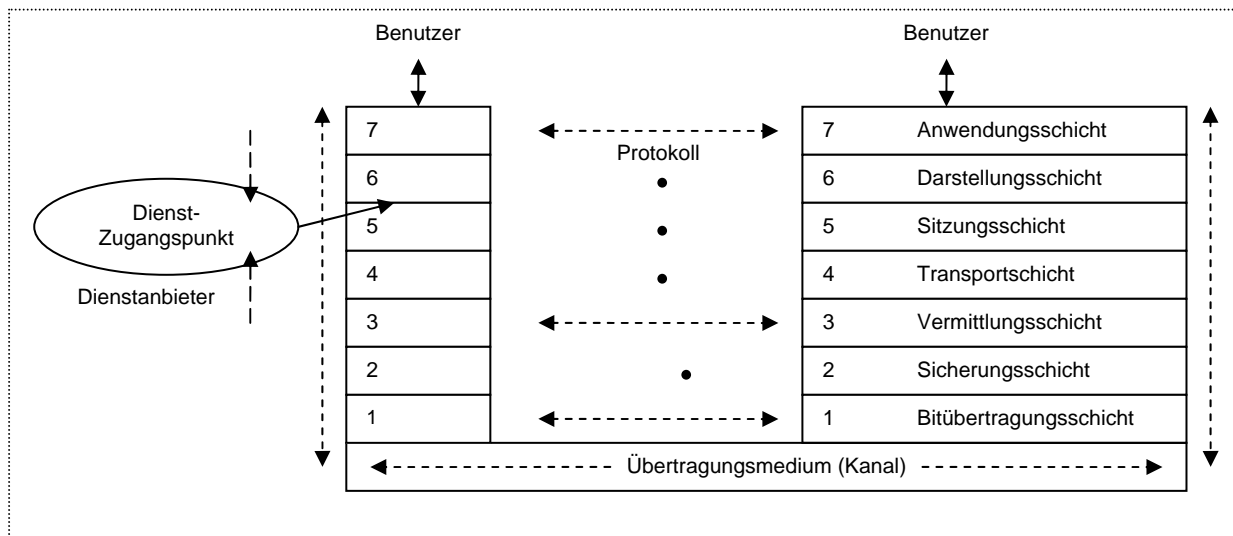


Abbildung 2-1 ISO/OSI Referenzmodell [tane97]

Die ISO (International Standards Organisation) hat im Jahr 1983 eine OSI (Open System Interconnection) Referenzmodell veröffentlicht. Wie die Abbildung 2-1 zeigt, werden sieben Netzwerk-Schichten in ISO/OSI Modell definiert:

| | | |
|----|------------------------|---|
| 7. | Anwendungsschicht | Anwendungsbezogene Dienste (z.B. WWW) |
| 6. | Darstellungsschicht | Globale Datendarstellung (Abstrakte Syntax...) |
| 5. | Sitzungsschicht | „Sitzung“ zwischen Prozessen (Resynchronisation nach unterbrochener Verbindung...) |
| 4. | Transportschicht | Kommunikation von Nachrichten zwischen Prozessen |
| 3. | Vermittlungsschicht | Kommunikation von Paketen zwischen Endsystemen (Routing, Adressierung, z.B. IP) |
| 2. | Sicherungsschicht | Zuverlässige Übertragung von Rahmen (Fragmentierung, Fehlererkennung und -beseitigung, Flusskontrolle) |
| 1. | Bitübertragungsschicht | Übertragung von Bitströmen Hardwareformat (Kabel, Stecker, Dose...) Signalfomat (Spannung, Codierung..) |

Tabelle 2-1 Beschreibung der 7 Netzwerkschichten [roth01]

2.2 LAN-Standards - IEEE 802.x Familie

Im Bereich der LANs (Lokale Netzwerke, vgl. Abschnitt 2.3 und 2.5) hat die IEEE (Institute for Electrical and Electronic Engineers) die wichtigsten Standards veröffentlicht, wie z.B. IEEE 802.3 Ethernet, IEEE 802.5 Token-Ring in der IEEE 802.x Familie (siehe Abbildung 2-2). Neben dem Fortschritt der Technologie nimmt die IEEE 802.x-Familie weiter zu. Im Jahr 1997 wurde der IEEE 802.11 Standard für Wireless LAN (Drahtlose Lokale Netzwerke) entstanden. Die Standardisierung bietet viele Vorteile für Entwicklung, Erweiterung und Anwendung.

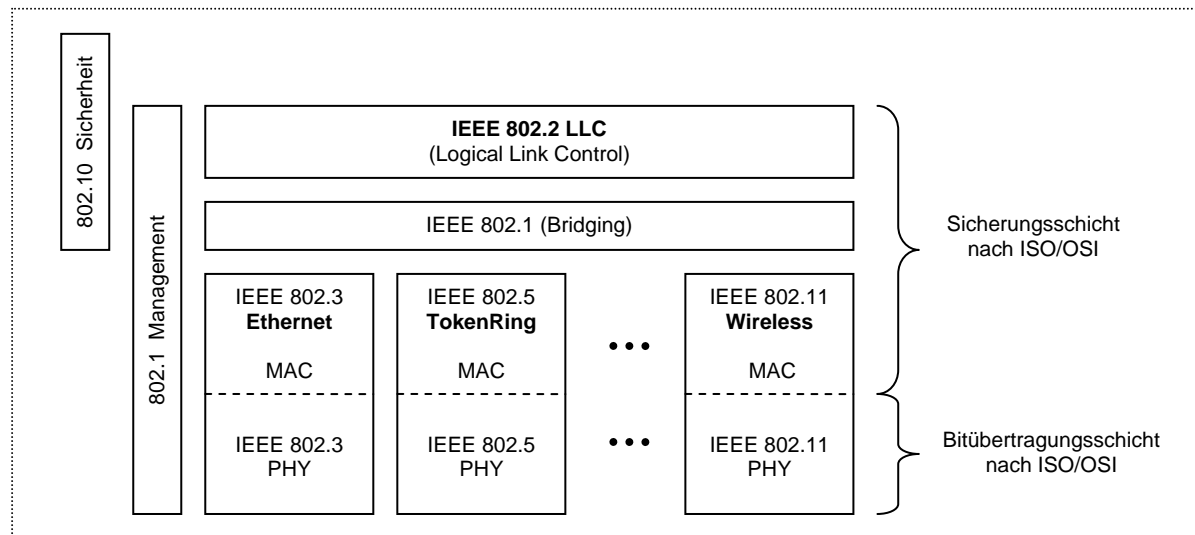


Abbildung 2-2 LAN Standards - IEEE 802.x Familie [IEEE 811]

Durch Standardisierung ist der Einsatz der Hardware-Technik für die Entwicklung der oberen Software-Ebene transparent, wie z.B. bei den Entwicklungen im Bereich von Ethernet. Auf der Basis von Koaxialkabeln ist die Übertragungsrate bis zu 10 MBit/s. Die Übertragungsrate steigt in Ethernet auf 100 MBit/s mit Einsatz von Twisted-Pair-Verkabelung. Trotz dem Fortschritt der Übertragungstechnik bleiben die bereits bestehende Software und Infrastruktur unverändert und weiterhin verwendbar.

Dank der Standardisierung können die unterschiedlichen Hardware- und Softwarehersteller vielfältige Produkte und Komponenten aufbauen. Die Erweiterung, Verwaltung und Integration sind dadurch vereinfacht.

Das ISO/OSI Referenzmodell beschreibt die Schichtenstruktur grundsätzlich für Punkt-zu-Punkt-Kommunikationen. In Bezug auf lokale Netzwerke wird ein Broadcast-fähiges Medium zur Datenübertragung in einem Netz verwendet. Damit die logischen Punkt-zu-Punkt-Kommunikationen auf ein zusammenbenutztes und freigegebenes Übertragungsmedium funktionieren, ist eine MAC-Schicht (Medienzugriffssteuerung) in die 2-te Schicht gesondert eingefügt, wie die Abbildung 2-2 zeigt. Die Standards IEEE 802.3, 802.5 und 802.11 beschreiben die Bitübertragungsschichten und MAC-Schichten, entsprechend in Ethernet, Token-Ring und Wireless LAN. Die MAC-Schicht sorgt für die ordentlichen und reibungslosen Medienzugriffe, z.B. wer darf das Zugriffrecht haben und wann, wenn eine Kollision auftritt, wie soll es behandelt werden? Anschließend steht über der MAC-Schicht noch eine andere Teilschicht in der Sicherungsschicht nach ISO/OSI, nämlich die LLC-Schicht (Logische Verbindungssteuerung). Die LLC-Schicht übernimmt die übrigen Aufgaben in Sicherungsschicht nach ISO/OSI, wie z.B. Flusskontrolle. Für diese LLC-Teilschicht sind die Hardware, Netzwerktopologien, Medienzugriffsverfahren transparent. Die IEEE 802.2 bietet eine einheitliche Schnittstelle zwischen der Vermittlungsschicht und IEEE 802.x MAC-Schicht.

Die Abbildung 2-3 zeigt den Pfad des allgemeinen Nachrichtentransports. Auf der Senderseite, nimmt die untere Schicht die Daten aus der oberen Schicht als ihre Nutzdaten und fügt einen entsprechenden Kopf. Danach schickt die gesamte Nachricht nach unten. Umgekehrt, nach der Analyse des entsprechenden Kopfs, wird der Nachrichtenkopf auf der Empfängerseite schichtweise entfernt.

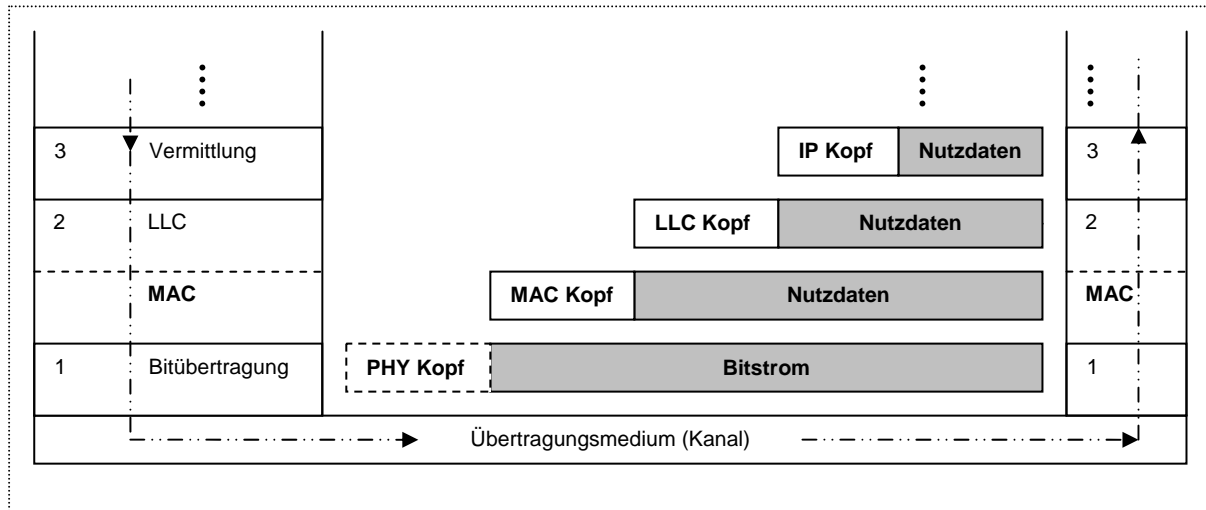


Abbildung 2-3 Transportpfad

2.3 Wired LAN (Kabelgebundene Lokale Netzwerke)

Die kabelgebundenen lokalen Netzwerke unterscheiden sich von den Netzwerken anderer Arten, wie z.B. WAN (Wide Area Network) und MAN (Metropolitan Area Network) durch ihre Netzwerktopologien, verwendete Übertragungstechnik und die Entfernung.

Kabelgebundene Lokale Netzwerke sind in einem gewissen räumlichen Bereich eingesetzt. Seine maximale Entfernung beschränkt sich auf einige Kilometer, z.B. in einem Gebäude, in Campus. Damit hat die Ausbreitungsdauer des Signals eine obere Grenze. Alle Teilnehmer in einem lokalen Netzwerk sind normalerweise mit einem zusammenbenutzten, freigegebenen Übertragungsmedium in Form von Bus oder Ring verbunden. Die Abbildung 2-4 zeigt zwei typische Netzwerktopologien.

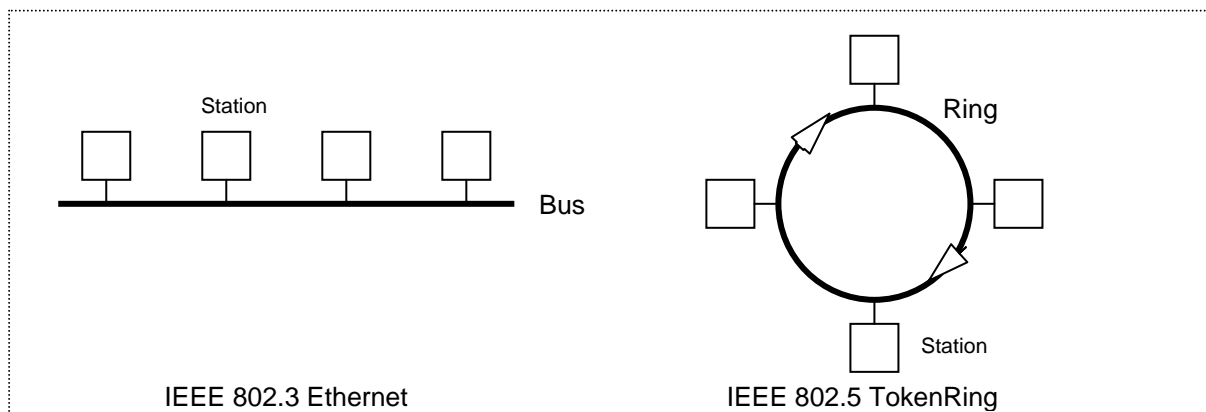


Abbildung 2-4 Typische Netzwerktopologien von LAN [tane97]

Das in IEEE 802.3-Standard (vgl. Abschnitt 2.3) definierte Ethernet ist eine der am weitesten verwendeten LAN-Techniken. In Ethernet wird ein Koaxialkabel als Übertragungsmedium verwendet

und die Übertragungsrate beträgt bis zu 10 MBit/s. Durch Einsatz von Twisted-Pair-Verkabelung kann die maximale Übertragungsrate 100 MBit/s erreicht werden. IEEE 802.5 Token-Ring ist ein Ring-Topologie-Netzwerk. Die typische Übertragungsrate in Token-Ring ist 4 MBit/s oder 16 MBit/s. Das gemeinsam benutzte Medium bietet einen Kommunikationskanal, sowohl für Broadcasting als auch für logische Punkt-zu-Punkt-Verbindungen. Jeder Netzwerkteilnehmer besitzt eine weltweit identische Hardware-Adresse (MAC-Adresse). Wenn eine Station eine Nachricht senden möchte, fügt sie zunächst die Adresseinformationen in den Rahmenkopf (MAC-Kopf, siehe Abbildung 2-4) der zu sendenden Nachricht. (Das Format des Rahmenkopfs ist in Abbildung 2-5 zu sehen.)

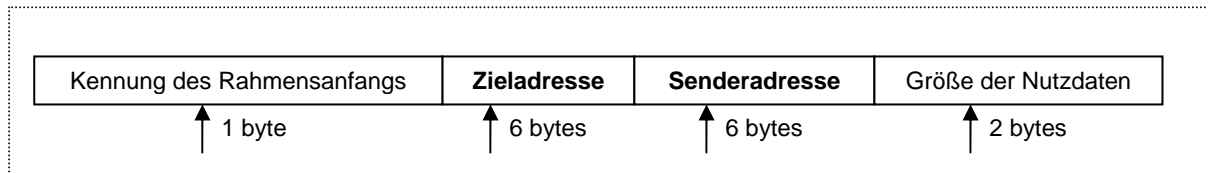


Abbildung 2-5 Wichtige Bestandteile des MAC-Kopfs in Ethernet

Alle anderen Stationen, die an das Netz gebunden sind, können alle auf Medium laufenden Rahmen hören. Nach der Analyse der Adresseinformation in dem Rahmenkopf entscheiden sich die Stationen, ob dieser Rahmen empfangen werden soll. So ist die logische Punkt-zu-Punkt-Verbindung auf der Empfängerseite realisiert. Auf der Senderseite ist es mit Hilfe von Medienzugriffssteuerungen realisiert worden

2.4 Medienzugriffsverfahren in LAN

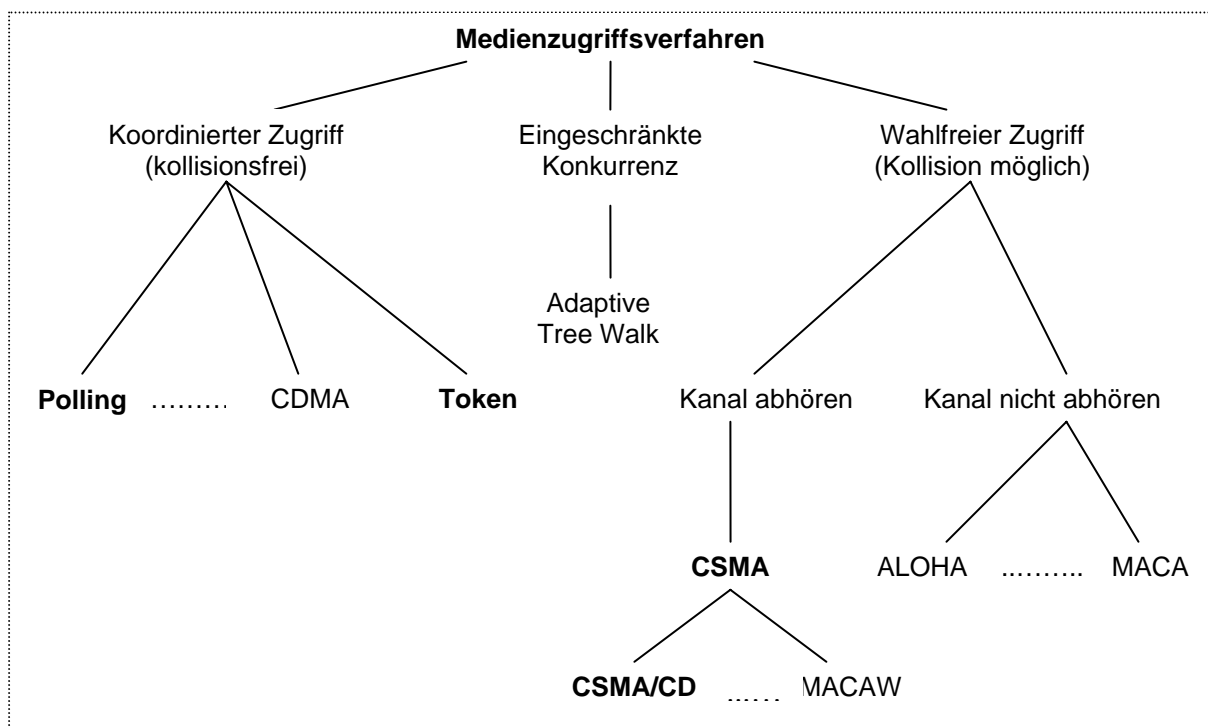


Abbildung 2-6 Medienzugriffsverfahren [roth01]

In LAN ist ein Medienzugriffsverfahren notwendig. Solches Verfahren ist zuständig für die exklusiven Kanalzugriffe und Kollisionsbeseitigung.

2.4.1 Kollisionsfreies Verfahren

2.4.1.1 Polling

Das einfachste kollisionsfreie Medienzugriffsverfahren ist „Polling“. Wie die Abbildung 2-7 zeigt, ist ein Koordinator zuständig für die Zugriffssteuerung auf das Medium. Er verwaltet eine Polling-Liste, auf der alle Netzwerkteilnehmer eingetragen sind. Unter dem Befehl von Koordinator läuft ein „Polling“-Zeiger zyklisch in der Liste. Nach dem Ende einer Übertragung fragt der Koordinator dem Zeiger, welche die nächste Station in der Liste ist. Danach schickt der Koordinator eine Nachricht an die entsprechende Station, dass sie auf Medium zugreifen darf. Eine Station darf ihre Daten nur senden, wenn sie vom Koordinator aufgerufen wird.

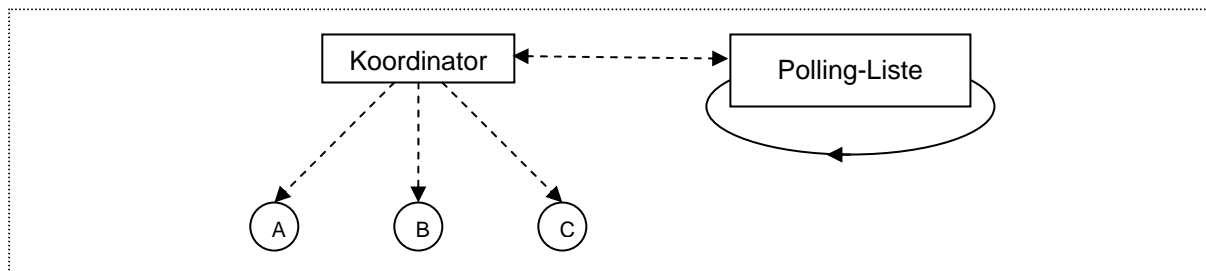


Abbildung 2-7 Polling-Verfahren

2.4.1.2 TokenRing

Ein anderes Beispiel für kollisionsfreies Verfahren, koordiniertes Zugriffsverfahren, ist der in IEEE 802.5 definierte Token-Ring-Mechanismus. Wie die Abbildung 2-4 zeigt, läuft auf dem Ring ein „Token“. Eine Station darf Nachrichten senden, wenn sie „Token“ besitzt. Am Ende des Sendens setzt die Senderstation das „Token“ wieder frei, d.h. das „Token“ läuft wieder auf dem Ring und ist für andere Stationen verfügbar. Dadurch werden Kollisionen theoretisch nie auftreten, bzw. ein kollisionsfreier Übertragungskanal wird aufgebaut. Allerdings ist die Verwaltung von „Token“ sehr aufwendig. Die Hardware-, Software- oder Übertragungsfehler könnten dazu führen, dass das „Token“ nicht mehr verfügbar, beschädigt oder verdoppelt auf dem Ring läuft. Deswegen ist darüber hinaus ein effizienter „Token“-Verwaltungsmechanismus in Token-Ring erforderlich.

2.4.2 CSMA

2.4.2.1 CSMA - Überblick

Das CSMA-Verfahren (Carrier Sense Multiple Access) ist eine Erweiterung von ALOHA [tane97]. Der Hauptunterschied zwischen den Beiden ist: bei CSMA, vor dem Senden hört die Senderstation das Medium ab, falls das Medium im Augenblick frei ist, sendet sie ihre Rahmen auf Medium, wenn nicht wird das Senden sich verzögert. Dadurch wird die Kollisionswahrscheinlichkeit im Vergleich mit ALOHA dramatisch verringert. Es gibt viele Varianten von CSMA, 1-Persistentes-CSMA und P-Persistentes-CSMA [roth01].

2.4.2.2 CSMA/CD

Das in IEEE 802.3-Ethernet beschriebene Verfahren zur Medienzugriffssteuerung ist der auf 1-persistentem-CSMA basierende CSMA/CD-Mechanismus (CSMA mit Erkennung von Kollisionen).

Nach den Regeln in CSMA, obwohl der Zustand der Kanalbelegung vor der Datenübertragung berücksichtigt wird, besteht noch Kollisionsgefahr. Beispielsweise wenn zwei weit entfernte Stationen gleichzeitig ihre Daten auf Medium senden möchten, sind die Kanalbelegungszustände bei beiden Stationen momentan als „frei“ erkannt worden. Wegen einer Signalausbreitungsverzögerung werden die zwei gesendete Rahmen in einem späteren Zeitpunkt in Kollision treffen. In Ethernet wird bei jedem Netzwerkteilnehmer ein analoges Gerät verwendet, das zur Erkennung von Kollision dient. Während des Sendens vergleicht das Erkennungsgerät der Senderstation ihre gesendeten Signale mit den auf Kanal laufenden Signalen, z.B. die Energie oder Impulsbreite der Signale. Wenn Kollision auftritt, kann das Erkennungsgerät das durch Kollision verursachte Geräusch detektieren und dementsprechend das Senden sofort abbrechen. Damit ist die unnötige Kanalnutzung erspart, da die Nachricht wegen Kollision sowieso beschädigt worden. Alleine aus diesem Grund ist CSMA/CD deutlich effizienter als normales CSMA Verfahren. Abbildung 2-8 veranschaulicht das Grundprinzip von CSMA/CD.

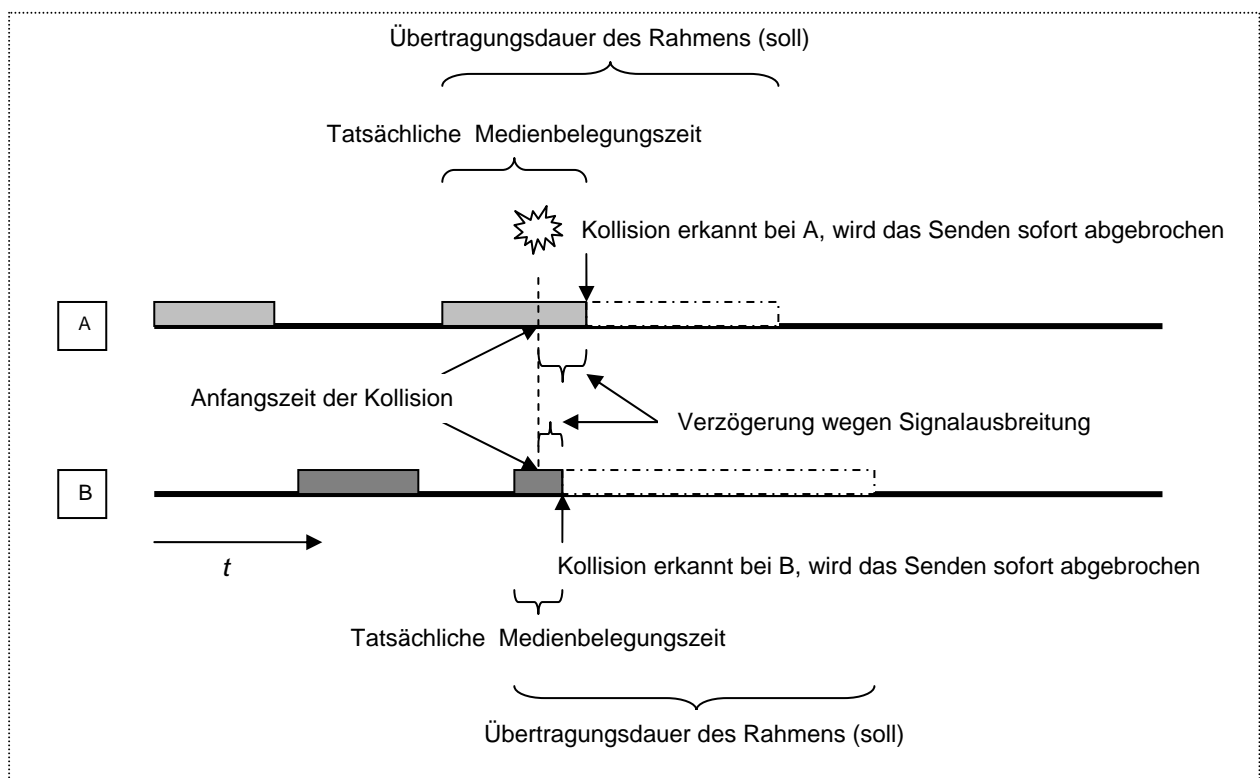


Abbildung 2-8 Grundprinzip des CSMA/CD

Wie die Abbildung 2-9 zeigt, wenn das von A am t_0 gesendete Rahmen sehr kurz wäre und eine Kollision in der Nähe von einer weit entfernten Senderstation B am t_1 passieren sollte, ist der Sendevorgang bei A am t_1 schon beendet, bevor das Geräusch an A ankam. D.h. die Kollision kann nicht mehr bei A erkannt werden.

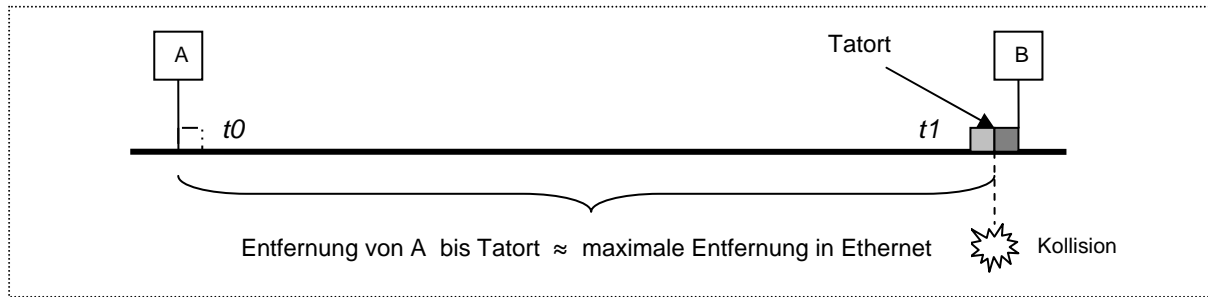


Abbildung 2-9 Minimale Rahmensgröße in CSMA/CD

Um dieses Problem zu lösen, ist eine minimale Rahmensgröße bei Ethernet definiert. Die Entfernung zwischen A und B sei die maximale Buslänge. Bei IEEE 802.3 Ethernet, ist die maximale Buslänge L 2500 Meter mit maximal 4 Verstärkern, die Signalausbreitungsgeschwindigkeit V ca. 200 T KM/s und die Bitübertragungsrate B 10MBit/s. Daraus folgt die minimale Rahmensgröße ca. 64 Bytes (beträgt $\tau \approx 51.2 \mu s$).

In IEEE 802.3 Ethernet ist eine maximale Rahmensgröße (nur Nutzdaten) auf 1500 Bytes beschränkt, damit der Kanal für eine Station nicht zu lang belegt wird.

2.4.2.3 Binär-Exponential-Back-Off-Funktion für CSMA/CD

Durch Verwendung von wahlfreien Verfahren können Kollisionen auftreten. Um eine bessere Kanalnutzung und einen besseren Durchsatz zu erzielen, soll eine Station von den in Kollision beteiligten Senderstationen nach einer Kollision so schnell wie möglich das Zugriffsrecht auf Kanal bekommen und die Anderen die Übertragung nicht stören. Der Einsatz des Binär-Exponential-Back-Off-Algorithmus kann diese Anforderung in CDMA/CD am besten erfüllen.

| Anzahl der Kollisionen | Intervall der Zufallszahl | Mögliche Wartezeit nach Kollision |
|------------------------|---------------------------|---|
| 1 | [0,1] | 0, τ |
| 2 | [0,1...3] | 0, τ , 2τ , 3τ |
| 3 | [0,1,2...7] | 0, τ , 2τ , 3τ ... 7τ |
| 4 | [0,1,2,3...15] | 0, τ , 2τ , 3τ , 4τ ... 15τ |
| | | |
| n , n in [1,10] | [0,1,2,3... 2^{n-1}] | 0, τ , 2τ , 3τ , 4τ , 5τ ... $(2^{n-1})\tau$ |
| n , n in (10,16] | [0,1,2,3 ... 1023] | 0, τ , 2τ , 3τ , 4τ , 5τ ... 1023τ |

Tabelle 2-2 Wartezeit nach Kollisionen (Binär-Exponential-Back-Off)

Da in CSMA/CD keine zentrale Station zum Ordnen und Verteilen der Wartezeit für die in Kollision beteiligten Stationen eingesetzt ist, besteht hier keine globale Sicht auf Kanal. Wenn ein Rahmen zum ersten Mal an einer Kollision beteiligt ist, weiß deren Senderstation, dass es mindestens eine andere Station gibt, die gerade auch Rahmen sendet. So wartet sie eine diskrete Zeit von 0 bis τ ab, wird sie ihr Rahmen nach dem Ablauf der Wartezeit noch mal absenden. Die Zeitdauer τ ist ein Zeitschlitz wie es gerade in Abschnitt 2.4.2.2 erklärt ist. Wenn der Rahmen noch einmal in Kollision getroffen ist, wird das Intervall der Wartezeit verdoppelt, wie die Tabelle 2-1 zeigt.

Um die unendliche Wiederholung des Zugriffsversuchs zu vermeiden, ist die maximale Anzahl der Wiederholung auf 16 beschränkt. Übrigens ist die obere Grenze des Intervalls auf 1023 ($2^{10}-1$, also

10 Wiederholungen) definiert, da die Wartezeit nach 10 Wiederholungen schon im Verhältnis von Datenübertragungen zu lang würde.

2.5 Wireless LAN (Drahtlose Lokale Netzwerke)

Die kabelgebundenen lokalen Netzwerke, wie z.B. Ethernet, sind sehr weit verbreitet dank seiner Zuverlässigkeit, Einfachheit, schnellerer Übertragungsrate und verfeinerter Technik und Anwendungen. Mobilität und Flexibilität sind die neuen Herausforderungen für die Informationstechnik im Bezug auf lokale Netzwerken.

Die Eigenschaften der drahtlosen Netze unterscheiden sich wesentlich von klassischen kabelgebundenen Netzen. Statt die Netzwerkteilnehmer durch Kabel miteinander anzubinden, ist Funkkanal in Wireless LAN zur Datenübertragung verwendet. Wegen der Unzuverlässigkeit des Funkkanals ist die Verlust- und Fehlerrate der Datenübertragung höher als die in kabelgebundenen Netzen. Die Anzahl von Netzwerkteilnehmern in Wireless LAN ist hoch dynamisch. Die Mobilität der Netzwerkteilnehmer verursacht auch eine dynamische Änderung von Netzwerktopologien und -Partitionen.

2.5.1 Wireless LAN Standard

Wie in Abschnitt 2.2 erläutert, durch Standardisierung wird die Entwicklung beschleunigt und die Systemintegration und Verwaltung auch vereinfacht. Der Standard für Wireless LAN ist IEEE 802.11. Er wurde ursprünglich im Jahr 1997 von dem „Institute of Electronic and Electrical Engineers“ (IEEE) veröffentlicht. Wie der Name dieses Standards schon andeutet, gehört diese Spezifikation zu der Standard-Familie IEEE 802.x für lokale Netzwerke. Die IEEE 802.11 Standards beschreiben die Bitübertragungsschicht und MAC-Schicht, die speziell für die Übertragung in drahtlosen Netzwerken notwendig sind. Die MAC-Schicht bietet nach oben hin die gleiche Schnittstelle wie andere Mitglieder der IEEE 802.x Familie, wie die Abbildung 2-2 zeigt.

2.5.2 Bitübertragungsschicht in IEEE 802.11 - Überblick

Eine Erweiterung auf der Bitübertragungsschicht (PHY) des IEEE 802.11, nämlich der IEEE 802.11b-Standard, wurde im September Jahr 1999 veröffentlicht. Die in IEEE 802.11b spezifizierte Bitübertragungsschicht unterstützt nun DSSS-Technologie (Direct Sequence Spread Spektrum) für hohe Übertragungsrate. Dadurch erreicht die Übertragungsrate 11 MBit/s, während im originalen IEEE 802.11 nur bis zu 2 MBit/s unterstützt ist. Tabelle 2-3 listet die Familiemitglieder in IEEE 802.11 auf. Die Abbildung 2-10 veranschaulicht die Frequenzbänder im elektromagnetischen Spektrum als Referenz.

| Standard | PHY-Technologie | Basis-Frequenz | Unterstützte Übertragungsrate [MBit/s] |
|--------------|---|----------------|--|
| IEEE 802.11 | IR (Infrared) | 300 THz | 1, 2 |
| | FHSS (Frequency Hopping Spread Spektrum) | 2.4 GHz | 1, 2 |
| | DSSS (1997) | 2.4 GHz | 1, 2 |
| IEEE 802.11a | OFDM (Orthogonal Frequency Division Multiplexing) | 5 GHz | 6, 9, 12, 18, 24, 36, 54 |
| IEEE 802.11b | DSSS (1999, extended DSSS) | 2.4 GHz | 5.5, 11 |

Tabelle 2-3 Übertragungsrate in IEEE 802.11 [IEEE811], [IEEE811b], [nemg01]

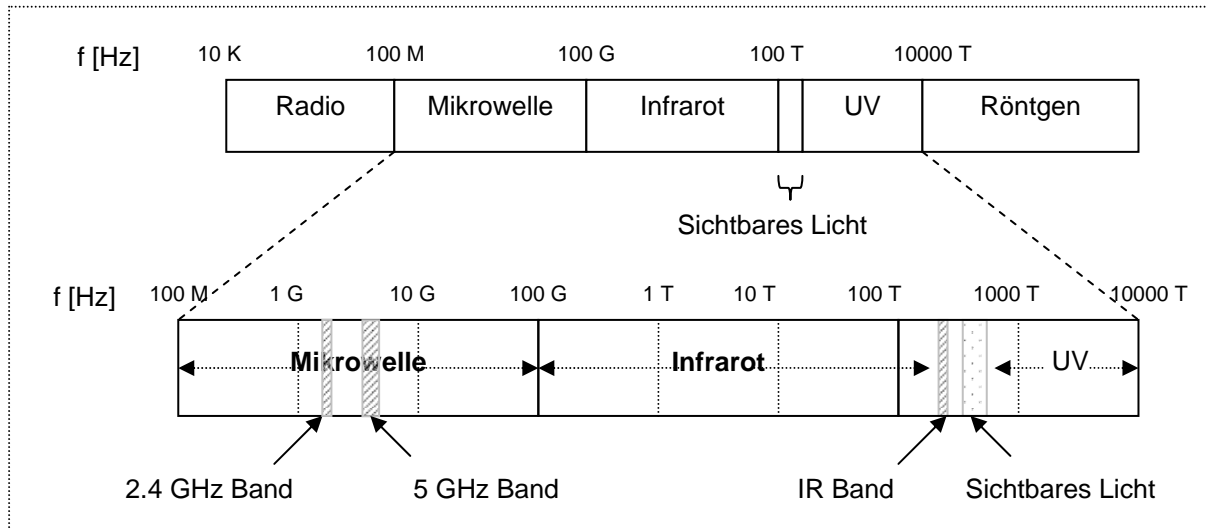


Abbildung 2-10 Frequenzbänder im elektromagnetischen Spektrum [nemg01]

Wie die Abbildung 2-11 zeigt, besteht die Bitübertragungsschicht aus zwei Teilschichten: PLCP und PMD. Die PLCP-Teilschicht stellt ein Signal zur Überwachung des Mediums zur Verfügung und bietet einen einheitlichen PHY-Dienstzugangspunkt unabhängig von der Übertragungstechnik an. Die PMD umfasst die Datenmodulation und Signalcodierungs- und Decodierungsschemata.

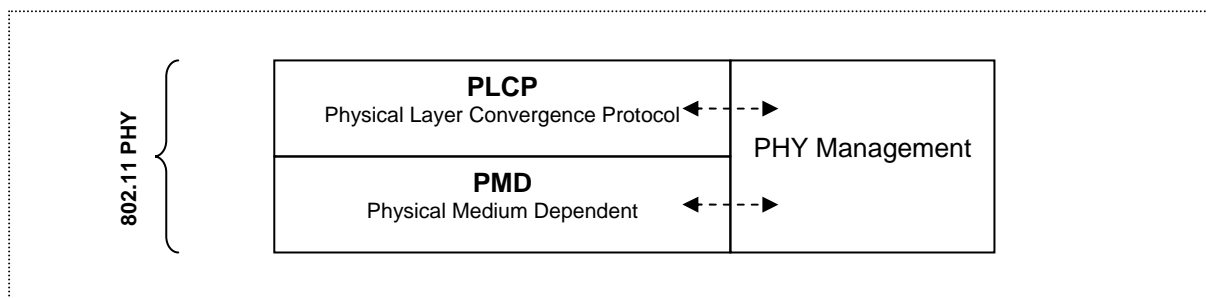


Abbildung 2-11 PHY-Architektur in IEEE 802.11 [IEEE811]

Die in IEEE 802.11b definierte Übertragungstechnik FHSS und DSSS operieren im lizenzfreien elektromagnetischen Band mit 2.4GHz (2.4 bis 2.483GHz) Frequenz (RF). Dieses Band ist für ISM (industrielle, wissenschaftliche und medizinische)-Anwendungen reserviert und benötigt damit keine amtliche Genehmigung. Die Kerneigenschaft dieser SS (Spread Spektrum)-Technologie ist: sie benutzt einen größeren Frequenzbereich, als zur eigentlichen Datenübertragung benötigt wird. Diese Signalspreizung erfolgt beim Senden durch einen Code, der dem Empfänger bekannt sein muss, um die Daten entschlüsseln zu können [nemg01].

In IEEE 802.11b DSSS wird das Frequenzband(von 2.4G Hz bis 2.483 GHz) in drei nicht überlappenden bzw. sieben überlappenden Kanälen unterteilt (In Europa, bis auf Frankreich und Spanien), die in Abbildung 2-12 und Abbildung 2-13 veranschaulicht sind.

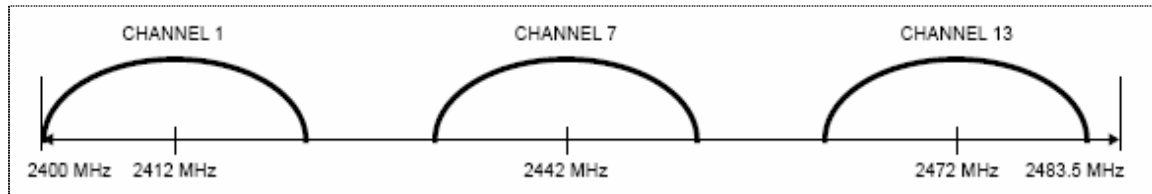


Abbildung 2-12 Europäische Kanalselektion- nicht überlappend [IEEE811b]

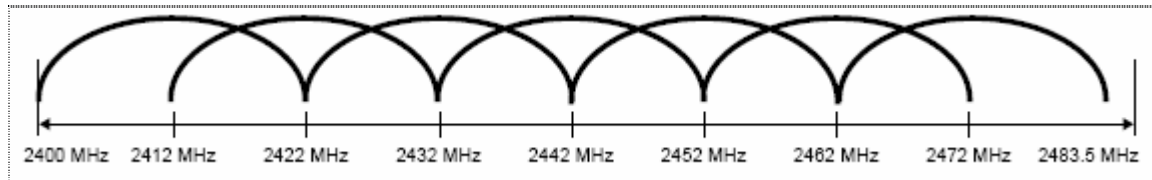


Abbildung 2-13 Europäische Kanalselektion – überlappend [IEEE811b]

In jedem DSSS-System wird ein Erkennungscode (PN-Code, Pseudo-Noise-Code) verwendet, damit die Signale von verschiedenen Systemen nicht durcheinander gestört werden. Unterschiedliche DSSS-Systeme dürfen gleiche Codes benutzen, nur wenn deren Sendebereiche sich nicht überschneiden. Wenn die Sendebereiche von mehreren DSSS-Systemen sich überschneiden, kann nur die nicht-überlappende Kanalselektion eingesetzt werden (Abbildung 2-12), da die gleichen Codes von unterschiedlichen DSSS-Systemen sonst zur gegenseitigen Störung kommen können.

Auf verschiedenen Übertragungsraten werden unterschiedliche Technologien für Signalmodulation eingesetzt, die Wirkung für Signalausbreitung haben. Wegen dieser technischen Eigenschaft ist die Reichweite des Signals auf höherer Übertragungsrate kürzer als es auf niedriger Übertragungsrate. Abbildung 2-14 zeigt diese Abhängigkeit.

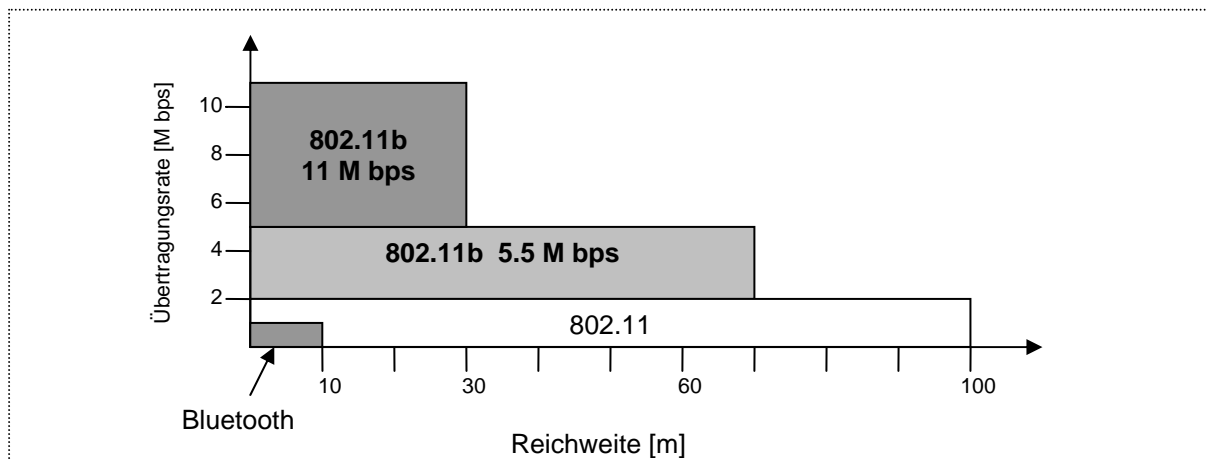


Abbildung 2-14 Übertragungsraten und Reichweiten in IEEE 802.11b [nemg01]

2.5.3 Netzwerk-Topologien

Während metallische oder optische Kabel in kabelgebundenen LAN zur Datenübertragung dienen, wird der Funkkanal mit Hilfe von elektromagnetischen Wellen als Übertragungsmedium in Wireless LAN verwendet. Eine Zelle in IEEE 802.11 ist eine räumliche Region, innerhalb der Stationen (Netzwerkteilnehmer) unter Berücksichtigung ihrer jeweiligen Reichweiten untereinander

kommunizieren können[das drahtlose Ethernet]. Der Begriff „Zelle“ wird in IEEE 802.11 als „BSS“ (Basic Service Set) bezeichnet. In einem BSS ist es nicht erforderlich, dass zwei Netzwerkteilnehmer direkt kommunizieren können. D.h., die Reichweiten zwei Stationen müssen nicht überlappend sein. Des Weiteren darf ein Teilnehmer zu mehreren BSS gehören. In IEEE 802.11 sind zwei wichtige Netzwerk-Topologien definiert, das Ad-hoc- und Infrastrukturnetzwerk.

2.5.3.1 Infrastrukturnetzwerk

In Infrastrukturnetzwerk findet die Kommunikation grundsätzlich über APs (Access Points, Zugangspunkte) statt. In einem BSS befinden sich ein AP und beliebige Mobilstationen. Der AP ist ein Koordinator für Medienzugriffssteuerung und die Netzwerkverwaltung. D.h. alle Mobilstationen liegen innerhalb der Reichweite des APs und sind die Kommunikationen durch den AP geregelt. Es muss nicht unbedingt sein, dass die übertragenen Nachrichten über diesem AP übermittelt werden. Die Sendstation darf die Nachrichten auch direkt an die Empfangsstation senden, aber muss ein Medienzugriffsrecht vom AP erhalten. Ein Infrastrukturnetzwerk darf beliebige BSS und ein DS (Distributed System, Verteilungssystem) haben, wie die Abbildung 2-15 zeigt. Dann entspricht es einem ESS (Extended Service Set). Das DS kann ein Kabelgebundenes Netzwerk sein, wie z.B. IEEE 802.3 Ethernet oder wiederum ein drahtloses System sein, z.B. Satellitensystem oder Richtfunksystem. Ein Infrastrukturnetzwerk aus zwei BSS und einem Richtfunksystem wird in Abbildung 2-16 dargestellt. Der Nachrichtentransport zwischen verschiedenen BSS oder zwischen BSS und einem weiteren Kabelgebundenen Netzwerk wird durch die IEEE 802.1-Brücke (Bridging) ermöglicht (vgl. Abbildung 2-2 LAN Standards - IEEE 802.x Familie [IEEE 811]). Normalerweise funktioniert ein AP auch als eine Brücke.

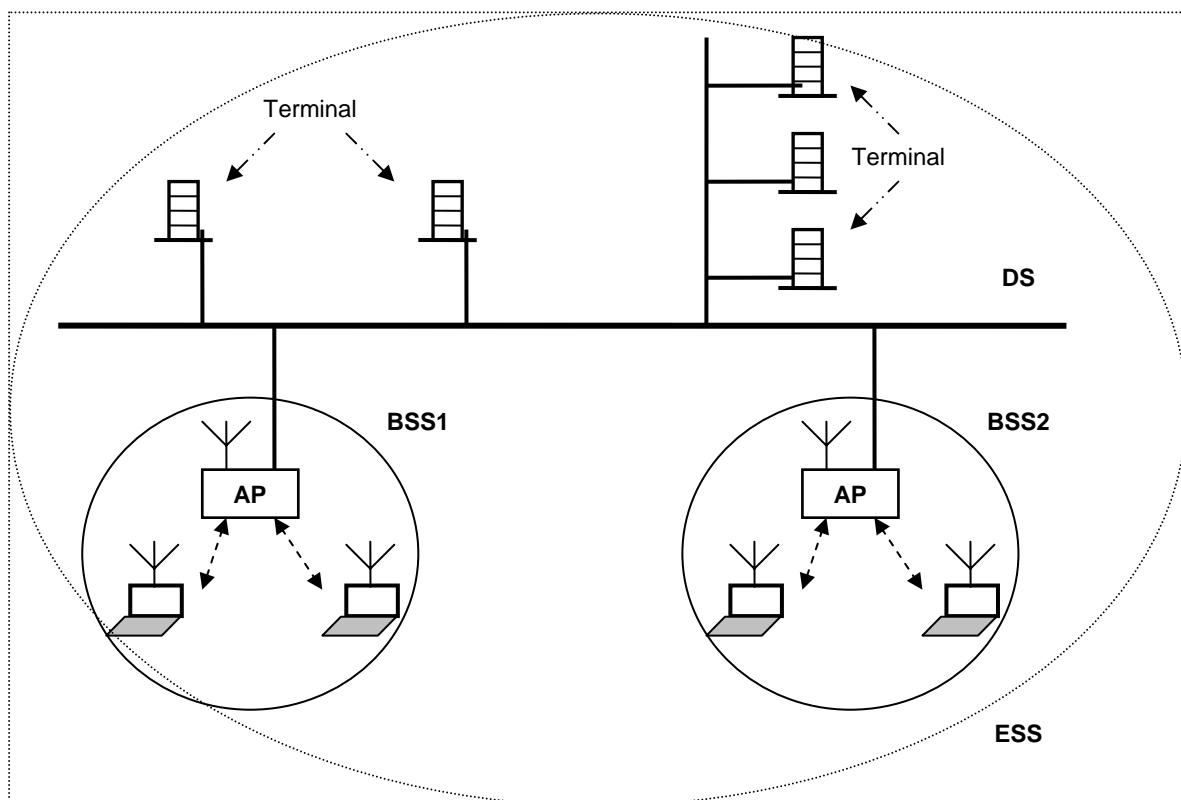


Abbildung 2-15 Infrastrukturnetzwerk mit Ethernet

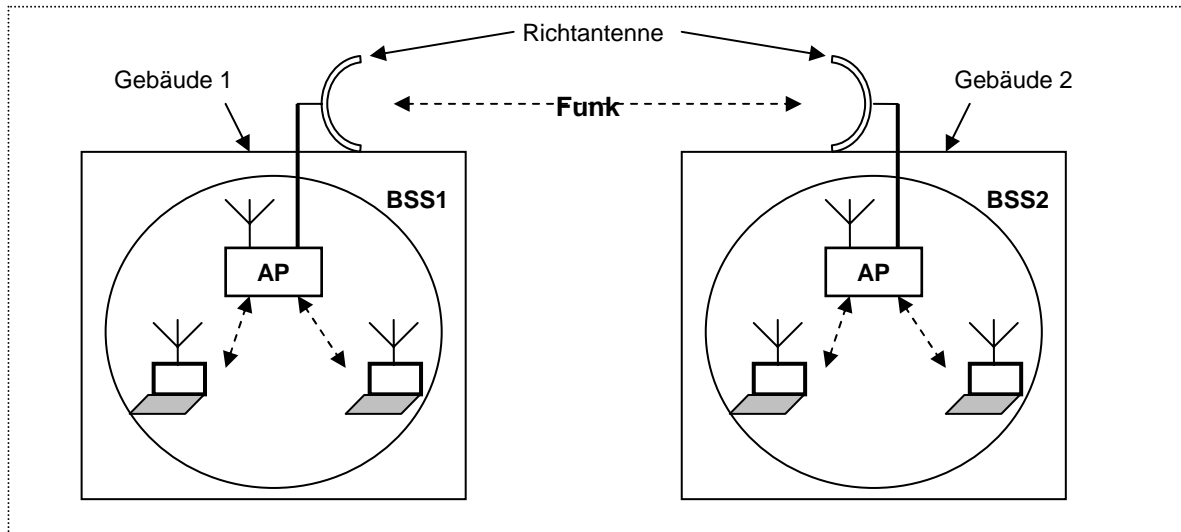


Abbildung 2-16 Infrastrukturnetzwerk mit Richtfunksystem

2.5.3.2 Ad-hoc-Netzwerk

Im Gegensatz zum Infrastrukturnetzwerk gibt es bei einem Ad-hoc-Netzwerk, auch Punkt-zu-Punkt-Netzwerk genannt, keinen Zugangspunkt (AP). Ein Ad-hoc-Netzwerk besteht nur aus einem einzigen BSS. Im IEEE 802.11-Standard wird ein BSS in Ad-hoc-Netzwerktopologie als IBSS bezeichnet (Independent BSS, unabhängiges BSS). Alle Netzteilnehmer in einem IBSS können miteinander direkt kommunizieren, wenn sie gegenseitig in Reichweiten liegen, wie die Abbildung 2-17 zeigt. Broadcasting wird in Ad-hoc-Netzwerk unterstützt. Ein Netzwerkteilnehmer darf seine Nachricht über eine Broadcast-Adresse abschicken. Alle anderen Mobilstationen können diese Nachricht empfangen, wenn sie in Reichweite des Senders liegen.

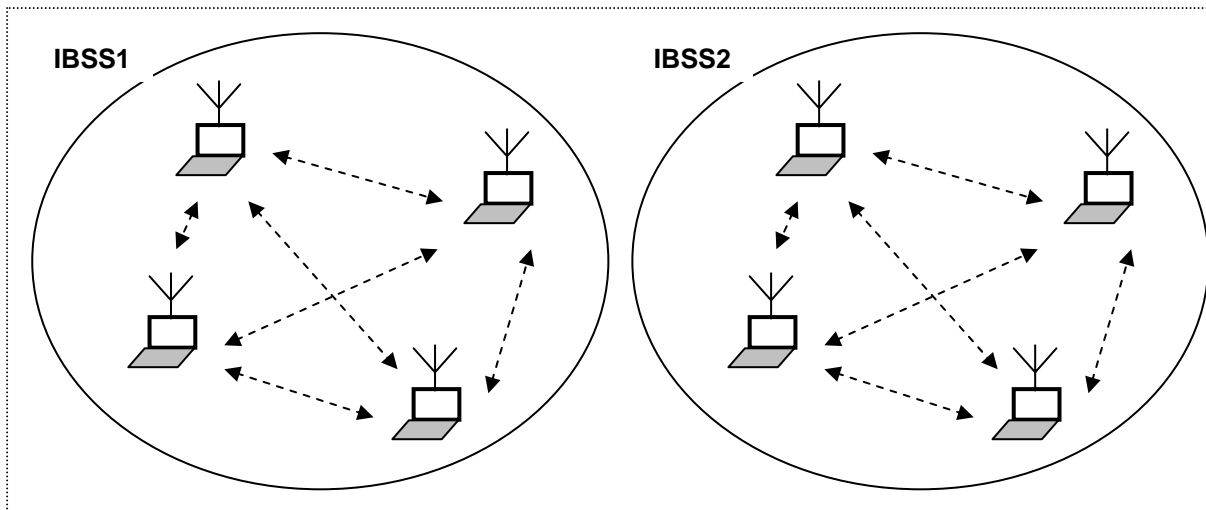


Abbildung 2-17 Ad-hoc-Netzwerke

2.5.4 Netzwerkverwaltungen

In Infrastrukturnetzwerk spielt der Zugangspunkt eine zentrale Rolle für die Verwaltung des Netzwerks. Der AP bietet einen Dienst zur Anmeldung und Abmeldung von Mobilgeräten. Der AP

sendet periodisch einen Beacon-Rahmen (Management-Rahmen) per Broadcast aus. Der Beacon-Rahmen beinhaltet die wichtigen Informationen, wie z.B. BSS-Adresse, unterstützte Übertragungsrate usw. Wenn sich ein Mobilgerät an ein Netzwerk anmelden möchte, muss es für eine gewisse Zeit warten, um einen Beacon-Rahmen zu empfangen. Danach meldet es sich mit der im Beacon-Rahmen enthaltenen BSS-Adresse am AP an. Wenn das Mobilgerät mehrere Beacon-Rahmen aus verschiedenen BSS empfangen hat, wählt es einen günstigen AP aus, z.B. den mit der höheren Übertragungsrate. Stattdessen gibt es noch eine einfache Lösung, dass das Mobilgerät aktiv einen Anfrage-Rahmen per Broadcast absenden darf. Die APs, die diesen Anfrage-Rahmen empfangen haben, antworten mit einem Beacon-Rahmen. Danach kann sich das Mobilgerät entscheiden, an welchen AP es sich anmelden soll. Im Gegensatz zum Infrastrukturnetzwerk ist die Verwaltung eines Ad-hoc-Netzwerks eine völlig verteilte Lösung, da kein AP im Netzwerk existiert. Alle Netzwerkteilnehmer senden den Verwaltungsrahmen, geregelt jeweils durch einen zufälligen Timer.

2.5.5 Verschlüsselte Datenübertragung

Die Sicherheit im Bereich von Internet und Telekommunikation spielt eine immer größere Rolle, da die über das Netzwerk transportierten Nachrichten von unerwünschten Personen abgehört oder sogar manipuliert werden können. Im lokalen Netzwerk wird ein gemeinsam benutztes Medium zur Datenübertragung benutzt. Die auf dem Medium laufenden Daten kann jeder Netzwerkteilnehmer abhören. Besonders in Wireless LAN, ist der Übertragungskanal ein offenes Medium. Die gängigen Sicherheitsstrategien, die auf höheren Schichten funktionieren, können diese besondere Anforderung nicht erfüllen.

Also Option spezifiziert das IEEE 802.11 Standard das WEP-Verfahren (Wired Equivalent Privacy), das die verschlüsselte Datenübertragung auf der niedrigen MAC-Ebene ermöglicht. Das Grundprinzip des WEP sieht wie die Abbildung 2-18 aus.

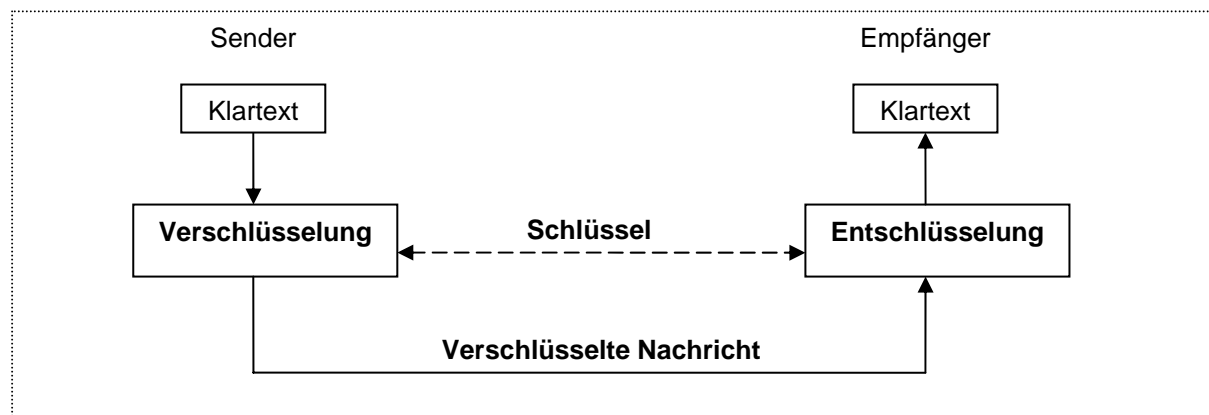


Abbildung 2-18 Grundprinzip des WEP in IEEE 802.11

In WEP wird ein gemeinsamer Schlüssel (Shared Key) zur Verschlüsselung und Entschlüsselung benutzt. Aus dem Schlüssel und einem IV (Initialisierungsvektor) wird ein Bytestrom mit Hilfe von einem Zufallzahlgenerator erzeugt. Danach wird der Klartext mit dem Bytestrom durch „XOR“ verknüpft [IEEE811]. WEP ist ein recht einfaches Verfahren. Wie dieser Schlüssel an die Benutzer verteilt wird, bleibt in IEEE 802.11 noch offen.

2.6 Drahtlose- vs. Kabelgebundene LANs

Die Eigenschaften der drahtlosen Netze unterscheiden sich wesentlich von klassischen kabelgebundenen Netzen. In kabelgebundenen Netzen sind die Übertragungsleitungen normalerweise Twisted-Pair oder Glasfaser. Die Signale breiten sich entlang des Leiters aus, d.h. der Leiter gibt die Art der räumlichen Signalausbreitung vor. Die gewissen Abstrahlungseffekte sind gut vernachlässigbar, und die Dämpfung des Signals ist auch sehr niedrig. Wegen der Unzuverlässigkeit des Funkkanals ist die Verlust- und Fehlerrate der Datenübertragung in Wireless LAN höher als in kabelgebundenen Netze, da die Signalausbreitung in Funkkanal von vielen Faktoren abhängt: Strahlung, Antennentechnik, Gegenstände usw. Die Mobilität der Netzwerkteilnehmer verursacht eine dynamische Änderung von Netzwerktopologien und -Partitionen. Die Hauptunterschiede zwischen IEEE 802.11b Wireless LAN und IEEE 802.3 Ethernet sind in Tabelle 2-4 aufgelistet.

| | IEEE 802.11b Wireless LAN | IEEE 802.3 Ethernet |
|---------------------------------------|---|-----------------------------------|
| Übertragungsmedium | Funkkanal (Elektromagnetische Welle) | Kabel (Koaxial, Glasfaser...) |
| Übertragungsrate | bis zu 11 MBit/s | bis zu 1000 MBit/s |
| Verlust- und Fehlerrate | höher als bei Ethernet | gering |
| Entfernung zwischen zwei Stationen | 30 Meter bei 11 MBit/s, maximal 100 Meter bei 2 MBit/s | max. 2500 Meter mit 4 Verstärkern |
| Geschwindigkeit der Signalausbreitung | 300 T KM/s (Lichtgeschwindigkeit) | ca. 200 T KM/s |
| Topologien | Ad-hoc und Infrastruktur | Bus |
| Anzahl der Teilnehmern | hoch dynamisch | geplant |
| Sicherheit | verschlüsselte Datenübertragung | nicht verschlüsselt |
| Max. Größe der Nutzdaten | 2312 Bytes | 1500 Bytes |
| Min. Größe des Rahmens | Keine Beschränkung | 64 Bytes |

Tabelle 2-4 Unterschiede zwischen IEEE 802.11b und IEEE 802.3

2.7 Das NET (Network Emulation Testbed)

Das NET (Network Emulation Testbed) der Abteilung Verteilte Systeme an der Universität Stuttgart ist ein flexibel vernetzter Linux PC-Cluster mit 64 Knoten, in dem beliebige Netztopologien und deren Eigenschaften nachbildet werden können. Damit bietet das NET eine ideale Analyseumgebung für verteilte Anwendungen und Netzprotokolle [hero02].

2.7.1 Hardwareumgebung

Die Hardwareumgebung des NETs sieht wie die Abbildung 2-18 aus.

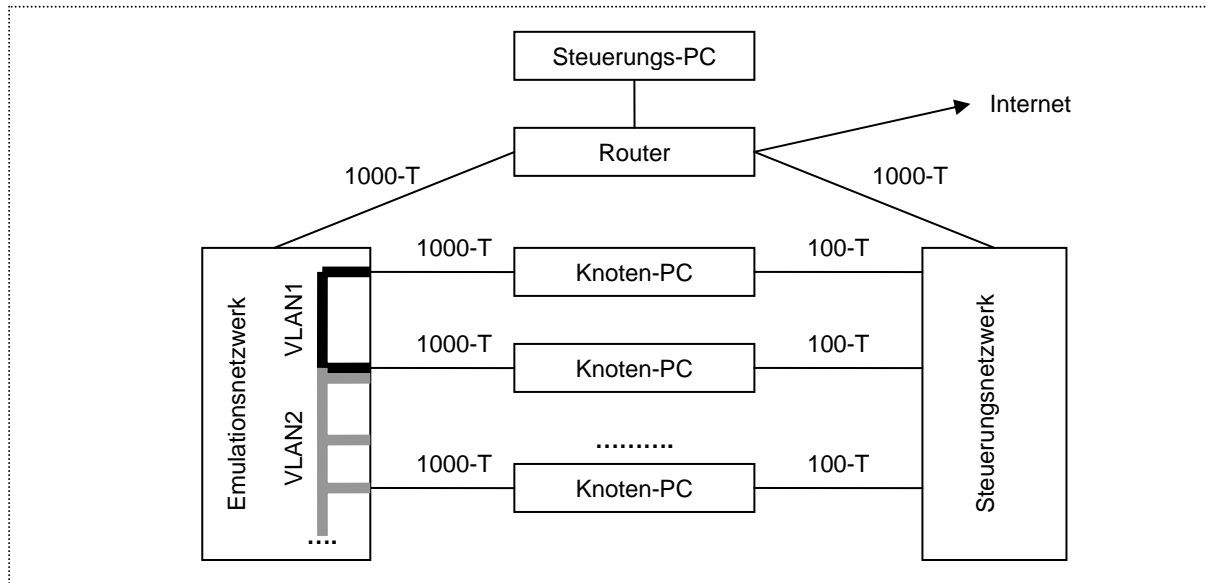


Abbildung 2-19 Hardwareumgebung des NETs [hero02]

Die Knoten-PCs und ein Giga-Bit-Ethernet-Switch (Übertragungsrate 1000 MBit/s), mit dem die Knoten-PCs angebunden sind, bilden ein Emulationsnetzwerk. Durch die VLAN-Technologie (Virtuelle LANs) kann das gesamte Emulationsnetzwerk auf vielen Teilnetzwerken geteilt werden, die beliebige Netzwerktopologien haben dürfen. Ein Knoten-PC darf auch zu mehreren VLANs definiert werden, da die virtuelle Verbindungen und deren Adressierungen durch das Betriebssystem des Knoten-PCs verwaltet werden können. Neben dem Emulationsnetzwerk ist ein Steuerungsnetzwerk vorhanden, das einen Steuerungs-PC und allen Knoten-PC durch ein Fast-Ethernet-Switch (Übertragungsrate 100 MBit/s) verbindet. Die zwei Netzwerke sind miteinander physisch getrennt, d.h. der Datenverkehr in Steuerungsnetzwerk stört die Datenkommunikationen in Emulationsnetzwerk nicht. Der Steuerungs-PC dient zur Verwaltung von Knoten-PC und den Szenarien des Emulationsnetzwerks, sowie zur Emulationsverwaltung, wie z.B. die dynamischen Änderungen von Bandbreit.

2.7.2 Emulationsstrategie

Auf allen Knoten (Pentium 4 Prozessor mit 2.4 GHz) läuft Linux als Betriebssystem (Redhat Linux 7.3 mit Kernel-Version 2.4). Für die Netzwerke emulationen ist eine Emulationsschicht in die Protokollstapel jedes Knoten-PCs gesondert eingefügt. Die Abbildung 2-20 zeigt, wo die Emulationsschicht in den Protokollstapel eingereift. [hero02]

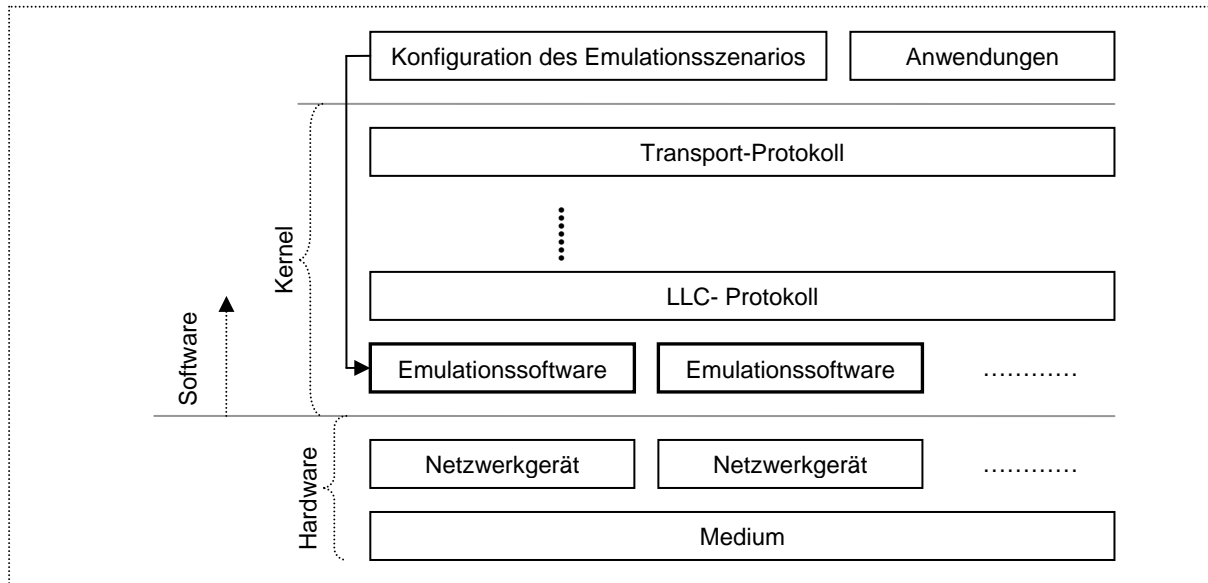


Abbildung 2-20 Protokollstruktur der Emulation für MAC-Protokoll

Die Emulationssoftware funktioniert als Modul in Linux-Kernel. Die Änderung der Modulparameter während Laufzeit ist möglich. Diese Eigenschaft ist besonders nützlich für MANET-Emulation, da die Netzwerkeigenschaften sich ständig ändern können (z.B. Fehlerrate). Die dynamischen Änderungen von Parametern können sowohl manuell als auch per „Konfigurations-Daemon“ realisiert werden. Die Änderungen der Parameter werden vom Steuerungs-PC per UDP über das Verwaltungsnetzwerk zu allen in der Emulation beteiligten Knoten-PC geschickt. Nachdem der „Konfigurations-Daemon“ die Nachricht analysiert hat, greift er auf die Emulationssoftware (Linux-Modul) zu. So wird die dynamische Änderung von Parametern in NET unterstützt.

Kapitel 3 Analyse des Medienzugriffsverfahrens in IEEE 802.11

Wie in kabelgebundenen lokalen Netzen, wird das Medium (Funkkanal) zur Datenübertragung in Wireless LAN auch für alle Netzwerkteilnehmer freigegeben und gemeinsam benutzt. Um Daten im Netzwerk reibungslos zu übertragen, ist exklusiver Zugriff auf das Übertragungsmedium erforderlich, die Medienzugriffe müssen also geregelt werden. Das in IEEE 802.11 beschriebene MAC-Protokoll ist eng mit den Vorgaben aus IEEE 802.3 Ethernet verwandt (vgl. Abbildung 2-2). Die grundlegenden Aufgaben der MAC-Schicht in IEEE 802.11 sind die Steuerung des Medienzugriffs, Behandlung der Kollisionen und Fehlererkennung. Zusätzlich sind die Fragmentierung von langen Nachrichten und die verschlüsselte Datenübertragung auch in IEEE 802.11-MAC spezifiziert.

In Abschnitt 2.5 wurde kurz erläutert, wie mehrere BSS in Infrastruktur-Modus bzw. IBSS in Ad-hoc-Modus kooperieren können. Mehrere BSS bzw. IBSS können entweder mit Hilfe von Raummultiplex (durch genügend Abstand) oder durch Verwendung unterschiedlicher Trägerfrequenz (Frequenzmultiplex) gebildet werden. Das in IEEE 802.11 beschriebene MAC-Protokoll spezifiziert die wichtigen Medienzugriffsmethoden, die unabhängig von Bitübertragungsschicht funktionieren, d.h. spezifisch in einem BSS bzw. IBSS.

3.1 Elementare Medienzugriffsmethoden

Zur Steuerung der Medienzugriffe sind zwei elementare Zugriffsmethoden in IEEE 802.11 spezifiziert: DCF (Distributed Coordination Function) und PCF (Point Coordination Function).

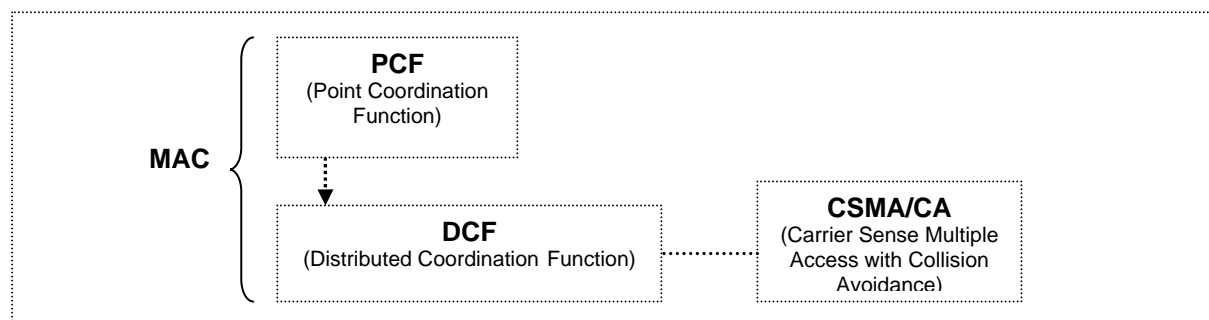


Abbildung 3-1 MAC-Architektur in IEEE 802.11 [IEEE811]

DCF beschreibt die grundlegende Medienzugriffsmethode des IEEE 802.11 und folgt dem CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) Algorithmus, wie die Abbildung 3-1 zeigt. In DCF ist die Steuerung der Medienzugriffe auf die Netzwerkteilnehmer völlig verteilt. Im Rahmenkopf jedes MAC-Rahmens in IEEE 802.11 wird die Information der voraussichtlichen Übertragungsdauer eingefügt. Nach der Analyse des angekommenen Rahmens wissen die Mobilstationen, wann sie Nachrichten auf den Kanal senden dürfen, also die Kanalbelegungsinformation wird ermittelt. Als Option ist die PCF in IEEE 802.11 beschrieben, die auf DCF aufbaut. Die grundlegenden Regeln in DCF werden auch in PCF verwendet, z.B. der Mechanismus zur Ermittlung der Kanalbelegungsinformationen. Im Gegensatz zur DCF wird ein PC (Point Coordinator) in PCF eingesetzt, der zur Regelung der gesamten Kommunikationen dient. Der PC entscheidet, wann und welche Mobilstation Daten senden darf. Da ein zentraler Verwalter benötigt wird, ist PCF nur einsetzbar im Infrastruktur-Modus. In Infrastruktur-Netzwerken wird normalerweise der AP als PC benutzt.

3.2 CSMA/CA

Das CSMA/CA ist sehr eng mit dem CSMA/CD verwandt, das in Ethernet benutzt wird (vgl. Abschnitt 2.4.2.2). Es ermöglicht, dass mehrere Netzwerkteilnehmer einen gemeinsamen Kanal zur Datenübertragung nutzen (MA, Multiple Access). Bevor sie Daten senden können, müssen die Sendstationen den Kanalzustand überprüfen. Wenn der Kanal frei ist, dürfen sie ihre Daten absenden. Wenn dies nicht der Fall ist, wird der Sendvorgang sich auf einen späteren Zeitpunkt verzögern. Das entspricht dem CS (Carrier Sense, Trägererkennung).

Bei CSMA/CD-Verfahren, sendet eine Sendstation eine Nachricht und hört den Kanal während des Sendens ab. Wenn sie Signalgeräusch gehört hat, also eine Kollision erkannt wurde, bricht die Sendstation ihren Sendvorgang sofort ab, sodass die unnötige Kanalbelegung vermieden werden kann. Wegen der technischen Beschränkungen in Wireless LAN, während des Sendens einer Nachricht, kann die Sendstation den Kanal nicht gleichzeitig abhören. D.h. wenn Kollisionen auftreten würden, werden sie nicht während Sendens bei der Senderstation erkannt. Im Gegensatz zum CSMA/CD, sendet eine Sendstation ihre Daten immer weiter, bis der ganze Rahmen fertig abgesendet worden ist, wie in Abbildung 3-2 gezeigt.

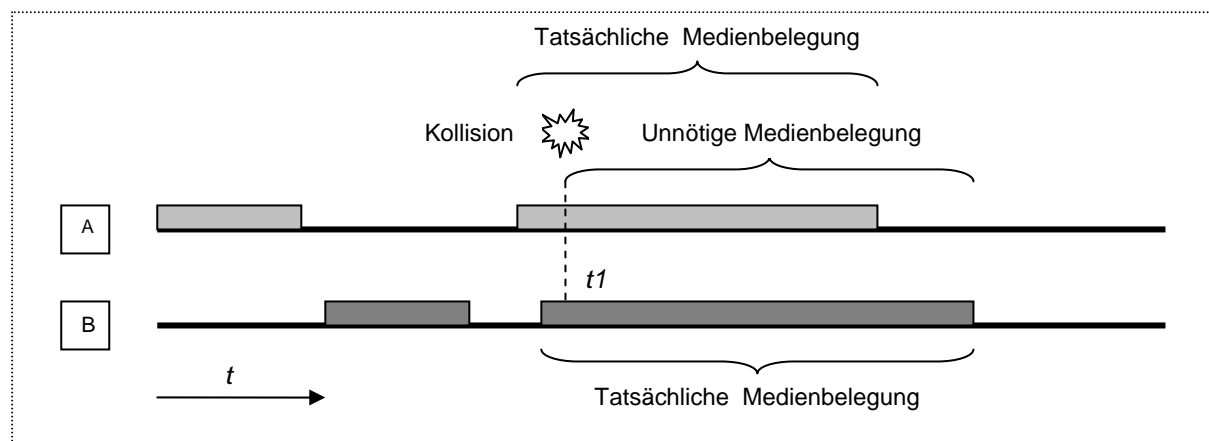


Abbildung 3-2 Sendstationen senden ihre Daten immer weiter, obwohl Kollision auftritt.

Eine Kollision tritt zum Zeitpunkt t_1 auf. Da die Sendstationen A und B diese Kollision nicht erkennen können, senden sie ihre Daten immer weiter, sodass der Kanal für eine lange Zeit unnötig belegt wird und eine schlechte Kanalausnutzung folgt. Deswegen sollen Kollisionen möglichst in Wireless LAN vermeiden werden.

Das in IEEE 802.11 spezifizierte CSMA/CA Verfahren erfüllt diese besondere Anforderung in Wireless LAN. CA (Collision Avoidance) bedeutet, dass die Kollisionsgefahr möglichst vor dem Senden vermieden werden muss, damit die Kollisionswahrscheinlichkeit reduziert werden kann. Es gibt aber keine Garantie, dass die Kollisionen überhaupt nicht vorkommen.

Die Trägererkennung (CS, Carrier Sense) in CSMA/CA besteht aus zwei Mechanismen, die physische und virtuelle Erkennung. Die physische Trägererkennung erfolgt nur, wenn die Mobilstation nicht sendet. Wenn die Antenne der Mobilstation Signale detektiert, wird das Medium als „besetzt“ erkannt. In CSMA/CA verwaltet jede Mobilstation einen NAV (Network Allocation Vector)-Timer. Der NAV-Wert gibt die Zeitdauer an, wie lange das Medium voraussichtlich besetzt sein wird. Der Kopf jedes MAC-Rahmens in IEEE 802.11 enthält eine berechnete Zeitdauer, die der voraussichtlichen nachfolgenden Übertragungsdauer entspricht (siehe Abbildung 4-4 Rahmenformat). Wenn eine Mobilstation einen Rahmen senden möchte, muss sie die Dauerinformation in den MAC-Kopf

eintragen. Die NAV-Werte aller anderen Mobilstationen werden durch die Zeitdauer aktualisiert, die sich in jedem mitgehörten Rahmen befindet. Eine Datenübertragung wird nicht unternommen, wenn die NAV-Wert der Station, der Nachricht senden möchte, nicht Null ist. Dieser Mechanismus wird auch virtuelle Trägererkennung genannt.

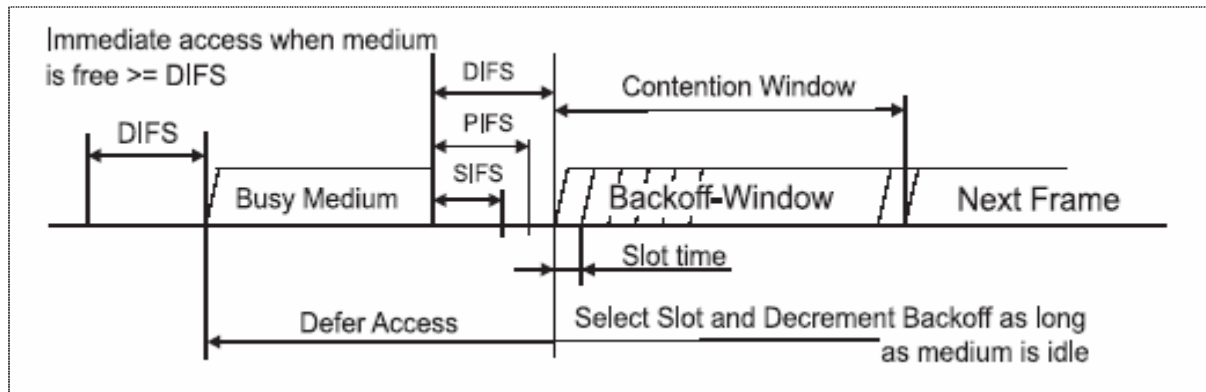


Abbildung 3-3 Zugriffsmechanismus in CSMA/CA [IEEE811]

Wenn eine Mobilstation Nachricht senden möchte, muss sie zunächst ihren Kanalzustand (physisch und virtuell) überprüfen. Die Kanalzugriffe unterscheiden sich in folgenden Fällen:

1. Aus der Sicht der Sendstation, wenn der Kanal für eine DIFS-Zeit (DCF Interframe Space, wird in folgenden Kapitel genau erklärt) frei ist, d.h. ist ihr NAV-Wert Null (virtuelle Trägererkennung) und hat die physische Trägererkennung auch kein Signal auf Kanal für länger als DIFS-Zeit detektiert, darf sie den Kanal sofort zugreifen, wie die Abbildung 3-3 zeigt. Also, ein schneller Kanalzugriff ist erlaubt, da der Kanal seit einer gewissen Zeit frei ist und die Wahrscheinlichkeit, dass mehrere Stationen gleichzeitig senden möchten, sehr niedrig ist.
2. Wenn ihr NAV-Wert Null ist (virtuelle Trägererkennung), aber die physische Trägererkennung während der DIFS-Zeit Signale detektiert, wird der Sendvorgang auf einen späteren Zeitpunkt verzögert, d.h. es ist gerade eine andere Übertragung im Gang. Am Ende jener Übertragung muss die Sendstation den Kanal noch mal für eine DIFS-Zeit überwachen und danach wird ein Zugriffswettbewerb-Prozess folgen, da die anderen Mobilstationen nach der vorherigen Kanalbelegung mit großer Wahrscheinlichkeit mit dem Senden beginnen werden. Während des Wettbewerbs muss sie den Kanal auch noch abhören. Wurde Signal detektiert, hält der Wettbewerb-Vorgang sofort an und wird der gesamte Sendprozess sich verzögern. Wenn es nicht der Fall ist, sendet die Sendstation ihre Daten sofort auf Funkkanal, d.h. Kanal ist während des Wettbewerbs frei, hat die Sendstation das Zugriffsrecht bekommen. Wie in CSMA/CD wird die Wartezeit für Wettbewerb durch die BackOff-Funktion generiert.
3. Wenn ihre NAV-Wert nicht Null ist, darf keine Datenübertragung starten, d.h. der Kanal ist gesperrt, solange der NAV-Timer noch nicht abgelaufen ist. Wenn ihr NAV-Timer abgelaufen ist, wird der Sendvorgang geweckt. Wie in Fall 2, wartet die Sendstation eine DIFS-Zeit ab und danach ruft den Wettbewerb-Prozess auf. Wenn der Wettbewerb-Prozess erfolgreich abgeschlossen würde, sendet die Sendstation ihre Nachricht ab.

3.2.1 Basismechanismus

Die Signalausbreitung in Funkkanal ist abhängig von vielen Faktoren, Strahlung, Antenne, Gegenstände usw. Daraus können z.B. verkürzte oder unsymmetrische Reichweiten folgen. In diesem Kapitel vereinbaren wir eine ideale Umgebung, wie in Abbildung 3-4 gezeigt.

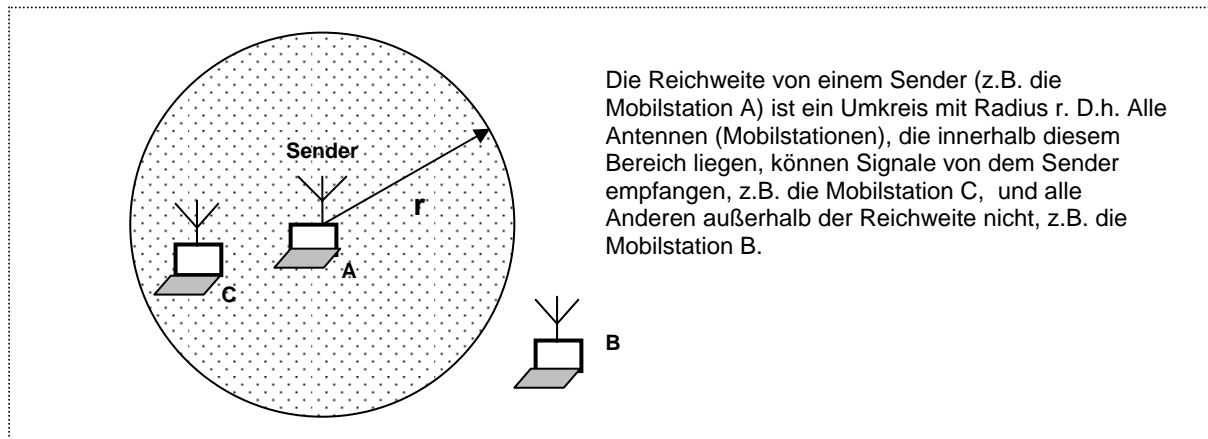


Abbildung 3-4 die ideale Umgebung

Als Beispiel, betrachten wir diese Situation, die in Abbildung 3-5 dargestellt ist.

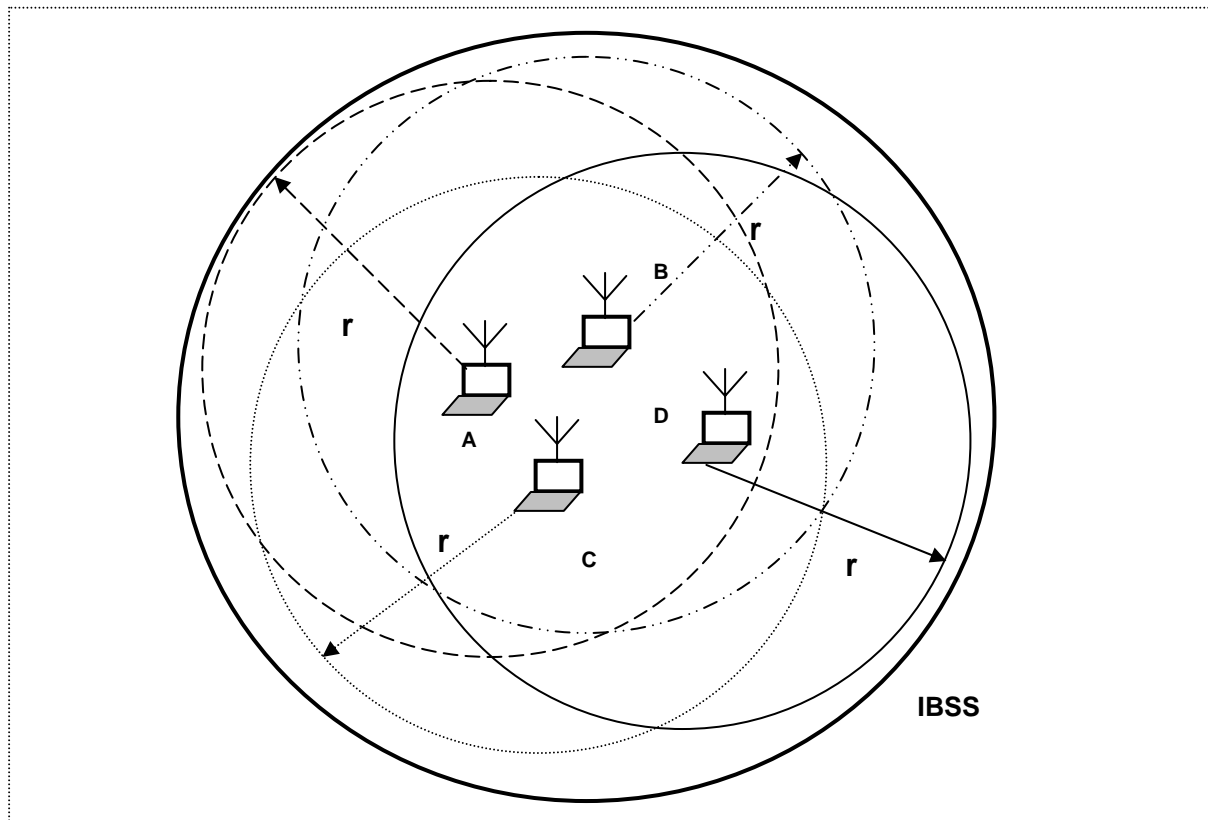


Abbildung 3-5 Mobilstationen liegen innerhalb eines überlappenden Bereiches

1. Alle Mobilstationen kommunizieren mit einander in einem Ad-hoc-Netzwerk.
2. Die Reichweiten von Stationen A, B, C und D sind überlappend. Und liegen die vier Mobilstationen innerhalb dieses überlappenden Bereichs. D.h. das von einer Station abgesendete Signal können alle anderen Stationen empfangen.

Die Abbildung 3-6 veranschaulicht in einem Zeitdiagramm den Basismechanismus von CSMA/CA. Die Signalausbreitungsverzögerungen werden in dieser Abbildung ignoriert.

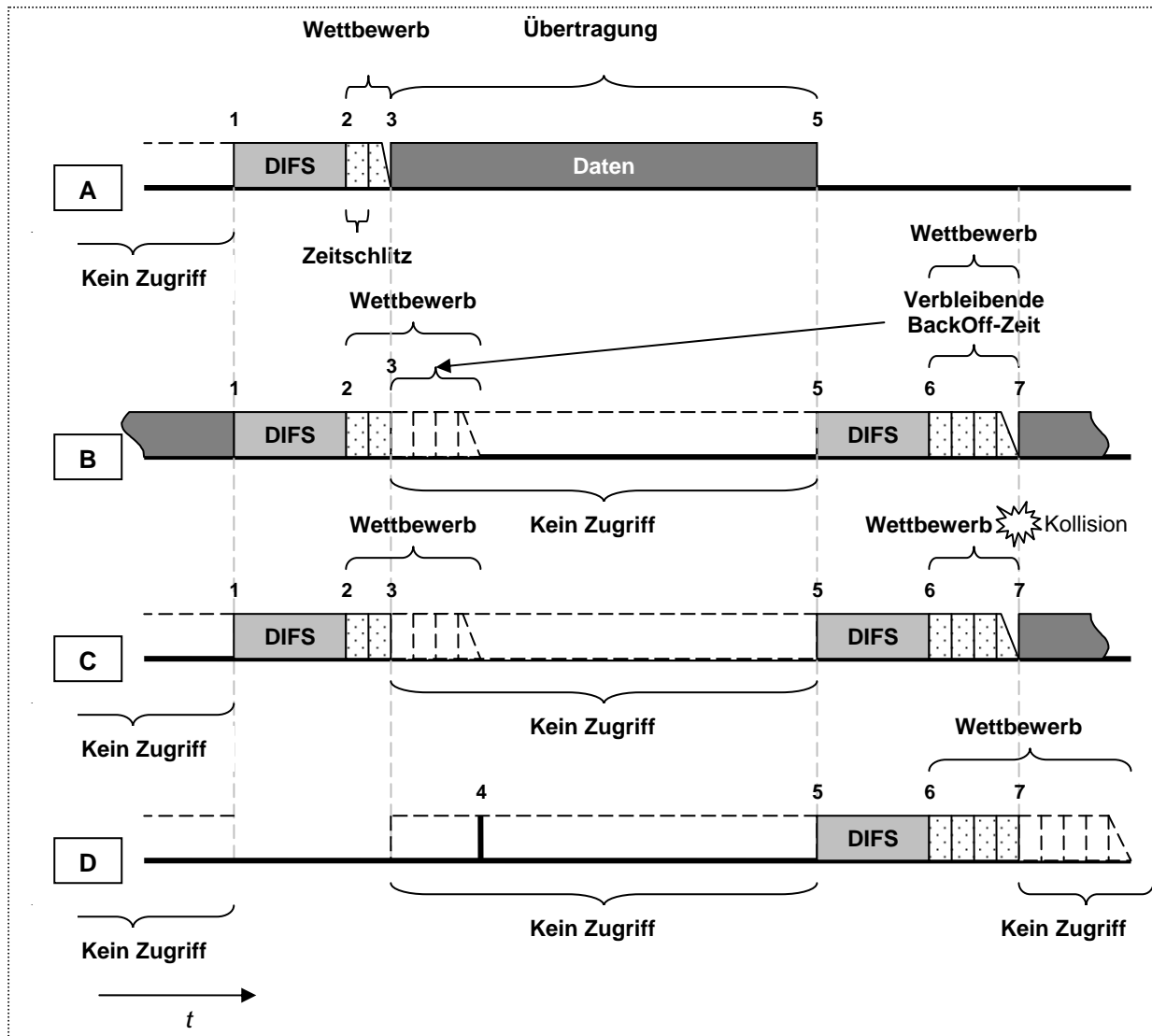


Abbildung 3-6 Zeitdiagramm des Basismechanismus von CSMA/CA

1. Nach einer Übertragung von Station B, möchte B noch weitere Nachricht senden. Da die NAV-Timer von Stationen A und C abgelaufen sind, hören sie zuerst den Kanal für eine DIFS-Zeit ab. B möchte gleichzeitig auch ein Rahmen senden, hört sie den Kanal ab und wartet eine DIFS-Zeit ab. (DIFS wird später erklärt.)
2. Für die DIFS-Zeit haben die Stationen A, B und C keine Signale detektiert, gehen die drei Stationen in den Wettbewerb-Zustand mit der jeweils ausgewählten BackOff-Wartezeit.
3. Der BackOff-Timer bei Station A ist zuerst abgelaufen, da ihre BackOff-Wartezeit (CW, Contention Window) kürzer ist. A fängt an, ihre Daten auf Medium zu senden. Unglücklicherweise haben die Stationen B und C durch ihre BackOff-Funktion größere Wartezeit ausgewählt, sind sie noch im Wettbewerb-Zustand. Die Bitübertragungsschicht der Station B und C hat erkannt, dass der Kanal wegen der Übertragung aus A nicht mehr frei ist. Die Wettbewerb-Prozesse bei B und C werden angehalten.
4. D möchte einen Rahmen senden. Leider ist der Kanal durch ihr physisches CS als besetzt erkannt, so darf die Übertragung nicht unternommen werden. Ihr Sendeprozess wird zum Ende der Kanalbelegung geplant.

5. Das Senden bei Station A ist beendet. Der Kanal aus der Sicht von Stationen B, C und D ist frei. Die Stationen, die senden möchten, hören den Kanal für eine DIFS-Zeit ab.
6. Während der DIFS-Zeit der Kanal ist als frei bei Stationen B, C und D erkannt. Die drei Stationen beginnen, das Zugriffsrecht auf den Übertragungskanal zu konkurrieren. Weil der BackOff-Timer bei Station B und C während des letzten Wettbewerbs angehalten wurde, braucht B keine neue Zufallszahl zu generieren. Ihr BackOff-Timer zählt weiter.
7. Die BackOff-Zeiten, die von B und C ausgewählt wurden, sind gleich. Die beiden Stationen senden ihre Nachrichten ab. Das verursacht offensichtlich eine Kollision. Das Senden bei Station D wird sich wegen der Kanalbelegung verzögern.

Der Mechanismus zur Vermeidung von Kollisionen in CSMA/CA verursacht eine längere Wartezeit als in CSMA/CD. In CSMA/CA muss jede Sendestation vor Datenübertragung für eine DIFS-Zeit den Kanal überwachen, während das Senden in CSMA/CD sofort beginnen kann, wenn der Kanal im Augenblick als frei erkannt wurde. Der Wettbewerb-Prozess in CSMA/CD wird gestartet, wenn eine Kollision aufgebaut wurde, da es tatsächlich andere Stationen gibt, die auf den Kanal zugreifen wollen. Im Gegensatz zum CSMA/CA, wenn die Station nach dem Sendewunsch und während der DIFS-Zeit Signale auf dem Kanal detektiert hat (physische Trägererkennung) oder durch NAV (virtuelle Trägererkennung) gesperrt ist, wird der Senderversuch angehalten und die BackOff-Funktion ausgeführt, sodass eine BackOff-Wartezeit generiert wird. Bei nächstem Senderversuch muss die Station den Kanal noch zusätzlich für die BackOff-Zeit überwachen. D.h. wenn die Sendestation wahrgenommen hat, dass es eine andere Station gibt, die eine Nachricht sendet, muss sie den jetzigen Senderversuch abbrechen und bei dem nächsten Senderversuch um ein Zugriffsrecht konkurrieren.

Wenn die Senderstation sich gerade im Zustand „Sperrung“ befindet, d.h. ihr NAV-Wert nicht Null ist, muss sie abwarten, bis der NAV-Timer angelaufen ist. Danach wartet sie eine DIFS-Zeit ab, generiert eine BackOff-Wartezeit und geht in den Wettbewerb-Zustand. Weil der NAV-Wert von einem vorbeilaufenden Rahmen aktualisiert ist, befindet sich eine andere Senderstation innerhalb der Reichweite und sendet sie mit großer Wahrscheinlichkeit noch weiter. Deswegen muss der Wettbewerb-Prozess ausgeführt werden.

Der wesentliche Unterschied zwischen CSMA/CA und CSMA/CD ist: In CSMA/CA wird der Zugriffswettbewerb grundsätzlich (wenn der Kanal entweder durch physisches CS oder durch virtuelles CS als besetzt erkannt wird) vor dem Rahmensenden ausgeführt, sodass die gleichzeitigen Zugriffe auf den Kanal am besten vermieden werden, damit die Wahrscheinlichkeit der Kollision verringert werden kann. Bei CSMA/CD wird dieser Zugriffswettbewerb erst nach der Erkennung einer Kollision aufgerufen.

3.2.2 CSMA/CA mit ACK

Obwohl Kollisionen in CSMA/CA durch NAV-Timer und den Wettbewerb um Kanalzugriffsrecht vermieden werden können, besteht noch Kollisionsgefahr, wenn mehrere Stationen das Zugriffsrecht gleichzeitig bekommen, wie die Abbildung 3-6 zeigt.

In Wireless LAN, merkt die Senderstation während des Sendens überhaupt nicht, wenn eine Kollision auftritt. Der Empfänger kann die Beschädigung eines Rahmens durch CRC-Check erkennen und wirft ihn weg, also liefert diesen Rahmen nicht aus. Die erneute Übertragung von dem beschädigten Rahmen wird durch eine höhere Schicht gesteuert.

Wie in Kapitel 2 erläutert, ist der Funkkanal im Vergleich mit elektronischen Leitungen unzuverlässig. Obwohl eine Kollision selten auftritt, kann ein Rahmen auch noch durch Übertragungsfehler beschädigt werden.

Das Mechanismus CSMA/CA mit ACK (Bestätigung) ist an diese Anforderungen abgestimmt und bietet eine sichere Übertragungen von Rahmen. Nach der Absendung eines Rahmens wartet die Senderstation auf der Bestätigung von dem Empfänger. Wenn die entsprechende Bestätigung innerhalb der Ablaufzeit (Timeout) nicht bei der Senderstation angekommen ist, erhöht sich die Anzahl des Sendversuchs bei Senderstation, d.h. wird das Wettbewerb-Fenster für die neue Übertragung exponentiell vergrößert (vgl. Tabelle 4-8 Werte der BackOff-Wartezeit in CSMA/CA). Nach einem Wettbewerb-Prozess wird dieser Daten-Rahmen erneut gesendet. Die maximale Anzahl der Sendversuche ist ein einstellbarer Parameter. Nach einer erfolgreichen Sendung wird das Wettbewerb-Fenster wieder auf CW_{min} zurückgesetzt (siehe Abbildung 4-7). Auf der Empfänger-Seite, nach dem erfolgreichen Datenempfang wartet der Empfänger eine SIFS-Zeit ab und hört den Kanal gleichzeitig ab. Wenn der Kanal für diese SIFS-Zeit frei ist, sendet sie eine Bestätigung (Acknowledgment, ACK) an den Sender zurück und liefert den Daten-Rahmen an eine höhere Schicht aus.

Die Abbildung 3-7 veranschaulicht mit einem Zeitdiagramm den CSMS/CA-Mechanismus mit ACK.

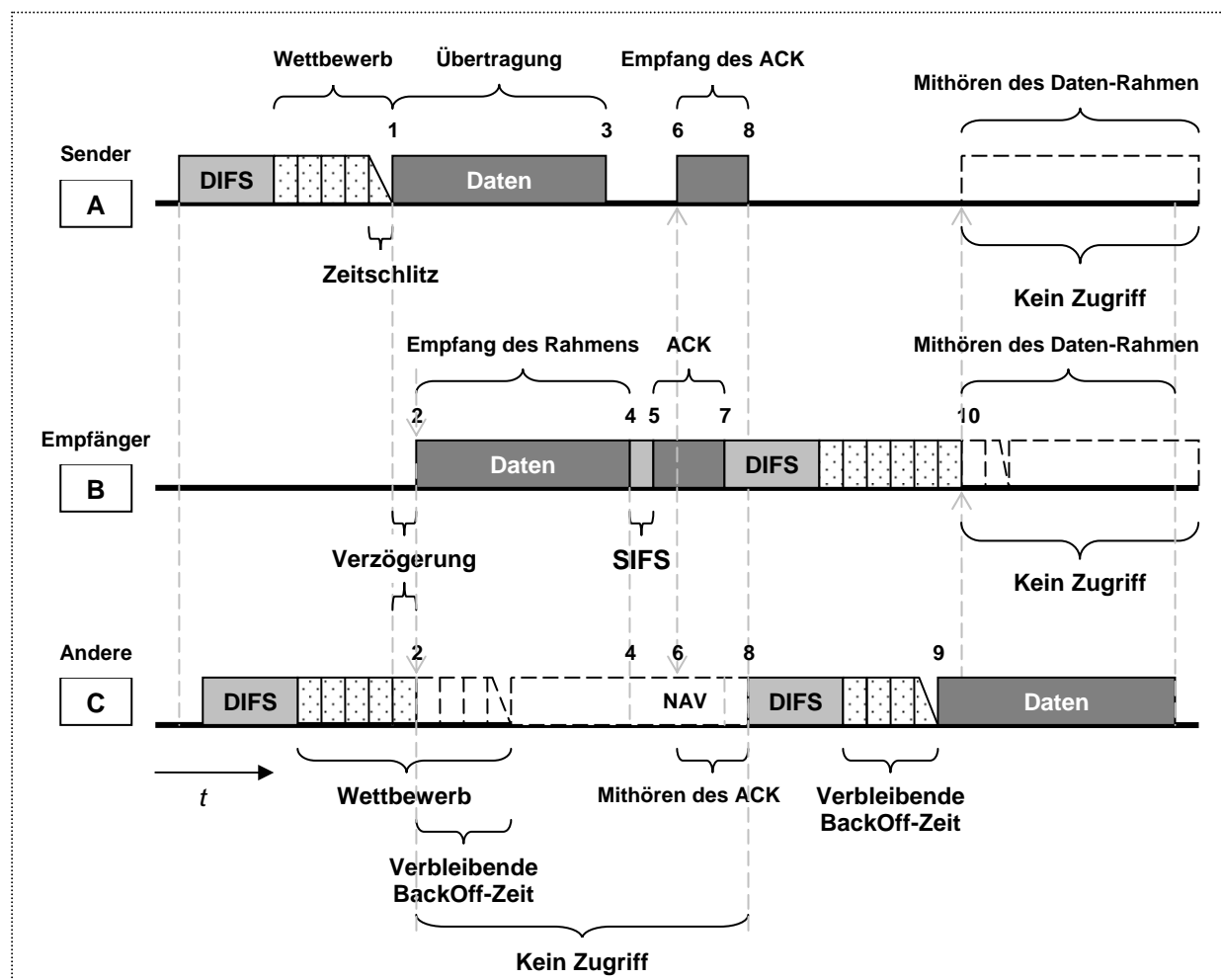


Abbildung 3-7 Zeitdiagramm von CSMA/CA mit ACK

1. A möchte einen Rahmen an Station B schicken. Nach einer DIFS- und einer BackOff-Zeit, hat sie das Zugriffsrecht bekommen und beginnt mit dem Senden.
2. Nach einer Verzögerung der Signalausbreitung beginnt Station B mit dem Empfang des Rahmens. Nun befindet sich die Station C gerade in Wettbewerb-Zustand, wird ihr BackOff-Timer angehalten und der Zugriff auf das Medium gesperrt.

3. A hat den Rahmen komplett gesendet und wartet nun auf die Quittung von Empfänger B.
4. B hat den von A gesendete Rahmen erfolgreich empfangen, wartet eine kurze SIFS-Zeit ab. Die Station C hat diesen Rahmen auch empfangen. Nach der Analyse der im Rahmenkopf enthaltenen Adressen, hat C entschieden, dass sie keinen Empfänger dieses Rahmens ist. Der NAV-Wert der Station C wird aktualisiert.
5. Nach einer SIFS-Zeit beginnt die Station B, die Empfangsbestätigung an A zurück zu senden.
6. Nach einer Verzögerung ist die Bestätigung an A angekommen. A fängt an, die Bestätigung von B zu empfangen.
7. Station B hat die Bestätigung an B abgeschickt. Ein erfolgreicher Empfang ist vorgesehen. B liefert diesen empfangenen Rahmen an die höhere Schicht. Nun möchte B noch ein Rahmen schicken: Sendprozess angefangen, also, Kanal zunächst für eine DIFS-Zeit abhören ...
8. Station A hat Bestätigung von B bekommen. Eine erfolgreiche Rahmensendung ist vorgesehen. Weil Station A keinen Rahmen zu senden hat, wechselt A ihren Zustand nach „Ruhe“. Das Medium aus der Sicht von C ist frei. Der verzögerte Sendprozess bei C ist zu diesem Zeitpunkt geweckt.
9. Nach Wettbewerb hat die Station C das Zugriffsrecht bekommen. Sie fängt an, ihre Nachricht zu senden.
10. Nach einer Verzögerung der Signalausbreitung detektiert die Station B ein Signal auf dem Kanal. Nun ist der Kanal als belegt markiert, der BackOff-Timer wird angehalten und C hat jetzt kein Zugriff auf den Kanal.

3.2.3 CSMA/CA – mit RTS/CTS

„Hidden Terminal“ Problem und „Exposed Terminal“ Problem

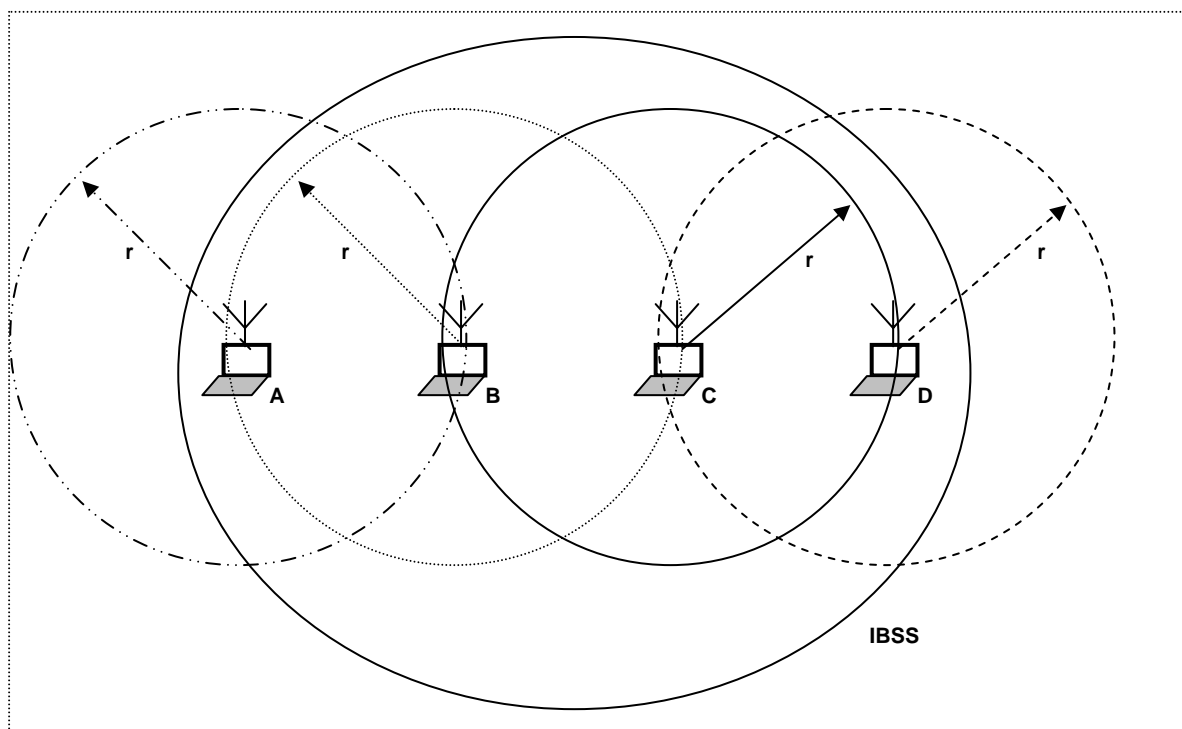


Abbildung 3-8 Standorte der Mobilstationen

Wir betrachten jetzt diese Situation. Die Positionen der Stationen A, B, C sehen wie Abbildung 3-8 aus:

1. A liegt innerhalb der Reichweite von B und kann Signale nur von B hören.
2. B liegt innerhalb der Reichweiten von A und C, kann nur die Signale von A und C empfangen.
3. C liegt innerhalb der Reichweiten von B und D, kann die Signale von B und D empfangen, von A aber nicht.
4. D liegt innerhalb der Reichweite von C und kann nur Signale von C hören

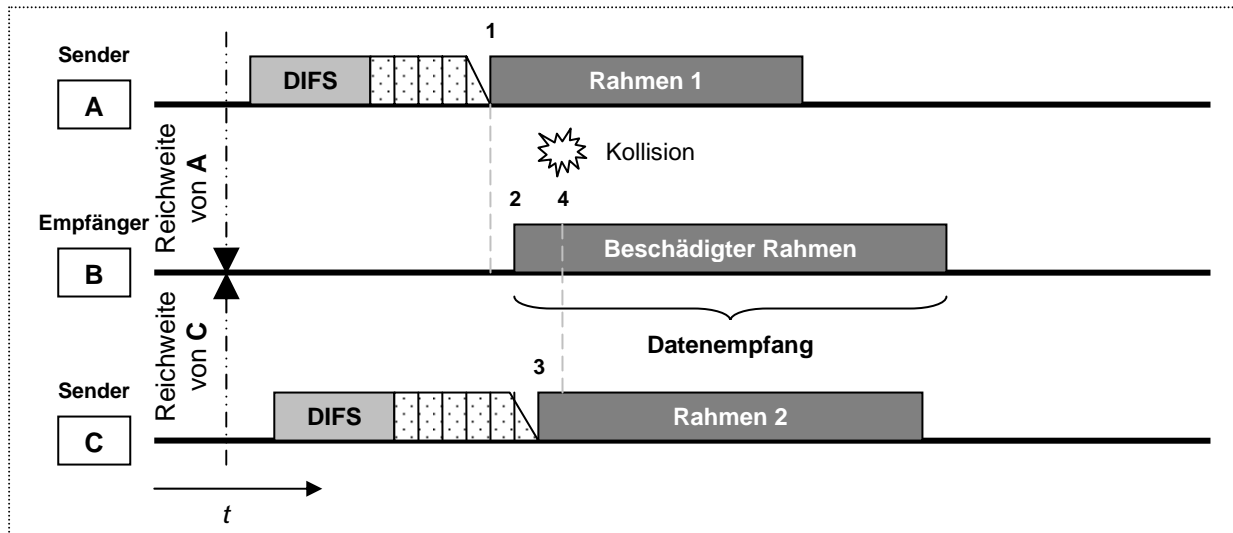


Abbildung 3-9 „Hidden Terminal“ Problem

Fall I: (siehe Abbildung 3-9)

1. A möchte ein Rahmen an B schicken. Nach dem Wettbewerb hat Station A das Zugriffsrecht auf Kanal genommen und beginnt mit der Sendung des Rahmen1.
2. Nach einer Verzögerung beginnt Station B den von A gesendete Rahmen zu empfangen.
3. Station C möchte auch einen Rahmen an B schicken. Aber C liegt außerhalb der Reichweite von A. Die Signale aus A kann C nicht hören. C hat nach dem Abhören des Kanals wahrgenommen, dass das Übertragungsmedium aus ihrer Sicht frei ist. Dann sendet C ihren Rahmen2 an B.
4. Nach einer Verzögerung ist der Rahmen2 an B angekommen. B empfängt nun Rahmen1 von A. Eine Kollision tritt offensichtlich auf.

Dies ist das bekannte „Hidden Terminal“ Problem (versteckte Station) in Wireless LAN. Nach dem CSMA/CA-Basismechanismus, können Kollisionen am besten vermieden werden, wenn die Reichweiten aller Stationen überlappend sind. Wenn eine Senderstation (z.B. die Station C) nicht in der Reichweite einer anderen Senderstation (z.B. A) liegt, kann sie den richtigen Kanalzustand bei Empfänger (z.B. B) nicht wahrnehmen und ihren Wettbewerb-Vorgang nicht anhalten.

In MAC-Protokoll des IEEE 802.11 ist eine Erweiterung des CSMA/CA-Basismechanismus beschrieben, nämlich CSMA/CA mit RTS/CTS.

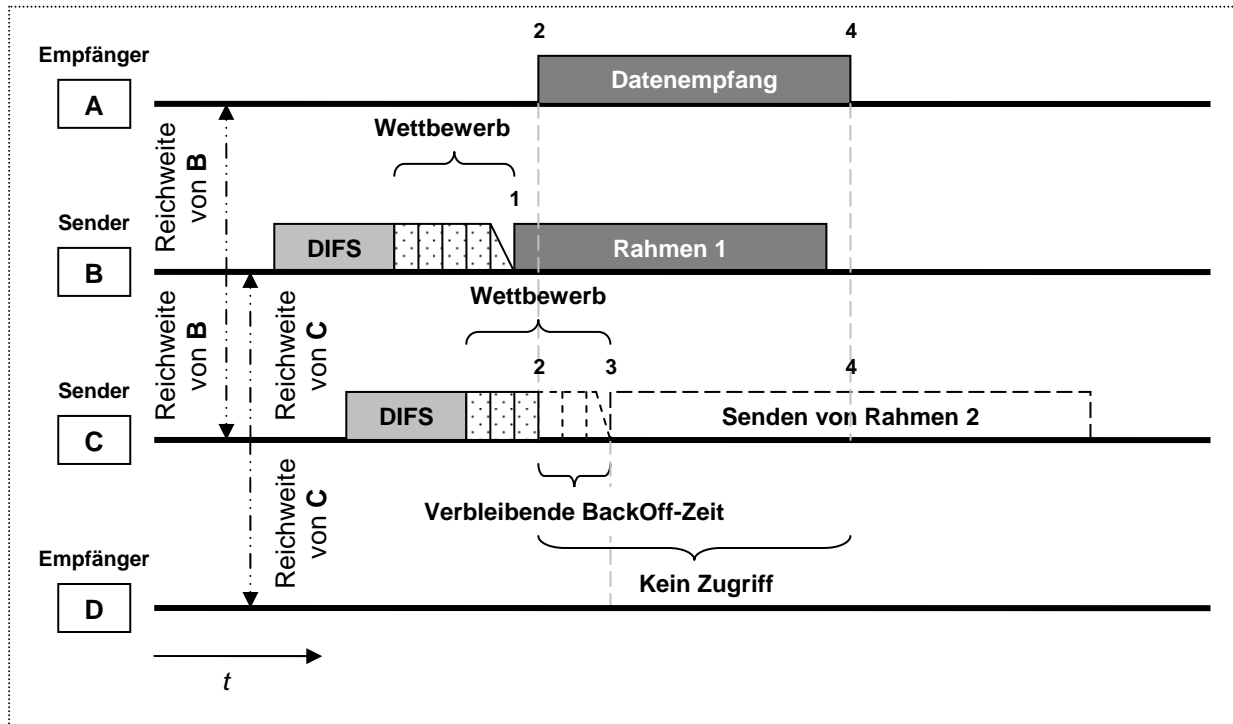


Abbildung 3-10 „Exposed Terminal“ Problem

Fall II: (siehe Abbildung 3-9)

1. Station B möchte einen Rahmen an A schicken. Nach dem Wettbewerb hat B das Zugriffsrecht auf Kanal genommen und beginnt, Rahmen1 zu senden.
2. Nach einer Verzögerung beginnt Station A, den von B gesendete Rahmen1 zu empfangen. Station C möchte eine Nachricht (Rahmen 2) an Station D schicken. Zu diesem Zeitpunkt detektiert die Station C, dass der Kanal nicht mehr frei ist, weil C innerhalb der Reichweite von B liegt. Den von B abgeschickte Rahmen hört C auch mit. Deswegen wird der Wettbewerb-Vorgang bei C angehalten. Allerdings liegt die Empfangesstation A des Rahmens1 außerhalb der Reichweite von C. Der von C abgeschickte Rahmen2 kann den Empfang von Rahmen1 bei A eigentlich nicht stören.

Dies ist das sogenannte „Exposed Terminal“ Problem in Wireless LAN. Das Senden von Rahmen2 soll zum Zeitpunkt 3 anfangen. Aber, wegen einer unnötigen Kanalsperrung wird es zum Zeitpunkt 4 verschoben. Es gibt bis jetzt leider noch keinen effizienten Mechanismus, um dieses Problem zu lösen.

RTS/CTS-Machanismus

Als eine Option der Methode der Medienzugriffssteuerung in IEEE 802.11 steht ein CSMA/CA mit RTS/CTS-Mechanismus (Request To Send / Clear To Send) zur Verfügung, um das „Hidden Terminal“ Problem zu lösen. Außerdem reduziert der Einsatz von RTC/CTS die Wartezeit, wenn eine Kollision auftritt.

Ein RTS ist eine Anfrage von einer Senderstation an eine Empfangsstation und fragt nach der Verfügbarkeit des Empfängers. In einem RTS Rahmen stehen die MAC-Adressen des Senders und Empfängers und eine voraussichtliche Übertragungsdauer. Nachdem der Empfänger das RTS erfolgreich empfangen hat, erzeugt der Empfänger ein entsprechendes CTS und sendet es zurück. Alle anderen Stationen, die innerhalb Reichweite der Senderstation liegen, hört das RTS auch mit und aktualisieren ihre NAV-Werte mit der in RTS enthaltenden Übertragungsdauer. Alle anderen

Stationen, die innerhalb der Reichweite des Empfängers liegen, hören das CTS mit und stellen ihre NAV mit der Übertragungsdauer in CTS ein, sodass sie während dieser Zeitdauer zur Datenübertragung gesperrt sind. Die Rahmenkommunikationen sind in Abbildung 3-9 dargestellt.

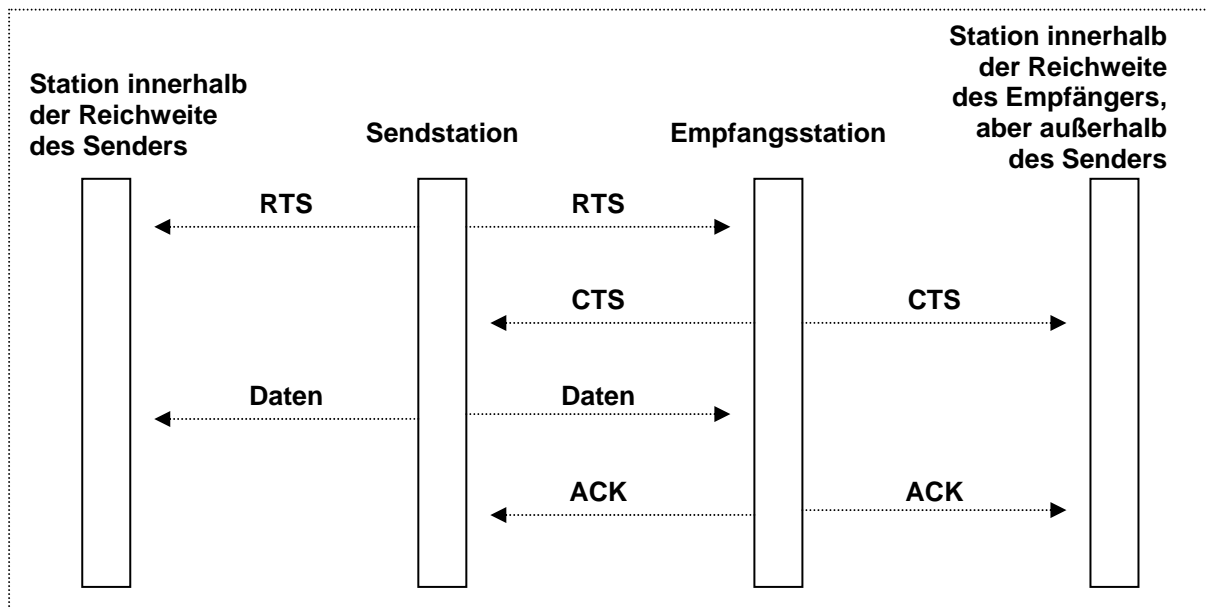


Abbildung 3-11 Rahmenkommunikationen bei CSMA/CA mit RTS/CTS

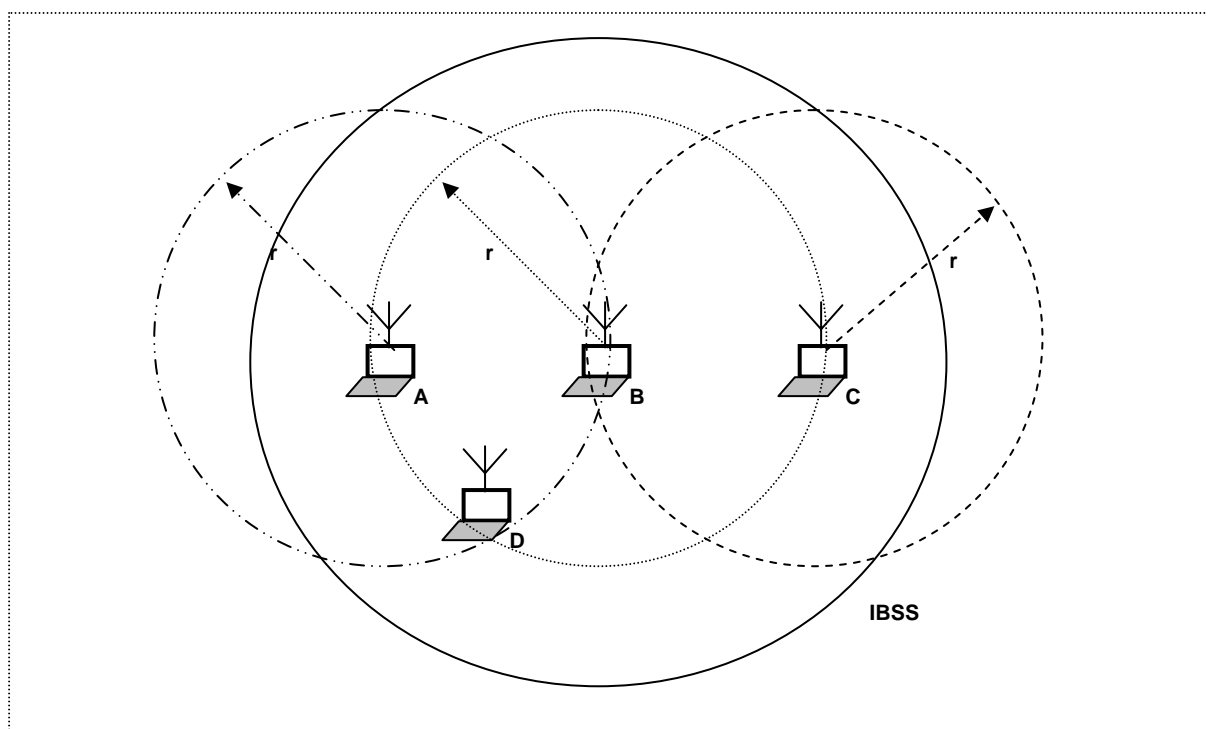


Abbildung 3-12 Standorte der Mobilstationen

Als Beispiel betrachten wir die Situation, wie Abbildung 3-9 zeigt:

1. A liegt innerhalb der Reichweiten von B und D.

2. B liegt innerhalb der Reichweiten von A, D und C.
3. C liegt innerhalb der Reichweite von B, kann die Signale nur von B empfangen.
4. D liegt innerhalb der Reichweite von A und B, kann Signale von A und B hören.

Ein Zeitdiagramm für CSMA/CA mit RTS/CTS ist in Abbildung 3-9 dargestellt.

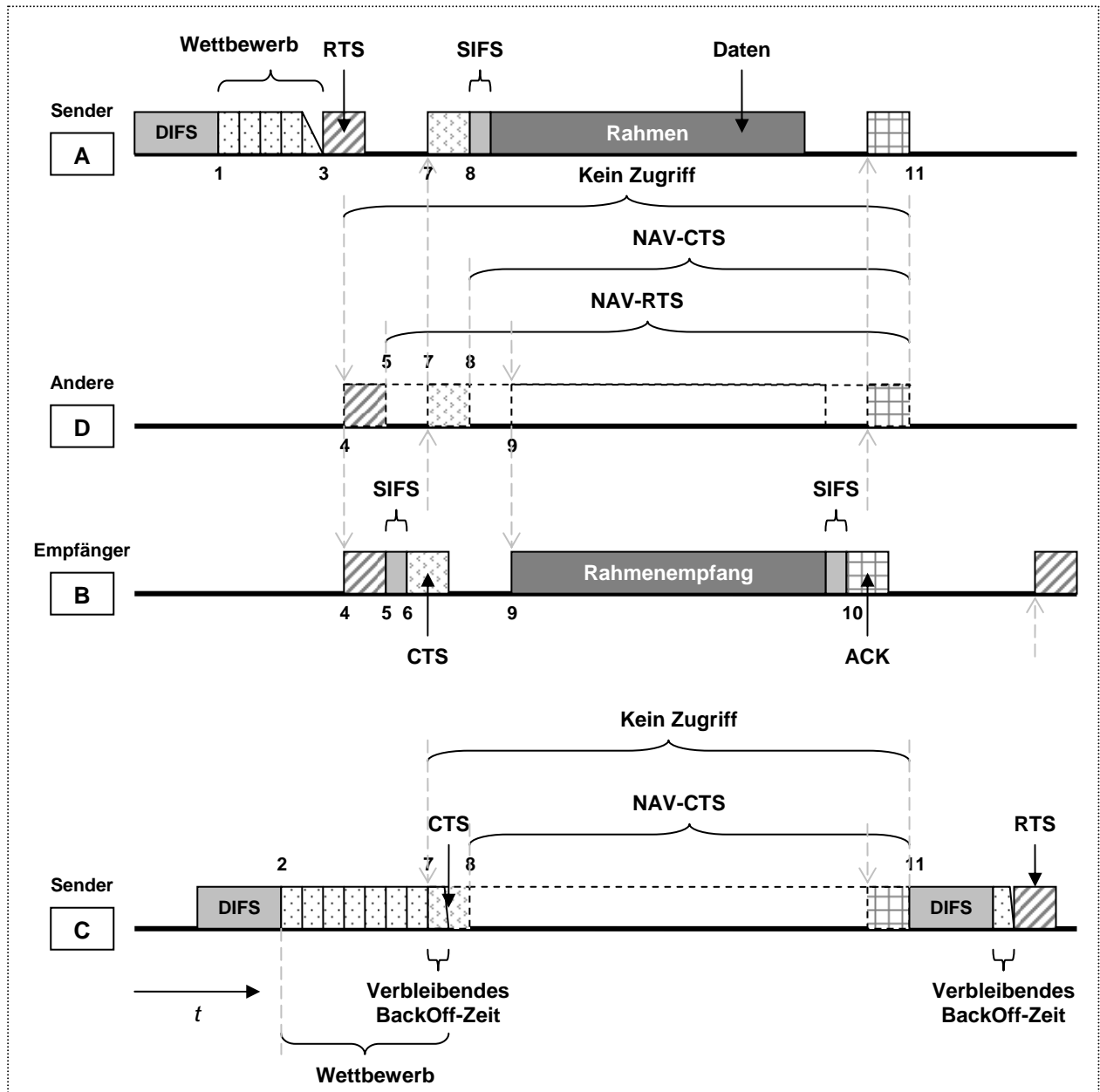


Abbildung 3-13 Zeitdiagramm des CSMA/CA mit RTS/CTS

1. Station A möchte einen Rahmen an die Station B senden. Nach dem Abwarten einer DIFS-Zeit ruft A die BackOff-Funktion auf. Mit der neu generierten Wartezeit geht sie in die Wettbewerbs-Phase.

2. C möchte einen Rahmen auch an die Station B schicken und wie Station A, fängt C mit einem Wettbewerb-Prozess an.
3. Der Wettbewerb-Prozess bei B ist abgeschlossen. A sendet einen RTS sofort aus.
4. Nach einer Verzögerung der Signalausbreitung starten die Stationen B und D, die innerhalb der Reichweite von A liegen, den RTS zu empfangen.
5. Nach dem fehlerfreien Empfang von RTS, wartet die Station B eine SIFS-Zeit ab. Station D hat den RTS auch erfolgreich empfangen und festgestellt, dass der RTS nicht für sie ist. Der NAV-Wert ist nun aktualisiert mit der Zeitdauer aus dem RTS-Rahmen.
6. Die SIFS-Zeit bei B ist abgelaufen. B sendet einen entsprechenden CTS an den Kommunikationspartner A zurück. Wie der RTS, beinhaltet der CTS eine neu berechnete Übertragungsdauer.
7. Die Stationen A fängt an, den von B abgesendeten CTS zu empfangen. Weil die Station C innerhalb der Reichweite von dem Sender des CTSs B liegt, hat C detektiert, dass das Medium nicht mehr frei ist. Ihr Wettbewerb-Prozess ist gestoppt und der entsprechende BackOff-Timer wurde angehalten.
8. Nach dem Empfang des CTS von B, weiß Station A nun, dass der Kanal für die weitere Übertragung von Daten-Rahmen erfolgreich reserviert wurde. Anschließend wartet A eine SIFS-Zeit ab. Zu diesem Zeitpunkt wird der NAV-Wert bei D mit der neuen Übertragungsdauer aktualisiert. Bei der Station C, wurde ihr NAV-Wert auch eingestellt.
9. B hat angefangen, den von A gesendete Daten-Rahmen zu empfangen.
10. Nach dem erfolgreichen Empfang des Daten-Rahmens, hat der Empfangsstation B erkannt, dass das Medium für eine SIFS-Zeit frei ist. B sendet dann die Bestätigung an A zurück.
11. Nach dem Erhalt des ACK-Rahmens ist das Senden bei A erfolgreich abgeschlossen. Der NAV-Timer bei C erreicht Null, darf ihr wegen einer Kanalbelegung verzögerte Sendevorgang erneut gestartet werden, und zwar mit der verbleibenden BackOff-Zeit von dem letzten Sendversuch.

Mit dem Einsatz von RTS/CTS-Mechanismus ist das „Hidden Terminal“ Problem gelöst. Obwohl die Wahrscheinlichkeit der Kollision nebenbei weiter reduziert wurde, besteht immer noch Kollisionsgefahr, da die CTS und RTS kollidieren könnten. Im Gegensatz zum CSMA/CA-Basismechanismus ist die Wartezeit zur Datenübertragung relativ länger. Wenn die Reichweiten von den Stationen in einem IBSS miteinander überlappend sind und voraussichtlich kürzere Rahmen am meisten gesendet werden, lohnt sich der Einsatz von RTS/CTS-Mechanismus nicht mehr. Der RTS/CTS-Mechanismus ist für Broadcasting und Multicasting nicht einsetzbar, weil die RTS- und CTS-Rahmen nur für einen einzelnen Empfänger sind.

3.3 PCF

Als Option ist PCF (Point Coordination Function) in IEEE 802.11 spezifiziert. PCF baut auf der Basis von DCF auf und verwendet die grundlegenden Regeln von DCF, z.B. den Mechanismus zur Ermittlung der Kanalbelegungsinformationen. Im Gegensatz zur DCF wird ein PC (Point Coordinator) in PCF eingesetzt, der zur Regelung der gesamten Kommunikationen dient. Wobei der PC entscheidet, wann und welche Mobilstation Daten senden darf. Da ein zentraler Verwalter benötigt wird, ist PCF nur in Infrastruktur-Modus einsetzbar. Der AP (Access Point, Zugangspunkt) in Infrastruktur-Netzwerk wird normalerweise als PC benutzt.

Der PC steuert die Übertragung während einer CFP-Zeitdauer (Contention free Period). Ein CFP-Intervall besteht aus einer CFP-Dauer und einer CP-Dauer (Contention Period). Die CP folgt der CFP, wie die Abbildung 3-14 zeigt. Wie es im Abschnitt 2.4.1.1 vorgestellt, verwaltet der PC eine Polling-Liste. In der CFP-Zeitdauer, darf nur die vom PC aufgerufene Station ihre Nachricht senden. In der CP funktioniert die DCF und dürfen alle Stationen um das Zugriffsrecht auf Kanal konkurrieren. Wobei ist eine PIFS (PCF Inter Frame Space) eingesetzt, die kürzer als DIFS und länger als SIFS ist. Wie bei

DCF, wartet eine Station eine DIFS-Zeit, bevor sie in der CP den Kanal zugreifen will. Statt DIFS-Zeit, wartet der PC eine PIFS-Zeit ab, damit der PC höhere Priorität als andere Stationen hat. Die Abbildung 3-14 zeigt, wie die PCF grundsätzlich funktioniert. Dabei sind die Nachrichten über PC vermittelt (Die Vermittlung über PC ist nicht erforderlich in Wireless LAN. Die aufgerufene Station darf ihre Nachricht direkt ans Ziel senden.).

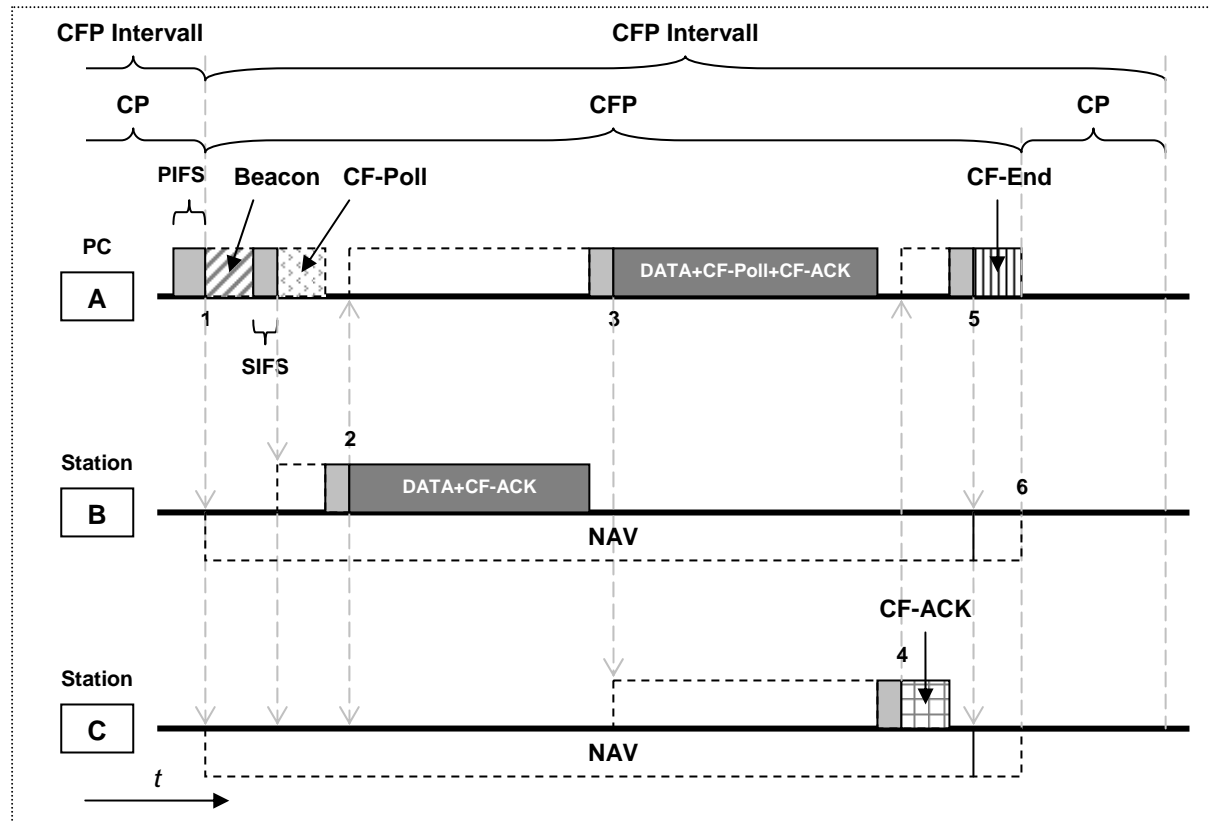


Abbildung 3-14 Zeitdiagramm von PCF

1. Der PC hat Zugriffsrecht bekommen und sendet einen Beacon ab. Der Beacon enthält die Zeildauer einer CFP. Nach dem Empfang von Beacon setzen alle anderen Stationen ihre NAV-Werte.
2. In der Polling-Liste ist die Station B die nächste Station. Der PC sendet einen CF-Poll-Rahmen an Station B. Nach dem Aufruf von PC sendet B ihre Nachricht (DATA+CF-ACK).
3. Der PC hat den von B gesendeten Rahmen „DATA+CF-ACK“ empfangen und sendet den „DATA+CF-Poll+CF-ACK“ ab. Dabei ist DATA die von B geschickte Daten, entspricht der CF-Poll dem Aufruf an Station C und ist CF-ACK die Quittung an Station B für die vorherige Sendung.
4. Station C hat den Rahmen erhalten und gerade keine Nachricht zu senden. So antwortet sie den PC mit einem CF-ACK, um dem PC anzugeben, dass sie den Datenrahmen erfolgreich empfangen wurde.
5. Nach dem Erhalt des CF-ACK von der Station C hat der PC festgestellt, dass das letzte Senden erfolgreich abgeschlossen ist und die C keine Daten senden möchte. Der PC sendet dann einen CF-End ab, der das Ende der CFP und den Anfang der CP angibt.
6. Die Stationen haben den CF-End empfangen, und sie wiesen damit, dass sie das Zugriffsrecht auf Medium konkurrieren dürfen.

3.4 Fragmentierung

Wie im Abschnitt 3.2 erläutert, kann die Sendstation die Kollisionen während des Sendens nicht erkennen. Obwohl Kollisionen in Wireless LAN mit dem Einsatz des CSMA/CA-Standardalgorithmus weitgehend vermieden werden können, besteht immer noch Kollisionsgefahr. Ein durch Kollision zerstörter Daten-Rahmen verursacht eine schlechte Kanalausnutzung. Als Option bietet CSMA/CA die Fragmentierung für Nachrichten.

Eine lange Nachricht kann auf mehrere kürzere Fragmente aufgeteilt gesendet werden. Auf der anderen Seite, setzt der Empfänger alle Fragmente wieder zu einer Nachricht zusammen (Defragmentierung) und liefert diese anschließend an die höhere Kommunikationsschicht. Allerdings müssen alle Fragmente einer Nachricht einzeln bestätigt werden. Wenn Kollisionen auftreten würden, kann der Sender keine Bestätigung für das kollidierte Fragment bekommen. Statt eine ganze Nachricht neu zu übertragen, wird nur das kürzere Fragment, das von dem Empfänger nicht bestätigt ist, erneut gesendet. Die unnötige Mediumsbelegung wird auch reduziert, da das Fragment kürzer ist.

Weil die Fragmente bestätigt werden müssen, ist die Fragmentierung von Nachrichten für die Rahmen mit Broadcast- oder Multicast-Adressen nicht geeignet. Die Fragmente, die zu einer Nachricht gehören, haben die gleiche Sequenznummer. Diese Sequenznummer ist im Kontrollfeld des Rahmenkopfes eingetragen (siehe Abschnitt 4.3.2 und Abbildung 4-3). Jedes Fragment besitzt eine Fragmentnummer, die von null an beginnt und um eins erhöht wird. Außerdem wird das Mehr-Frag-Bit im Kontrollfeld jedes Fragments gesetzt, bis auf das letzte Fragment. Dadurch kann der Empfänger erkennen, dass es sich um eine fragmentierte Nachricht handelt. Nachdem der Empfänger das letzte Fragment empfangen hat (Mehr-Frag-Bit im letzten Fragment ist nicht gesetzt), weiß er dann, dass die ganze Nachricht vollständig übertragen worden ist. Nach den Fragment-Nummern packt er die Nutzdaten in den Fragmenten in eine Nachricht ein und liefert sie aus.

Als Beispiel zeigt Abbildung 3-9, wie die Nachrichtübertragung mit Fragmentierung funktioniert. Hier ist die Signalausbreitungszeit zur Vereinfachung ignoriert und CSMA/CA-Mechanismus mit RTS/CTS verwendet. Die Nachricht ist auf drei Fragmente verteilt. Die Standorte der Stationen A, B und D sind wie in Abbildung 3-9 dargestellt.

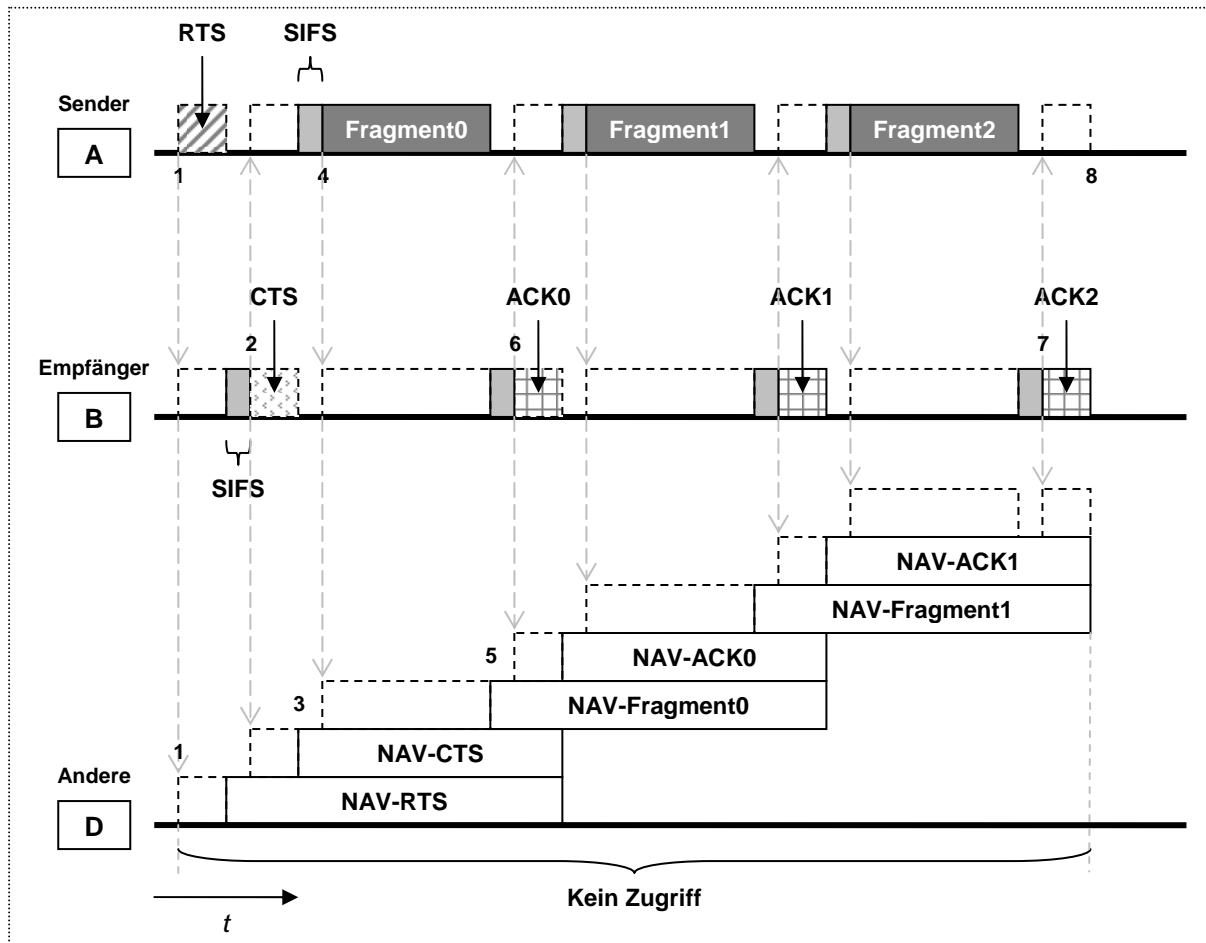


Abbildung 3-15 Nachrichtenübertragung mit Fragmentierung

1. Station A sendet einen RTS ab. Station D hört den RTS auch mit. Ab jetzt hat D kein Zugriff auf das Medium. Nach dem erfolgreichen Empfang des RTS, setzt D ihren NAV-Wert mit der im RTS enthaltene Übertragungsdauer.
2. Nach dem Empfang von RTS und Abwarten einer SIFS-Zeit sendet Station B das CTS zurück.
3. Station D aktualisiert ihren NAV-Wert mit der Dauer in CTS.
4. Die Sendestation A hat das CTS empfangen und eine SIFS-Zeit gewartet, sendet A nun das erste Fragment (Fragment0) ab.
5. D aktualisiert ihren NAV-Wert mit der Dauer in Fragment0, also mit der Zeitdauer für die nachfolgende Übertragung von Fragment1.
6. Nach dem erfolgreichen Empfang des Fragment0 weiß B nun, dass es ein Fragment einer Nachricht ist und noch weitere Rahmen gibt (More-Frag-Bit). Jetzt sendet B die entsprechende Bestätigung zurück (ACK0).
7. B hat das Fragment2 empfangen, weiß B, dass es das letzte Fragment ist (More-Frag-Bit = 0). B sendet ACK ab und setzt die empfangenen Fragmente zu einer Nachricht zusammen anschließend. Ein erfolgreicher Empfang ist abgeschlossen.
8. Nach dem Empfang von ACK2 bei Station A liegt ein erfolgreicher Sendevorgang vor.

Ab welcher Länge eine Nachricht fragmentiert gesendet werden soll, ist ein einstellbarer Parameter. Durch Fragmentierung wird die unnötige Kanalbelegung reduziert. Allerdings führen die zusätzliche Bestätigungen und mehrfache SIFS-Wartzeit zu einer größeren Verzögerung.

Kapitel 4 Konzeption und Entwurf

4.1 Architektur

Für die Emulation von drahtlosen Netzwerken gibt es grundsätzlich zwei Verfahren: eine zentralisierte und eine verteilte Methode. Bei der zentralisierten Emulationsmethode verwaltet ein Kommunikations- und Emulationskoordinator die Verzögerung, Kollision und die Zugriffe aufs Medium. Wenn der Koordinator ausfällt, kann die gesamte Emulation nicht mehr durchführen (Single Point of Failure). Da der Koordinator den gesamten Datenverkehr verwaltet, ist die Skalierbarkeit bei der zentralisierten Emulationsmethode schlecht. Die oben genannten Nachteile bestehen in der verteilten Emulationsmethode nicht.

Zur Emulation für Netzwerke mit gemeinsamem Funkmedium wird die verteilte Methode ausgewählt. Abgekürzt von WireLess EMULATION ist „WILEMU“ die entwickelte Emulationssoftware. Durch die VLAN-Konfiguration können mehrere BSS emuliert werden. Ein VLAN entspricht einem BSS. In einem BSS gibt es mehrere Mobilstationen, die durch Instanzen von WILEMU repräsentiert werden. Die Instanzen unterscheiden sich durch verschiedene MAC-Adressen. Zur Vereinfachung wird eine Instanz von WILEMU in diesem Kapitel auch WILEMU genannt. Abbildung 2-20 zeigt, wie WILEMU im Kommunikationsstapel eingreift. Nach der Bearbeitung durch WILEMU, werden die Nachrichten an höhere Schicht geliefert bzw. mittels NET-Hardware über das kabelgebundenes Emulationsnetzwerk verschickt.

4.2 Emulation des Funkkanals

4.2.1 Tunnelung

In NET-Emulationsumgebung wird ein Giga-Bit-Ethernet-Switch verwendet. Der Datenverkehr für Emulation geht über diesen Switch. Der Switch realisiert eine Stern-Netzwerktopologie und verwaltet eine Menge von Punkt-zu-Punkt Verbindungen. Jeder übertragene Rahmen wird durch den Switch analysiert und direkt ans Ziel geschickt. Da die Kollisionen durch Switch-Technologie verhindert werden und die Rahmen eventuell auch dazwischen gepuffert werden, ist diese Switch-Eigenschaft für die Emulation für Netze mit gemeinsamem Funkmedium unerwünscht. Um die Effekte dieser Eigenschaft zu beseitigen, muss jeder Rahmen, der über WILEMU übertragen wird, in einem Broadcasting-Ethernet-Rahmen getunnelt gesendet werden, wie die Abbildung 4-1 zeigt. Dabei wird der gesamte WILEMU-Rahmen als Nutzdaten eines Ethernet-Rahmens eingefügt.

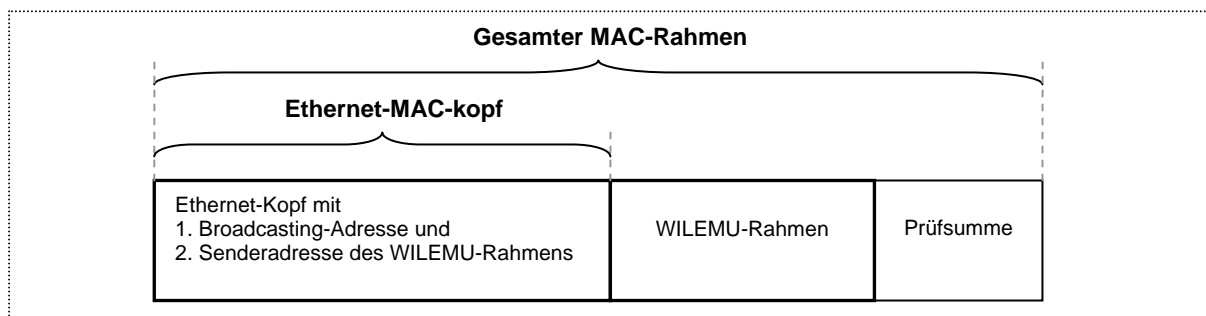


Abbildung 4-1 WILEMU-Rahmen muss getunnelt gesendet werden

Dadurch können alle Netzwerkteilnehmer in einem VLAN (ein VLAN entspricht hier einem IBSS in Wireless LAN) den Rahmen empfangen. So ist die Eigenschaft des gemeinsamen Mediums wiederhergestellt.

4.2.2 Reichweite

Im Gegensatz zur idealen Umgebung, die in Abschnitt 3.2.1 definiert ist, können die Reichweiten der Mobilstationen in der Realität verschiedene Größen haben, da die Signalausbreitung durch viele Faktoren beeinflusst werden kann (vgl. Kapitel 2) und Sendestärke der Mobilstationen unterschiedlich sein können.

Die Reichweite der Mobilstation wird in WILEMU durch die Fehlerrate realisiert. Nach der Konfiguration des Emulationsszenarios werden die Fehlerraten mittels des Steuerungsnetzwerks an den Mobilstationen übermittelt. Zu der ersten Version des WILEMUs besitzt die Fehlerrate nur zwei Extremwerte, 0 und 1 (100%).

Die Fehlerrate ist eigentlich für die Verbindungsqualität gedacht. Jede Fehlerrate ist immer mit zwei Stationen gebunden. Als Beispiel betrachten wir diese Konfiguration, die in Abbildung 4-2 dargestellt ist. Die entsprechenden Fehlerraten sind in Tabelle 4-1 aufgelistet.

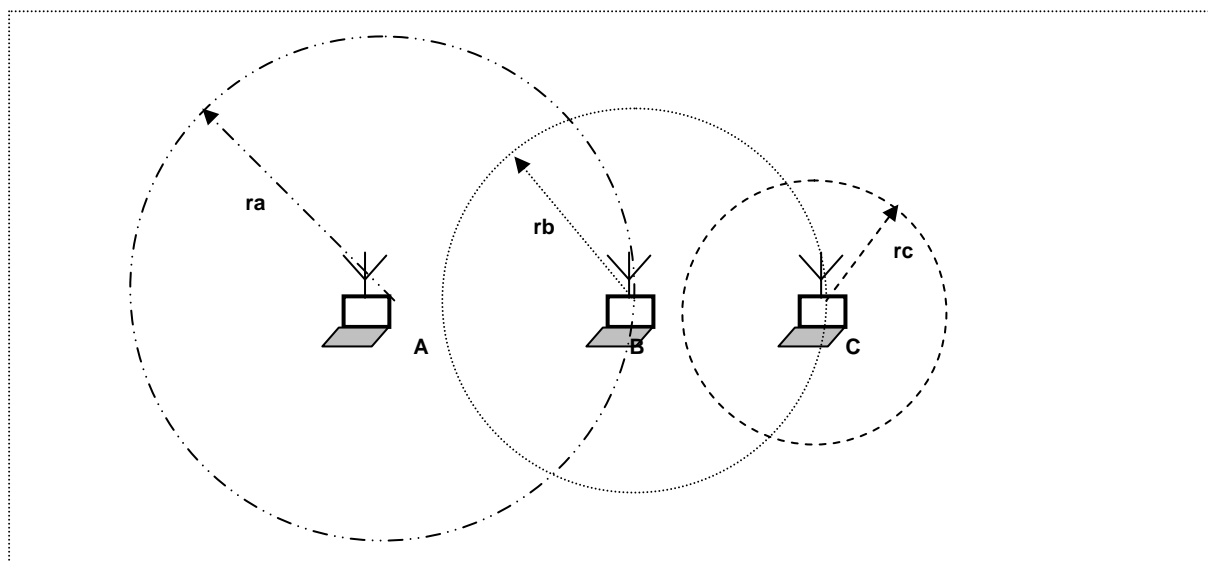


Abbildung 4-2 Mobilstationen haben verschiedene Reichweiten

| | | | | | | |
|-------------------|---|---|---|---|---|---|
| Sender | A | A | B | B | C | C |
| Empfänger | B | C | A | C | A | B |
| Fehlerrate | 0 | 1 | 1 | 0 | 1 | 1 |

Tabelle 4-1 Verbindungsfehlerrate

Die Signale aus A kann die Station B hören. Die Fehlerrate von A nach B beträgt 0, während die Fehlerrate von B nach A den Wert 1 hat. Da die Sendestation B eine kleine Reichweite besitzt, liegt die Empfangsstation A nicht innerhalb der Reichweite von B.

In WILEMU verwaltet jede Mobilstation eine Fehlerrateliste. Die Fehlerrateliste beschreibt die Fehlerraten nur in Empfangsrichtung. In diesem Sinn sind die in der Liste eingetragenen Stationen die Sender dem Besitzer der Fehlerrateliste. Die Tabelle 4-2 zeigt beispielsweise die Fehlerrateliste von Mobilstation B. Z.B. in der zweiten Zeile der Liste, haben die Signale aus C die Fehlerrate 1, d.h.

Station B kann keine Signale von Station C empfangen, da B außerhalb der Reichweite von C liegt und die Signale aus C verloren gehen werden.

| Sendstation | Fehlerrate |
|-------------|------------|
| A | 0 |
| C | 1 |
| | |

Tabelle 4-2 Fehlerrateliste bei Station B

Die Fehlerrateliste realisiert die Beschreibung der Reichweiten von den Sendstationen. Nach den Regeln des CSMA/CA sendet eine Station ihre Nachricht, ohne die Fehlerrateliste zu berücksichtigen. Wenn eine Station einen Rahmen empfangen hat, muss sie zunächst überprüfen, ob ein Eintrag des Senders in der Fehlerrateliste des Empfängers steht. Falls nicht, oder der gefundene Eintrag Fehlerrate 1 besitzt, muss der Rahmen weggeworfen werden, weil der Empfänger außerhalb der Reichweite des Senders liegt.

4.2.3 Signalausbreitungsverzögerung

Die Signalausbreitungsgeschwindigkeit in Wireless LAN ist die Lichtgeschwindigkeit. Nach der Spezifikation in IEEE 802.11 ist die Signalausbreitungsverzögerung sehr klein, siehe Tabelle 4-3.

| Übertragungsrate | Maximale Entfernung | Ausbreitungsverzögerung |
|------------------|---------------------|-------------------------|
| 11 MBit/s | 30 m | 0.1 μ s |
| 1 MBit/s | 100 m | 0.33 μ s |

Tabelle 4-3 Signalausbreitungsverzögerung in IEEE 802.11

In der Emulationsumgebung ist die Signalausbreitungsverzögerung länger als in realer Wireless LAN, da die Signalausbreitungsgeschwindigkeit in Ethernet relativ langsamer ist (etwa $\frac{2}{3}$ von Lichtgeschwindigkeit) und extra eine interne Bearbeitungszeit benötigt wird. Wegen der kürzeren Entfernung zwischen Stationen in Emulationsnetzwerk in NET-Umgebung und der Granularität des Timers (10 μ s, vgl. Abschnitt 4.3.4 BackOff) nehmen wir an, dass die Signalausbreitungsverzögerungen in Emulationsumgebung und realer Umgebung gleich sind.

Annahme1:

Die Signalausbreitungsverzögerungen in Emulationsumgebung und realer Umgebung sind gleich.

4.3 Emulation der MAC-Schicht

4.3.1 Netzwerktopologie, Medienzugriffsverfahren und optionale Funktionen in MAC

| Zugriffsmethode | Topologie | |
|-----------------|---------------|--------|
| | Infrastruktur | Ad-hoc |
| CSMA/CA (DCF) | X | X |
| PCF | X | --- |

Tabelle 4-4 Netzwerktopologie und Medienzugriffsverfahren

Ein Wireless LAN kann in Infrastruktur-Modus oder in Ad-hoc-Modus funktionieren. Die Grundlegende Medienzugriffsmethode in Wireless LAN ist die auf DCF basierende CSMA/CA. Die optionale Medienzugriffsmethode PCF benutzt auch die Basisregeln von DCF, wie es in Kapitel 3 erläutert. Wie die Tabelle 4-4 zeigt, kann CSMA/CA sowohl in Infrastruktur-Modus als auch in Ad-hoc-Modus eingesetzt werden, während die PCF nur in Infrastruktur-Modus funktionieren kann.

Da die Abteilung Verteilte Systeme der Universität Stuttgart höheren Bedarf an die Emulation für Ad-hoc-Netzwerke, muss die WILEMU Wireless LAN in Ad-hoc-Modus unterstützen. Für die Emulation werden CSMA/CA als Zugriffsmethode und Ad-hoc-Modus als Netzwerktopologie ausgewählt. Damit können die wichtigste Verhalten der MAC-Schicht emuliert, die Komplexität der Implementierung wegen der Zeitbeschränkung reduziert und die Anforderung der Abteilung erfüllt werden.

Da das „Hidden Terminal“-Problem in Ad-hoc-Netzwerk vorkommen kann, muss neben dem Basismechanismus des CSMA/CA auch noch das RTS/CTS-Mechanismus eingesetzt werden. WILEMU muss vor dem Senden eines Rahmens entscheiden, ob der Rahmen die Broadcast-Adresse besitzt. Wenn ja, soll der CSMA- Basismechanismus verwendet werden.

Aus den gleichen Gründen werden die optionalen Funktionen wie WEP (Abschnitt 2.5.5), Fragmentierung (Abschnitt 3.4) und Stromsparen (siehe [IEEE811]) in der ersten Version von WILEMU nicht unterstützt. Diese Diplomarbeit ist konzentriert auf der Medienzugriffsmethode in MAC-Schicht von IEEE 802.11.

4.3.2 Rahmentypen und Rahmenformat

Wie die Abbildung 4-3 zeigt, ist die MAC-Schicht nach ISO/OSI-Model eine Teilschicht der Sicherungsschicht. Zum Datenabsenden nimmt die IEEE 802.11 MAC-Schicht die Daten aus 802.11 LLC als MAC-Nutzdaten über und fügt einen MAC-Kopf (siehe Abbildung 2-3) ein. Anschließend gibt sie die Daten an die unterliegende Bitübertragungsschicht. Die Dateneinheit der MAC-Schicht nennt sich Rahmen (Frame).

In IEEE 802.11 werden die MAC-Rahmen in drei Kategorien definiert, nämlich Daten-Rahmen, Steuerungs-Rahmen und Verwaltungs-Rahmen. Nach ihren Anwendungen unterscheiden sich die Rahmen beim Haupttyp weiter mit der Subtyp-Kennung (siehe Tabelle 4-5). Die genauen Bedeutungen des Haupttyps bzw. der Subtypen und deren Werte sind in [IEEE811] beschrieben.

| Typ | Wert in MAC-Kopf | Subtyp | Wert in MAC-Kopf |
|---------------------------|------------------|----------------------|------------------|
| Daten-Rahmen | 10 | DATA | 0000 |
| | | CF-Poll | 0110 |
| | | | |
| Steuerungs-Rahmen | 01 | RTS | 1011 |
| | | CTS | 1100 |
| | | ACK | 1101 |
| | | | |
| Verwaltungs-Rahmen | 00 | Beacon | 1000 |
| | | Probe-Anfrage | 0100 |
| | | Assoziation -Anfrage | 0000 |
| | | | |

Tabelle 4-5 Wichtige Datentypen und Subtypen in IEEE 802.11 [IEEE811]

Durch die Analyse von Typ und Subtyp kann die Mobilstation erkennen, was für einen Rahmen sie empfangen hat.

Nach dem Auswahl von Netzwerktopologie und Medienzugriffsmethode werden in dem Emulationswerkzeug WILEMU nur zwei Typen unterstützt: Daten-Rahmen und Steuerungsrahmen. Da der Verwaltungsrahmen eher für die Stationsanmeldung und -Abmeldung, Authentifizierung und die Erkennung von Zelle verwenden, wird er in der Emulation weggelassen. Weil wir für Emulation kein PCF einsetzen wollen, werden die Subtypen, die in Daten-Rahmen und Steuerungsrahmen definiert und in PCF benutzt sind, auch weggelassen.

| Typ | Wert |
|------|--------------|
| DATA | Protokolltyp |
| ACK | 0x8881 |
| RTS | 0x8882 |
| CTS | 0x8883 |

Tabelle 4-6 Rahmenstyp in der Emulation

Da in WILEMU nur Daten-, ACK-, RTS- und CTS-Rahmen verwendet werden, werden die Rahmenstypen und Subtypen in der WILEMU nicht mehr kategorisiert. Sie sind einheitlich als Rahmenstypen definiert, wie die Tabelle 4-6 Rahmenstyp in der Emulation aufzeigt.

Abbildung 4-3 zeigt das allgemeine Rahmenformat in IEEE 802.11 MAC.

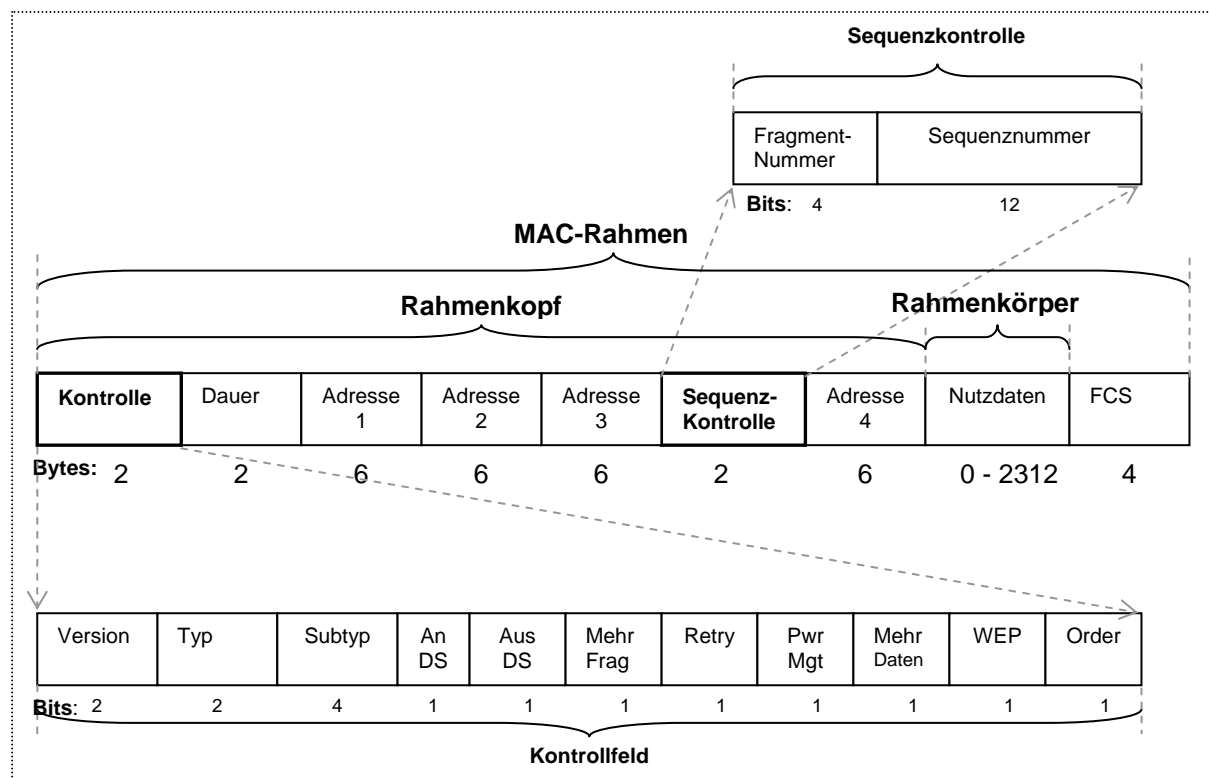


Abbildung 4-3 Allgemeines Rahmenformat in IEEE 802.11 MAC [IEEE811],[nemg01]

Ein MAC-Rahmen besteht aus drei Teilen: Rahmenkopf, Rahmenkörper und FCS (Frame Check Sequence). Das FCS beinhaltet ein 32-Bits-CRC (Cyclic Redundancy Code, Prüfsumme). Wie im übrigen Ethernet, wird die Prüfsumme zur Erkennung von Übertragungsfehler benutzt. Der Rahmenkörper entspricht den Nutzdaten, die von LLC-Schicht übernommen werden. Die maximale

Länge beträgt 2312 Bytes. Das Format des Rahmenkopfs sieht ganz anders aus als in Ethernet, da das IEEE 802.11 MAC verschiedene Medienzugriffsmethoden unterstützt. Am Anfang des Rahmenkopfs beträgt die Länge des Kontrollfelds 16 Bits. Die Bedeutungen des jeweiligen Feldes finden Sie in [IEEE811]. Die Informationen über Rahmenstyp sowie Subtyp werden in das Kontrollfeld eingetragen.

Die genaue Beschreibung der einzelnen Felder in Kontrollfeld finden Sie in [IEEE811].

Die Bedeutungen der vier Adressen-Felder in IEEE 802.11 sind in Tabelle 4-7 aufgelistet.

| | | | | |
|------------------|--|---|--|--|
| An DS | 0 | 0 | 1 | 1 |
| Aus DS | 0 | 1 | 0 | 1 |
| Adresse1 | Empfänger | Empfänger | Zelle des Senders | Zelle des Empfängers |
| Adresse2 | Sender | Zelle des Empfängers | Sender | Zelle des Senders |
| Adresse3 | Zelle | Sender | Empfänger | Empfänger |
| Adresse4 | - | - | - | Sender |
| Bedeutung | Der Rahmen wird nur innerhalb einer Zelle versendet. | Der Rahmen wird vom AP aus dem DS an eine Station in der Zelle geschickt. | Der Rahmen wird von einer Station aus einer Zelle an den AP zur Weiterleitung in DS versendet. | Der Rahmen wird von einer Station in einer Zelle über AP an eine Station in einer anderen Zelle versendet. |

Tabelle 4-7 Adressenfelder in MAC-Kopf nach IEEE 802.11 [siko01]

In der Emulation betrachten wir nur ein IBSS, da sich auf der physischen Ebene mehrere IBSS miteinander durch Raummultiplex oder Frequenzmultiplex nicht stören. In Tabelle 4-7 wird nur die Kombination in der ersten Spalte benutzt, da die Anderen nur in PCF vorkommen. Wenn mehrere IBSS emuliert werden sollten, kann die Erkennung von Zelle durch die entsprechende VLAN-Konfiguration realisiert werden. Deswegen wird die Zelladresse in Rahmenkopf nicht gebraucht, also das Feld „Adresse3“.

In der Emulation wird der Feld „Prüfsumme“ am Ende jedes Rahmens nicht eingefügt, da die Erkennung von Übertragungsfehler durch die „CRC-Check“-Funktion von dem realen kabelgebundenen Netzwerk erledigt wird.

Wie es in Abschnitt 4.3.1 angegeben wird, werden die Funktionen von Fragmentierung, Verschlüsselung und Stromsparen in der Emulation nicht unterstützt. Die entsprechenden Felder in Rahmenkopf dürfen auch weggelassen werden. Da es kein DS in Ad-hoc-Netzwerk existiert, enthält der Rahmenkopf kein „An DS“ und „Aus DS“.

Das Feld „Dauer“ im Rahmenkopf ist die wichtigste Besonderheit in IEEE 802.11 MAC. „Dauer“ enthält die voraussichtliche Zeitdauer der nachfolgenden Übertragung. Die Netzwerkteilnehmer, bis auf den Sender, empfangen einen Rahmen und entnehmen dementsprechend, dass sich eine Übertragung auf Kanal befindet, und das Medium wird für diesen Zeitraum reserviert (siehe folgende Kapitel).

Dafür wird ein einheitliches Rahmenformat in der Emulation vereinbart, wie die Abbildung 4-4 darstellt.

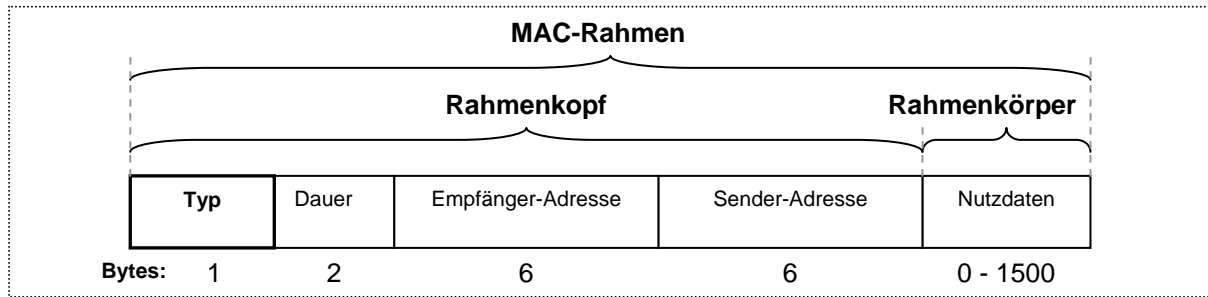


Abbildung 4-4 Allgemeines Rahmenformat in WILEMU

Wobei, statt 2312 Bytes wird die maximale Länge der Nutzdaten auf 1500 Bytes reduziert, weil die Rahmen mittels Ethernet übertragen werden und die maximale Länge der Nutzdaten in Ethernet auf 1500 Bytes beschränkt wird.

Die Formate von ACK, RTS und CTS, die ursprünglich in IEEE 802.11 definiert sind, werden in Abbildung 3-9 dargestellt. Sie bestehen nur aus einem MAC-Kopf und einem FCS.

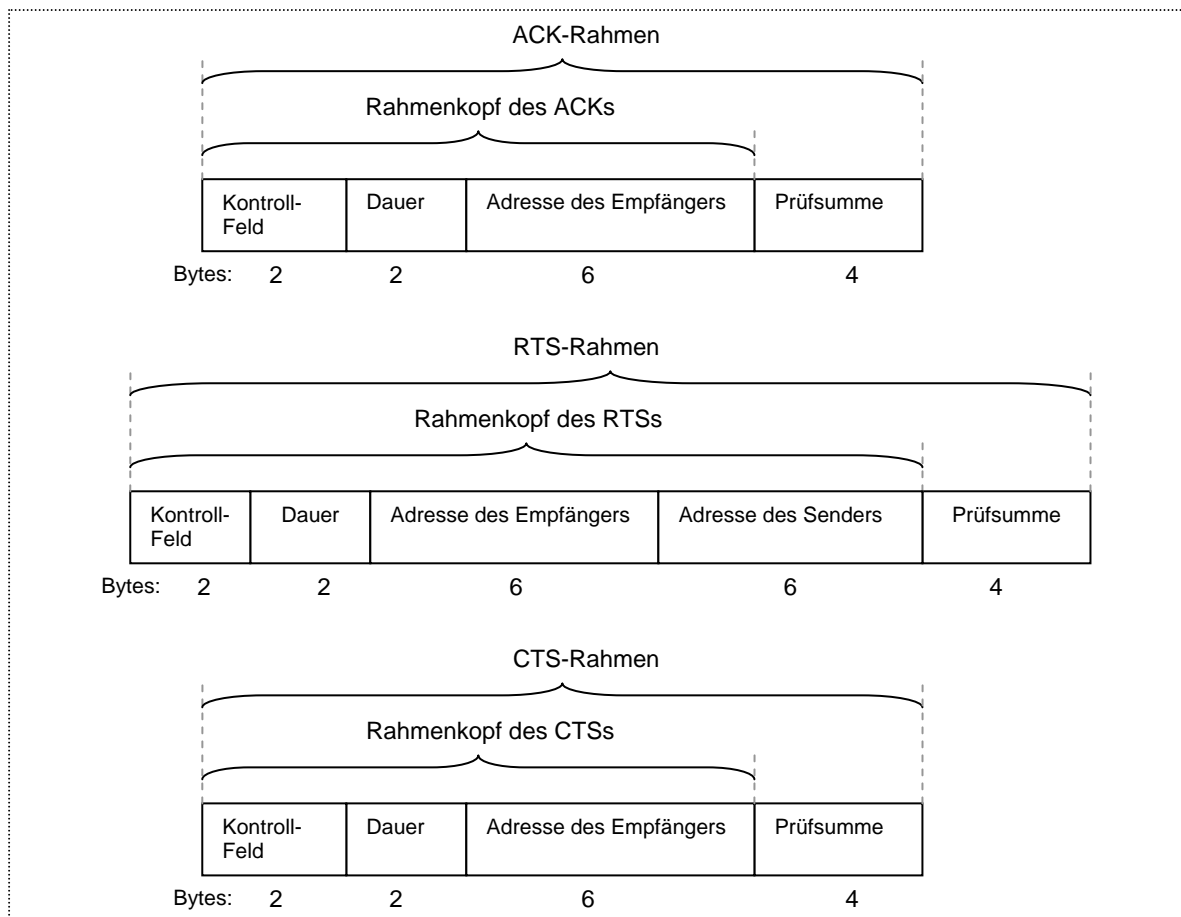


Abbildung 4-5 Rahmensformate von ACK, RTS und CTS nach IEEE 802.11 [IEEE811]

Zur Vereinfachung der Implementierung definieren wir die drei Steuerungs-Rahmen in einem einheitlichen Format. Wie auch im vereinfachten Format des Datenrahmens wird die Prüfsumme weggelassen. Aus diesem Format wird das Format des Steuerungsrahmens wie die Abbildung 3-9 gebaut.

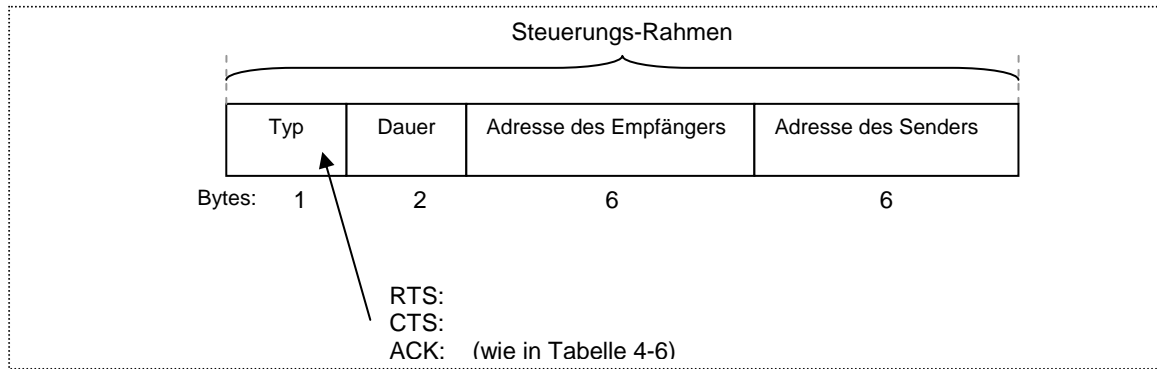


Abbildung 4-6 Format des Steuerungsrahmens in WILEMU

4.3.3 BackOff-Funktion

Wie bei CSMA/CD wird der Wettbewerb-Prozess durch die Binär-Exponential-BackOff-Funktion (vgl. Abschnitt 2.4.5) realisiert. Eine Zufallszahl für die BackOff-Wartezeit wird dadurch generiert. Allerdings sind die minimale und maximale Größen des Wettbewerb-Fensters (CW, Contention Window) in CSMA/CA anders als in CSMA/CD definiert, wie es in Abbildung 4-7 CW in CSMA/CA [IEEE811] dargestellt ist. Die entsprechenden Werte der BackOff-Wartezeit sind in Tabelle 4-8 aufgelistet.

$$\text{BackOff-Wartezeit} = \text{Zufallszahl} \times \text{Zeitschlitz} \\ (\text{Backoff Time}) \quad (\text{Random}) \quad (\text{aSlotTime})$$

1. Zufallszahl ist durch BackOff-Funktion generiert. Das Intervall der Zufallszahl ist $[0, CW]$.
2. $aSlotTime$ ist der Zeitschlitz in IEEE 802.11, wie die τ in CSMA/CD. Allerdings ist der Zeitschlitz abhängig von verwendeter Bitübertragungstechnik.
3. $CW_{min} \leq CW \leq CW_{max}$

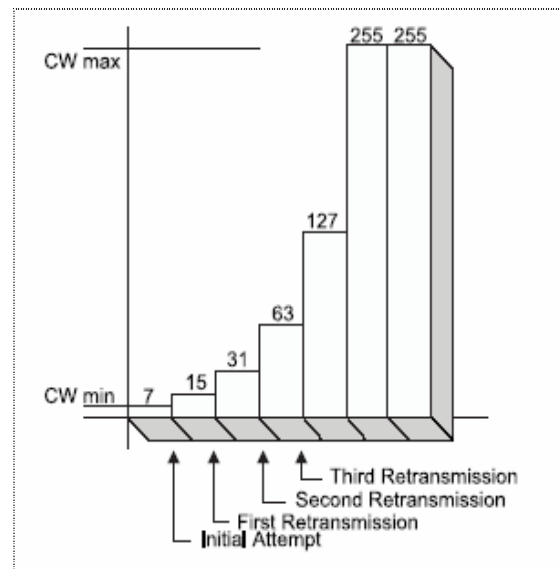


Abbildung 4-7 CW in CSMA/CA [IEEE811]

| Versuchszahl der erneuten Übertragung | Intervall der Zufallszahl $[0, CW]$ | Mögliche Wartezeit |
|---------------------------------------|-------------------------------------|--|
| 1 | $[0, 1, 2 \dots 7]$ | 0, aSlotTime ... 7 aSlotTime |
| 2 | $[0, 1, 2 \dots 15]$ | 0, aSlotTime ... 15 aSlotTime |
| | | |
| n, n in $[1, 6]$ | $[0, 1, 2 \dots 2^{(n+2)} - 1]$ | 0, aSlotTime ... $(2^{(n+2)} - 1)$ aSlotTime |
| n, n in $(6, 8]$ | $[0, 1, 2, 3 \dots 255]$ | 0, aSlotTime ... 255 aSlotTime |

Tabelle 4-8 Werte der BackOff-Wartezeit in CSMA/CA

Der Wert des Zeitschlitzes (aSlotTime) ist abhängig von Bitübertragungstechnik und ist in Tabelle 4-8 zu sehen.

| Bitübertragungstechnik | Wert des Zeitschlitzes |
|------------------------|------------------------|
| IR | 8 μ s |
| FHSS | 50 μ s |
| DSSS | 20 μ s |
| OFDM | 9 μ s |

Tabelle 4-9 Werte der Zeitschlitz in CSMA/CA [IEEE811], [IEEE811b]

Während des Wettbewerbs, wenn der Kanal wegen einer anderen Sendung als besetzt erkannt wird, wird der BackOff-Timer angehalten. Bei dem nächsten Wettbewerb wird der BackOff-Timer von dem letzten Stand an weiter abgezählt. Dabei wird die Zufallszahl nicht erneut generiert. Diese Vorgehensweise vermeidet, dass eine Sendestation wegen der Erkennung von Kanalbelegung zu lang warten muss. Es ist eine Besonderheit in CSMA/CA.

Der Wert des CWs wird bei jedem erfolglosen Sendversuch exponentiell vergrößert. Erreicht das CW CWmax, bleibt es bei CWmax. Nach der erfolgreichen Sendung wird der CW-Wert bei der Sendstation auf CWmin zurückgesetzt. Folgende Situationen können zu einer erfolglosen Sendung führen:

1. Der RTS- oder Daten-Rahmen ist durch eine andere Datenübertragung beschädigt worden, bzw. eine Kollision ist aufgetreten. Der Empfänger wird keine Bestätigung bzw. ein CTS-Rahmen zurückschicken.
2. Während SIFS-Zeit ist der Kanal aus der Sicht des Empfängers als besetzt erkannt, der Empfänger hat keinen Zugriff auf Kanal.
3. Obwohl der Empfänger einen CTS bzw. eine Bestätigung gesendet hat, ist der ACK-Rahmen in eine Kollision geraten. Daher wird der Sender nicht über die aktuelle Situation informiert.

In der Emulation muss jede Station zwei lokale Variablen haben: „tw“ und „collision“.

tw: Wenn der BackOff-Timer angehalten wurde, wird die Verbleibende BackOff-Wartezeit in „tw“ gespeichert. Bei dem nächsten Wettbewerb wird dieser Wert als Anfangswert geholt und zählt der BackOff-Timer herunter.
(tw: „Time to Wait“, Wartezeit, 64 Bit-Integer)

collision: Bei jedem erfolglosen Sendversuch wird sich der Wert des „collision“ erhöhen. Erreicht das „collision“ den maximalen Wert, wird eine Fehlermeldung an die höhere Schicht geliefert. Auf der Basis von „collision“ wird eine Zufallszahl für den Wettbewerb generiert.
(collision: Anzahl des Sendversuchs, 32 Bit-Integer)

Da der Zeitschlitz abhängig von Bitübertragungsschicht ist, soll der Wert des Zeitschlitzes als Parameter einstellbar sein.

4.3.4 Trägererkennung

Die Trägererkennung in CSMA/CA besteht aus der physischen Erkennung und der virtuellen Erkennung. In NET-Umgebung (Emulationsumgebung unter kabelgebundenen LAN) ist die physische Trägererkennung nur teilweise realisierbar.

Wie in Abschnitt 4.2.3 erläutert, sind die Signalausbreitungsverzögerungen in Emulationsumgebung und realer Umgebung vergleichbar. Damit nehmen wir weiter an:

Annahme2:

Der Zeitpunkt, zu dem die Emulationssoftware einen Rahmen empfangen hat, ist der Anfangszeitpunkt, ab dem der Kanal belegt ist.

Wie die Abbildung 3-9 zeigt, wird eine Rahmensankunft bei Emulationssoftware erkannt, nur nachdem die NET-Hardware den Rahmen erfolgreichen empfangen hat. D.h. WILEMU empfängt nur fehlerfreie Rahmen. Wenn die NET-Hardware festgestellt hat, dass der Rahmen fehlerhaft ist, wird dieser Rahmen weggeworfen, und die Software (WILEMU), kann diese Kanalbelegung überhaupt nicht wahrnehmen.

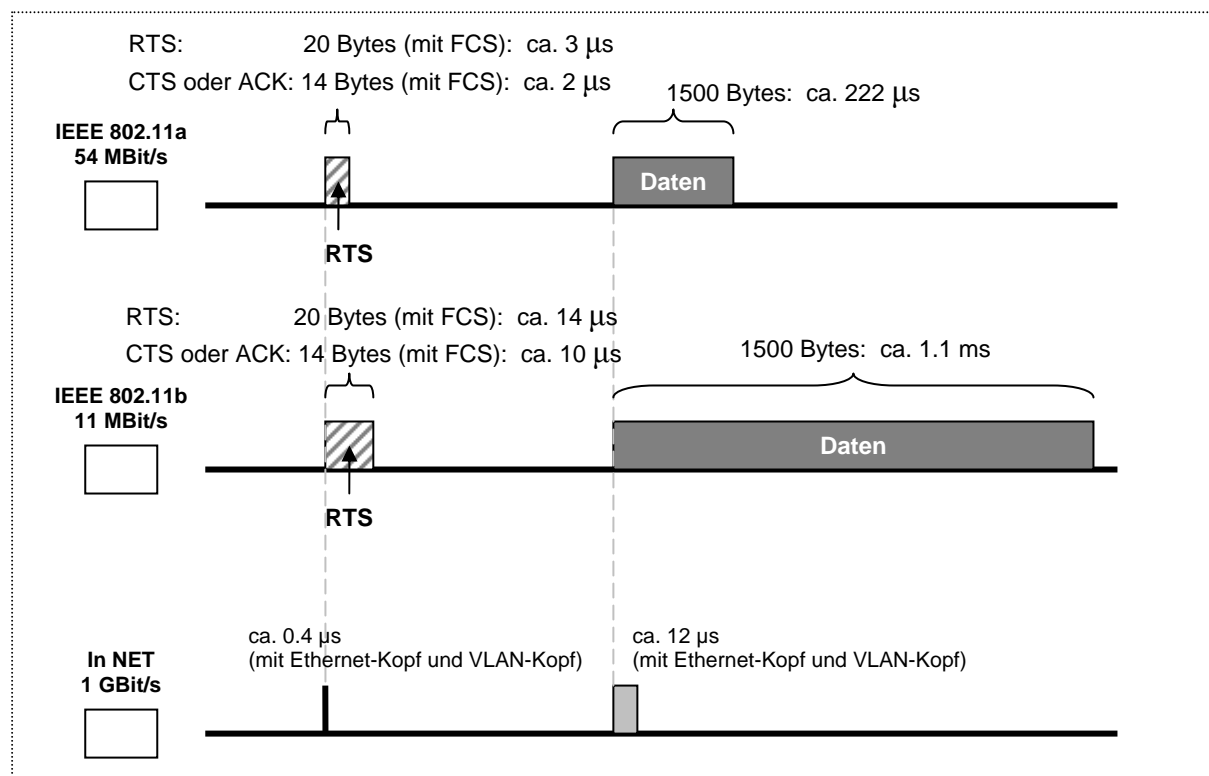


Abbildung 4-8 Vergleich der Übertragungsdauer

Wie die Abbildung 4-8 zeigt, ist die Zeitdauer für die Übertragung eines Rahmens in der Emulationsumgebung wesentlich kürzer als es in realer Umgebung. Um die richtige Kanalbelegung zu emulieren, muss die Übertragungsdauer künstlich addiert werden, also der Zeitpunkt des Endes der Kanalbelegung = Anfang der Kanalbelegung + Übertragungsdauer des Rahmens in realer Umgebung.

In der Emulation wird der Kanalzustand durch die Variablen „tbs“ und „tbe“ emuliert.

tbs: Anfangszeitpunkt der Kanalbelegung
(tbs, „Time Busy Start“, 64 Bit-Integer)

tbe: Zeitpunkt des Endes der Kanalbelegung
(tbe, „Time Busy End“, 64 Bit-Integer)

Damit kann auch die virtuelle Trägererkennung emuliert werden. Wenn die Stationen, die sich nicht an der Kommunikation beteiligen, also die Stationen außer Sender und Empfänger, einen Rahmen empfangen haben, werden ihre „tbs“ und „tbe“ so aktualisiert:

$tbs = \text{Zeitpunkt der Rahmenankunft}$

$tbe = tbs + \text{Übertragungsdauer in MAC-Kopf}$

4.3.5 Granularität des Timers

Wie die Tabelle 4-8 und Tabelle 4-9 zeigen, liegt die BackOff-Zeit im Mikrosekunden-Bereich. Das Emulationswerkzeug „WILEMU“ funktioniert als ein Linux-Modul und benutzt den Linux-Kernel-Timer. Die Takt-Periode dieses Timers (in Standard-Linux-Kernel, mit Version 2.4) beträgt 10 Millisekunden. Mit dieser Granularität des Kernel-Timers kann der BackOff-Prozess in Wireless LAN überhaupt nicht emuliert werden, da die maximale BackOff-Wartezeit nur 13 ms ist (FHSS). In OFDM ist es sogar nur 2.3 ms. Mit einer Verfeinerung kann die Granularität des Timers bis zu 1 ms erreichen. Mit dieser Granularität ist die BackOff-Funktion immer noch nicht emulierbar, da die Kollisionen wesentlich häufiger als in realem Wireless LAN auftreten können. Dank der Arbeit von Robert Sauter ist ein APIC-Timer für die Emulation einsatzbereit. Der APIC-Timer verwendet den CPU-Takt. Mit diesem Timer ist die Granularität auf 10 μ s verbessert. In dem Entwurf und späterer Implementierung wird dieser APIC-Timer eingesetzt.

4.3.6 Interframe Space (IFS)

Ansgenhend von dem originalen RTS/CTS Algorithmus, der in IEEE 802.11 spezifiziert wird, wartet der Sender nun auf den CTS-Rahmen, nachdem er den RTS-Rahmen gesendet hat. Wenn der Sender den Daten-Rahmen abgeschickt hat, wie bei RTS/CTS, erwartet er eine Bestätigung. Nach dem erfolgreichen Empfang von RTS- und Daten-Rahmen muss der Empfänger die entsprechende CTS bzw. die Bestätigung an den Sender zurücksenden. Eine komplette Sendung soll möglichst schnell abgeschlossen werden. Damit der Sender die CTS und ACK sofort erhalten kann, soll höhere Priorität der Sendstation beim Zugreifen auf Kanal und beim Absenden des CTS und ACKs gewährleistet werden. Um dies zu ermöglichen, sind in IEEE 802.11 Standard verschiedene lange Intervalle IFS (Interframe Space) zwischen dem Senden von Rahmen definiert. Sie sind in Abbildung 3-9 dargestellt.

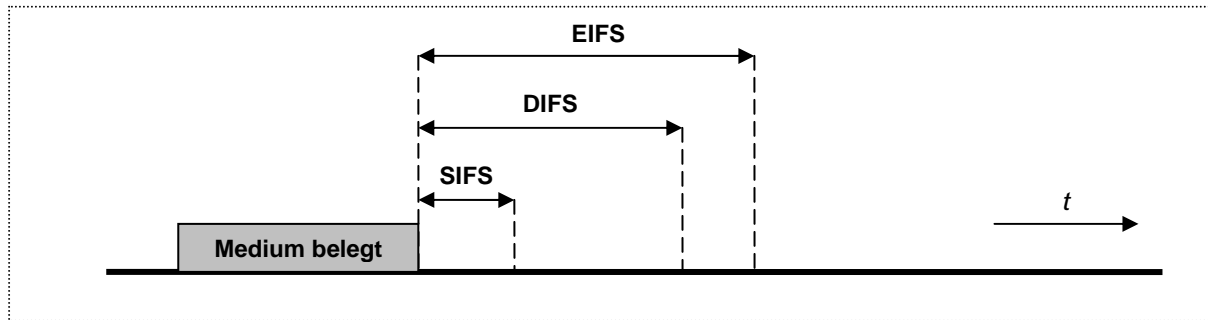


Abbildung 4-9 IFS-Intervalle in IEEE 802.11 DCF [IEEE811]

1. **SIFS** (Short IFS):
SIFS ist vor dem Senden von CTS, Daten und ACK benutzt. SIFS ist das kleinste IFS. D.h. die Sendung von CTS und ACK hat höhere Priorität.
2. **DIFS** (DCF IFS):
DIFS ist vor dem Senden von RTS- oder Verwaltungsrahmen verwendet. Wenn der Rahmen mit Broadcasting-Adresse ist, muss der Sender eine DIFS abwarten, statt SIFS zu benutzen.
3. **EIFS** (Extended IFS):
EIFS ist ein spezielles IFS. Wenn eine Station einen fehlerhaften Rahmen empfangen hat, ist die in dem Rahmenkopf enthaltene Dauer nicht mehr lesbar oder hat sie einen anderen Wert, der viel zu groß ist. Nach der Fehlererkennung, wird der NAV-Wert bei dieser Station durch EIFS aktualisiert, statt der Wert in Rahmenkopf zu benutzen.

Wie der Zeitschlitz, sind die IFS-Werte auch abhängig von der verwendeten Übertragungstechnik. Die Werte von den IFSs sind in Tabelle 4-10 aufgelistet.

| Bitübertragungstechnik | SIFS | DIFS | EIFS |
|------------------------|------------|-------------|--------------------|
| IR | 10 μ s | 26 μ s | 36 μ s + aACK |
| FHSS | 28 μ s | 128 μ s | 156 μ s + aACK |
| DSSS | 10 μ s | 50 μ s | 60 μ s + aACK |
| OFDM | 16 μ s | 34 μ s | 50 μ s +aACK |

Tabelle 4-10 IFS in IEEE 802.11 [IEEE811], IEEE811b]

$$DIFS = SIFS + 2 aSlotTime$$

$$EIFS = SIFS + DIFS + aACK$$

$aACK$ = Übertragungsdauer des ACK-Rahmens inklusive PHY-Kopf (z.B. PLCP-Kopf)
 $aACK$ ist abhängig von der Übertragungsrate. Je größer die Übertragungsrate, ist desto kleiner EIFS..

Die Zeitberechnung in DCF (CSMA/CA) in IEEE 802.11 ist in Abbildung 3-9 dargestellt.

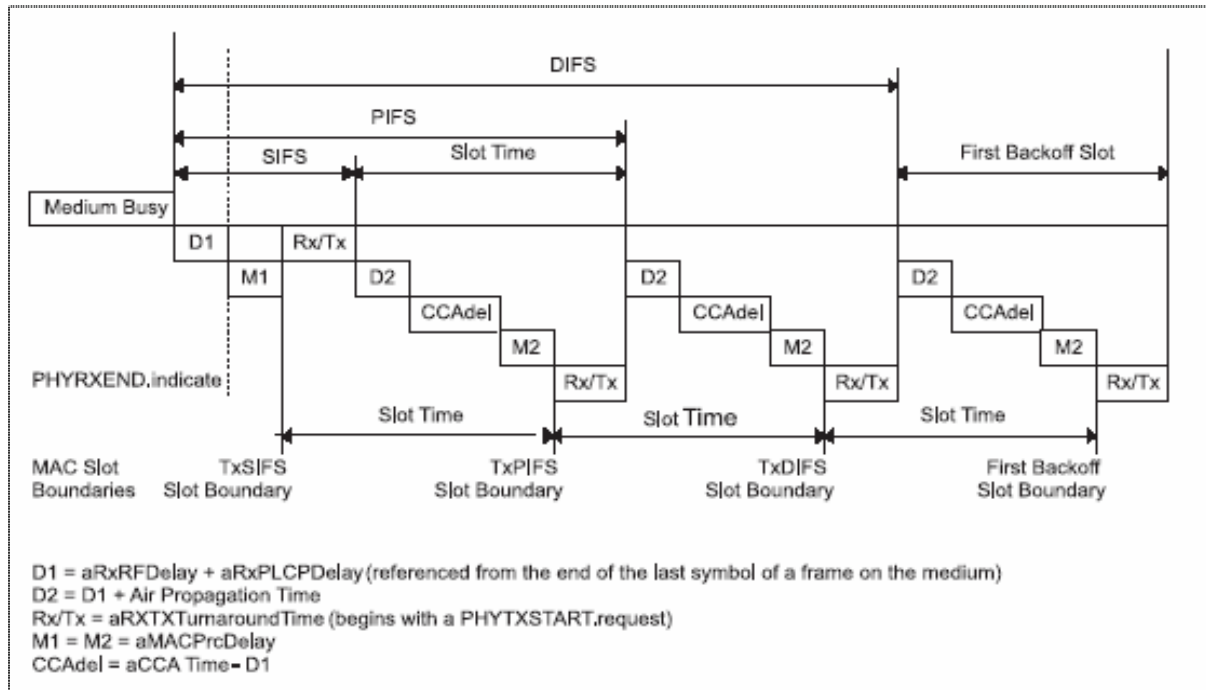


Abbildung 4-10 Zeitberechnung in IEEE 802.11 DCF [IEEE811]

Die SIFS ist die Zeitdauer vom Zeitpunkt, zu dem ist das letzte Bit des RTS-Rahmens (oder Daten-Rahmen, CTS-Rahmen) angekommen ist, bis dem Zeitpunkt, zu dem die Mobilstation anfängt, den nachfolgende Rahmen auf Medium zu senden.

Wegen der Abhängigkeit zwischen Übertragungstechnik und der Größe des IFS, müssen die IFSs in der Emulation als Parameter einstellbar sein. Die IFS in CSMA/CA werden in WILEMU wie folgt betrachtet (vgl. Tabelle 4-10):

| IFS | Einsetzbarkeit | Beschreibung |
|------|----------------|--|
| SIFS | Nein | Der minimale Wert des SIFS beträgt 10 μ s. Im originalen CSMA/CA-Algorithmus wartete eine Station für SIFS-Zeit nach Erhalt von RTS. Durch den Einsatz von APIC-Timer (Granularität 10 μ s) kann SIFS nicht realisiert werden. Zur Vereinfachung und auch wegen der Granularität des APIC-Timers wird ein CTS sofort abgeschickt, statt SIFS-Zeit abzuwarten. Die Konsequenzen werden im folgenden Kapitel diskutiert. |
| DIFS | Ja | Der minimale Wert des DIFS beträgt 26 μ s. Durch den Einsatz von APIC-Timer kann DIFS realisiert werden. |
| EIFS | Nein | Die NET-Hardware liefert nur fehlerfreie Rahmen an WILEMU. |

Tabelle 4-11 IFS in WILEMU

4.4 Entwurf der WILEMU

Nach der detaillierten Analyse von CSMA/CA und den oben erläuterten Annahmen und Vereinfachungen ist die WILEMU grundsätzlich durch Zustandmaschinen gesteuert. In diesem Kapitel werden noch weitere Vereinfachungen erläutert.

4.4.1 Definitionen von WILEMU-Zuständen und Lokalen Variablen

In WILEMU sind 9 Zustände definiert:

| Zustand | Beschreibung | Wert |
|----------------|--|------|
| IDLE | Ruhe-Zustand | 0 |
| CONTENTION | Wettbewerb -Zustand oder beim Abwarten von DIFS | 1 |
| LISTENING | Station hat kein Zugriff auf Kanal, da der Kanal wegen einer anderen Übertragung gesperrt ist. | 2 |
| WAITING_CTS | Station hat RTS abgesendet, nun wartet sie auf CTS. | 3 |
| WAITING_ACK | Station hat Daten-Rahmen abgesendet, nun wartet sie auf Bestätigung (ACK). | 4 |
| WAITING_DATA | Station hat CTS abgesendet, nun wartet sie auf dem Daten-Rahmen. | 5 |
| RECEIVING_DATA | Station empfängt gerade einen Daten-Rahmen. | 6 |
| BROADCASTING | Station ist beim Senden eines Rahmens mit Broadcast-Adresse. | 7 |
| RECEIVING_BC | Station empfängt gerade einen Daten-Rahmen mit Broadcast-Adresse. | 8 |

Tabelle 4-12 Definition der WILEMU-Zuständen

Jede Station besitzt folgende lokalen Variablen:

| Variable | Beschreibung | Datenformat |
|----------------|---|----------------|
| status | Zustand der Station | 32 Bit-Integer |
| t | Lokale Zeit | 64 Bit-Integer |
| tbs | Anfangszeitpunkt des Kanalbelegung | 64 Bit-Integer |
| tbe | Zeitpunkt des Endes des Kanalbelegung | 64 Bit-Integer |
| tw | Verbleibende BackOff-Wartezeit | 64 Bit-Integer |
| collision | Anzahl der Sendversuchs | 32 Bit-Integer |
| active | Der gerade sendende oder empfangene Daten-Rahmen | Zeiger |
| lossratio_list | Die Fehlerrateliste. Ein Eintrag davon ist ein Paar von MAC-Adresse des Senders und der entsprechenden Fehlerrate. Z.B.: (00-50-56-C0-05-04, 1) (00-50-56-C0-05-2F, 0) | Liste |

Tabelle 4-13 Lokale Variablen in WILEMU

Jede Station verwaltet außerdem noch einen Timer und eine Warteschlange. Die Warteschlange ist nur für das Senden benutzt und funktioniert nach FIFO-Regel. Der Timer steuert den Wettbewerb-Prozess und die verzögerte Datenlieferung. Die Benutzung und Funktionsweise werden in nachfolgenden Zustandsdiagrammen erklärt.

4.4.2 Zustandsdiagramme

4.4.2.1 „IDLE“

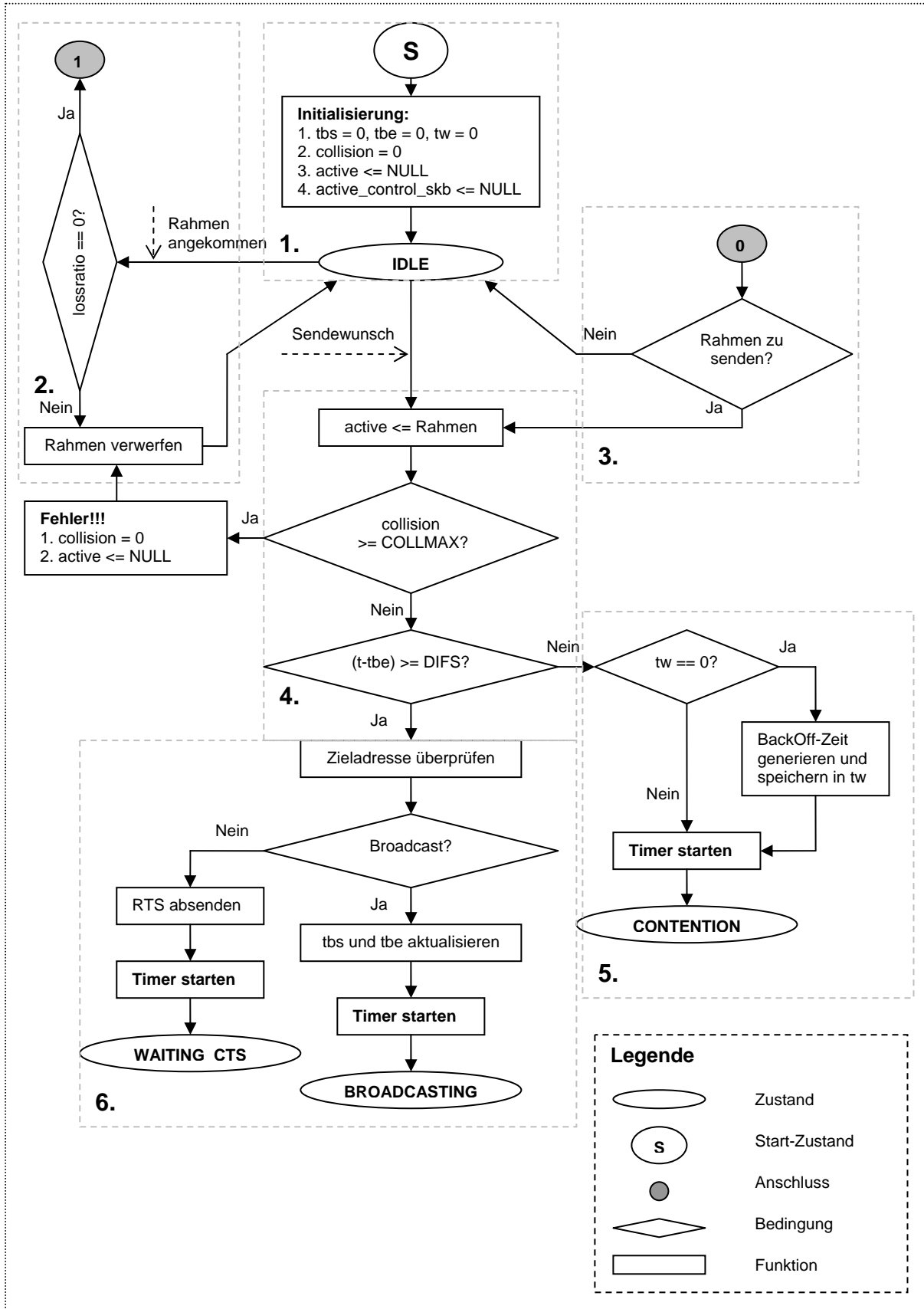


Abbildung 4-11 Zustandsdiagramm - „IDLE“

Erklärungen zum Zustand „IDLE“:

1. Nach der Initialisierung ist eine Station zum Senden und Empfang bereit.
2. Station hat einen Rahmen von der NET-Hardware bekommen, muss sie aber zuerst überprüfen, ob sie innerhalb der Reichweite des Senders liegt (vgl. Abschnitt 4.2.2). Wenn ja, ist dieser ein gültiger Rahmen. Ein Rahmenempfang darf gestartet werden (siehe Abbildung 4-15). Wenn nein, wird dieser Rahmen weggeworfen und der Zustand bleibt unverändert, da die Station diesen Rahmen in realer Umgebung nicht empfangen kann.
3. Wenn WILEMU von einem Zustand außer „IDLE“ zum Ruhe-Zustand „IDLE“ übergehen will, muss sie zuerst die Warteschlange überprüfen. Wenn die Warteschlange nicht leer ist, fängt WILEMU an, den Rahmen zu senden. (siehe nachfolgende Diagramme)
4. Nach Sendewunsch verweist die Variable „active“ auf den zu sendenden Rahmen. Wenn die maximale Anzahl des Sendversuchs (Nach jedem erfolglosen Sendversuch erhöht sich der Wert der Variable „collision“.) erreicht wird, gibt WILEMU der höheren Schicht an, dass der Rahmen nicht mehr gesendet werden kann. Genau wie im originalen CSMA/CA-Algorithmus, muss der Kanalzustand danach geprüft werden, ob ein schneller Zugriff erlaubt ist (vgl. Abschnitt 3.2 und Abbildung 3-3).
5. Wenn ein schneller Kanalzugriff nicht erlaubt ist, muss die Station um ein Zugriffsrecht konkurrieren. Wenn eine vorherige Sendung während des Wettbewerbs angehalten wurde, ist der alte Timerstand in „tw“ gespeichert, d.h. es noch „Verbleibende BackOff-Zeit“ gibt. Wenn der Wettbewerb erfolgreich abgeschlossen war, ist der Wert von „tw“ auf 0 gesetzt. Vor einem Wettbewerb muss zunächst der alte Timerstand überprüft werden. Wenn der Wert gleich 0 ist, heißt es, dass der alte BackOff-Timer abgelaufen war, muss eine neue BackOff-Wartezeit generiert werden. Ein Timer wird danach gestartet. Und geht die Station in Wettbewerbs-Zustand („CONTENTION“, vgl. die Erklärungen zur Abbildung 4-15 und zum Zustand „CONTENTION“). Nicht wie es im originalen CSMA/CA, zur Vereinfachung, wird das Abwarten für DIFS in Wettbewerb integriert. Damit ist die Ablaufzeit die Summe von DIFS und der generierten BackOff-Zeit („tw“), wie die Abbildung 4-15 zeigt:

$$\begin{aligned}tw &= t2 - t1 \\DIFS &= t1 - t0 \\T &= tw + DIFS = t2 - t0 \text{ (Ablaufzeit)}\end{aligned}$$

Wenn der Wettbewerb wegen einer Kanalbelegung angehalten werden würde, muss die verbleibende BackOff-Zeit in „tw“ gespeichert werden:

Wobei „t“ ist der Zeitpunkt (lokale Zeit), in dem der Timer gestoppt wurde.

Falls:

1. Timer wurde während DIFS angehalten, also, $t - t0 \leq DIFS$, dann $tw = 0$
2. Timer wurde während BackOff angehalten, also, $t - t0 > DIFS$, dann $tw = T + t0 - t$

Wenn der Timer abgelaufen ist, dann $tw = 0$.

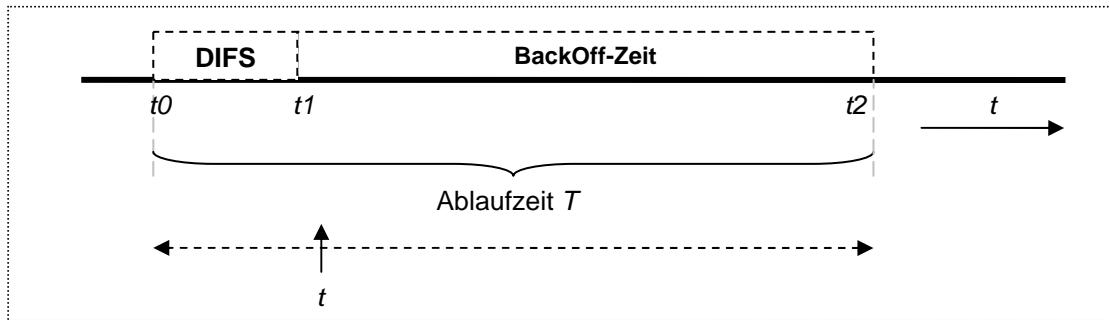


Abbildung 4-12 Zeitdiagramm – „CONTENTION“

6. Wenn ein schneller Kanalzugriff erlaubt ist, darf die Station mit dem Senden anfangen. Allerdings muss sie zunächst überprüfen, ob der Rahmen mit Broadcast-Adresse ist. Weil der RTS/CTS-Mechanismus das Broadcasting nicht unterstützt, soll dieser Daten-Rahmen, der eine Broadcast-Adresse als Zieladresse besitzt, sofort abgesendet werden. Anschließend müssen die Variablen „tbs“ und „tbe“ aktualisiert und ein Timer gestartet werden, wie die Abbildung 4-15 zeigt. Die in Rahmenkopf eingefügte „Dauer“ ist genau die Übertragungszeitdauer dieses Daten-Rahmens in realer Umgebung (vgl. die Erklärungen zur Abbildung 4-15 und zum Zustand „BROADCASTING“).

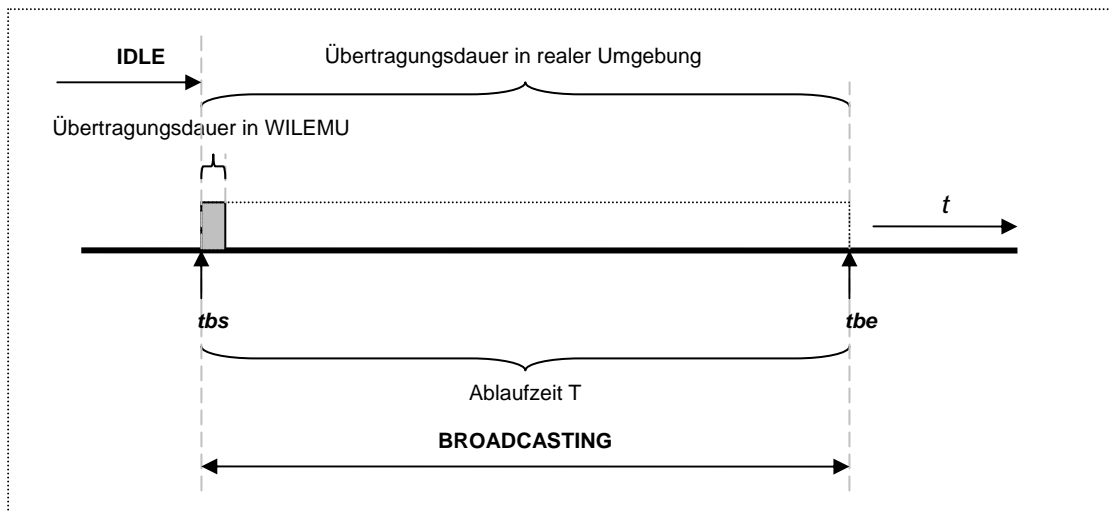


Abbildung 4-13 Zeitdiagramm - „BROADCASTING“

Wenn der Rahmen sich um ein „Unicast“-Rahmen handelt, muss zuerst ein RTS abgesendet werden. Wie die Abbildung 4-14 zeigt, wird gleichzeitig ein Timer gestartet. Die Ablaufzeit ist die Summe von SIFS und der realen Übertragungszeiten von RTS und CTS (vgl. Abbildung 4-8 und Tabelle 4-10). Dabei ist der Timer für die Erkennung von Ankunft und Kollision des CTS benutzt (vgl. die Erklärungen zur Abbildung 4-15 und zum Zustand „WAITING_CTS“).

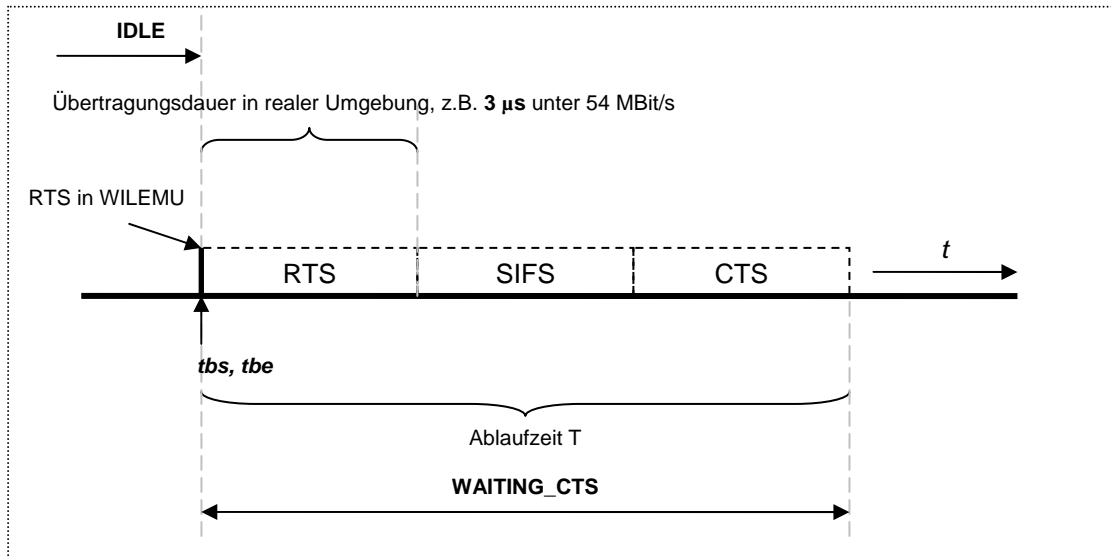


Abbildung 4-14 Zeitdiagramm - Senden des RTS und Warten auf CTS

4.4.2.2 „Anschluss 1“

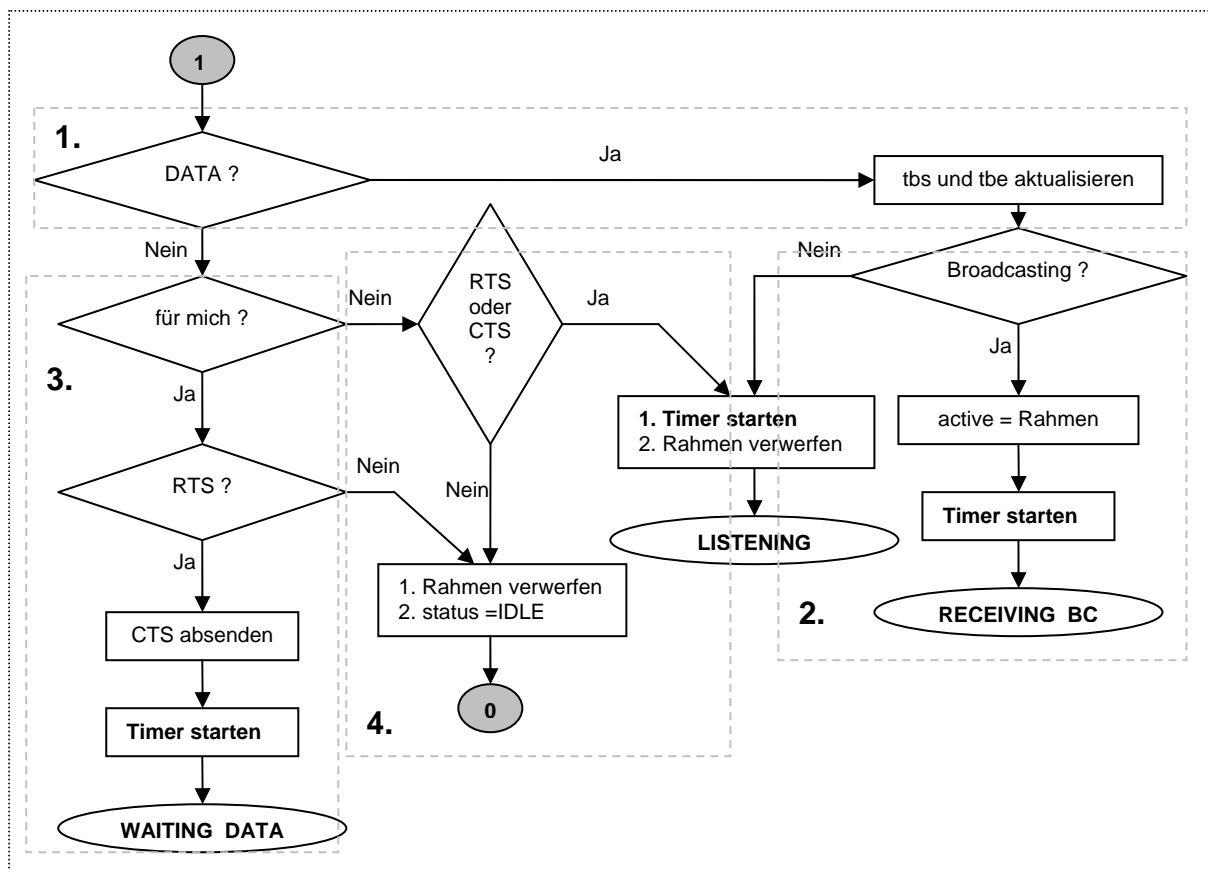


Abbildung 4-15 Zustandsdiagramm – Anschluss 1 (Empfang eines gültigen Rahmens)

Nun befindet eine Station im Zustand „IDLE“ und ist ein Rahmen nach der Reichweiteüberprüfung (nachschiessen in Fehlerrateliste) tatsächlich angekommen.

Erklärungen zum Anschluss 1:

1. Wenn der empfangende Rahmen ein Datenrahmen ist, müssen die Variablen „tbs“ und „tbe“ aktualisiert. Der Wert der „tbs“ ist der lokale Zeitpunkt, in dem der Rahmen angekommen ist.
2. Wenn der Rahmen sich um einen Broadcast-Rahmen handelt, geht die Station sofort in die Empfangsphase, wie die Abbildung 4-16 zeigt. Da der Daten-Rahmen eine Länge bis zum 1500 Bytes haben kann, darf die Übertragungsdauer in der Emulationsumgebung nicht vernachlässigen. Dabei ist die Ablaufzeit die Differenz von Übertragungsdauern in realer Umgebung („Dauer“ in Rahmenkopf) und in WILEMU:

$$T = \text{„Dauer“} - \text{temu}$$

$$\text{temu} = \text{Länge des Rahmens} / \text{Übertragungsrates in NET(1 GBit/s)}$$

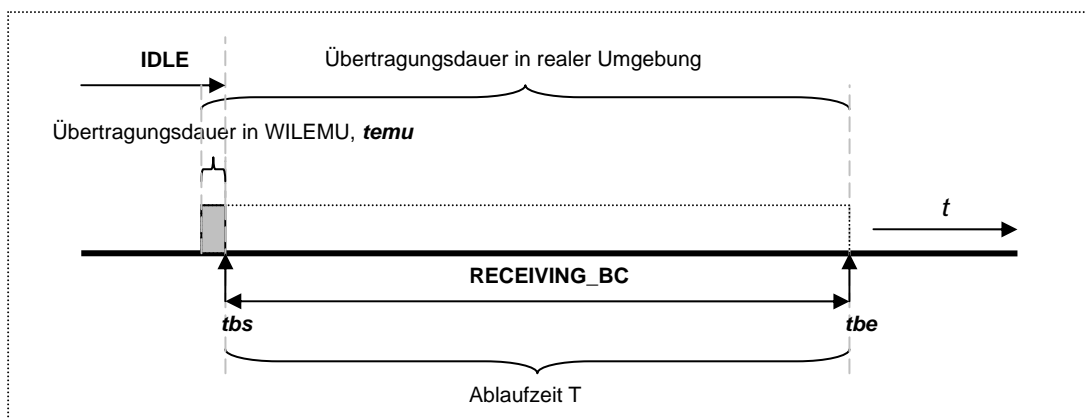


Abbildung 4-16 Zeitdiagramm – „RECEIVING_BC“ (Empfang eines Broadcast-Rahmens)

Wenn der Rahmen sich um ein „Unicast“-Datenrahmen handelt, geht die Station sofort in den Zustand „LISTENING“, wie die Abbildung 4-17 zeigt. Im Zustand „LISTENING“ hat Station kein Zugriff auf Medium.

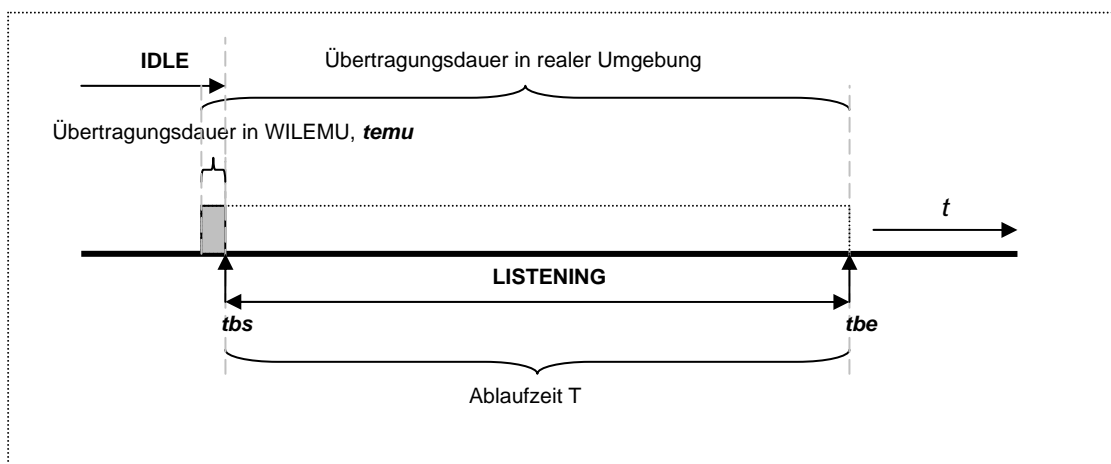


Abbildung 4-17 Zeitdiagramm – „LISTENING“

Dabei unterscheiden sich zwei Situationen:

- WILEMU ist die Zielstation: Dieser Datenrahmen ist ein fehlerhafter oder verspäteter Datenrahmen, da WILEMU sich im Moment in Zustand „IDLE“ befindet und keine Daten angefordert hat. Da der Kanal wegen dieser Datenübertragung besetzt ist, geht WILEMU auch zum „LISTENING“-Zustand über.
 - WILEMU ist keine Zielstation: Dieser Rahmen ist ein vorbeilaufender Rahmen für eine andere Station. WILEMU geht in den „LISTENING“-Zustand.
3. Wenn WILEMU erkannt hat, dass dieser Rahmen ein RTS und sie die Zielstation ist, muss WILEMU diesen Rahmen empfangen. Wie die Abbildung 4-8 und Abbildung 4-15 gezeigt haben, ist die Übertragungsdauer des RTS in Emulationsumgebung sehr kurz. Zur Synchronisation und Erkennung von Kollision mit RTS sollte ein Timer mit Ablaufzeit von Übertragungsdauer des RTS in realer Umgebung gestartet werden. Obwohl der APIC-Timer eingesetzt wird, ist diese künstliche Verzögerung nicht realisierbar. Nach dem originalen CSMA/CA-Algorithmus muss die Station nach der Erhalt eines RTS einen CTS schicken. Dazwischen ist noch noch SIFS-Zeit abzuwarten. Wegen der begrenzten Timer-Auflösung ist das Abwarten auch nicht realisierbar (vgl. Tabelle 4-11). Die Nachbildung von CSMA/CA wird weiter vereinfacht:

Vereinfachung

- **Die Übertragungsdauer des Steuerungsrahmens wird in der Emulation auf der Empfangsseite ignoriert.**
- **Vor dem Senden von aufeinander folgenden Rahmen (RTS->CTS->DATA ->ACK) wird kein SIFS gewartet.**

Konsequenzen

- **Die Kanalbelegung bei Senden und Empfang des Steuerungsrahmens wird in der Emulation vernachlässigt, d.h. die Variablen „tbs“ und „tbe“ werden nicht aktualisiert.**
- **Für RTS, CTS und DATA beträgt die in Rahmenkopf eingefügte „Dauer“ die reale Übertragungsdauer des Daten-Rahmens.**
- **Für ACK beträgt die in Rahmenkopf eingefügte „Dauer“ 0.**

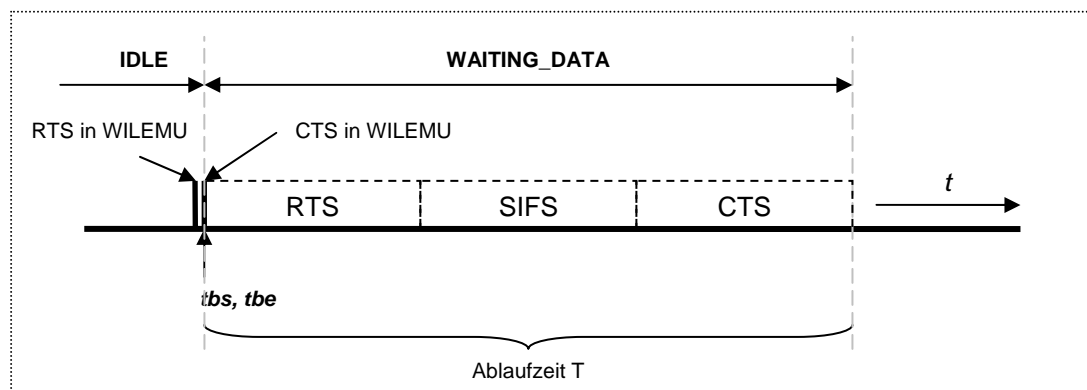


Abbildung 4-18 Zeitdiagramm - Empfang des RTS und Absenden von CTS

Wie die Abbildung 4-18 zeigt, nach der oben erklärten Vereinfachung, wird ein CTS sofort gesendet und ein Timer gleichzeitig gestartet.

4. Wenn der Rahmen ein vorbeilaufender Steuerungsrahmen ist, geht die Station in den Zustand „LISTENING“ (vgl. Abbildung 4-17). Dabei ist die Ablaufzeit des Timers die „Dauer“ in MAC-Kopf. Alle anderen Rahmen (alle ACK und CTS an diese Station) werden weggeworfen.

4.4.2.3 „CONTENTION“

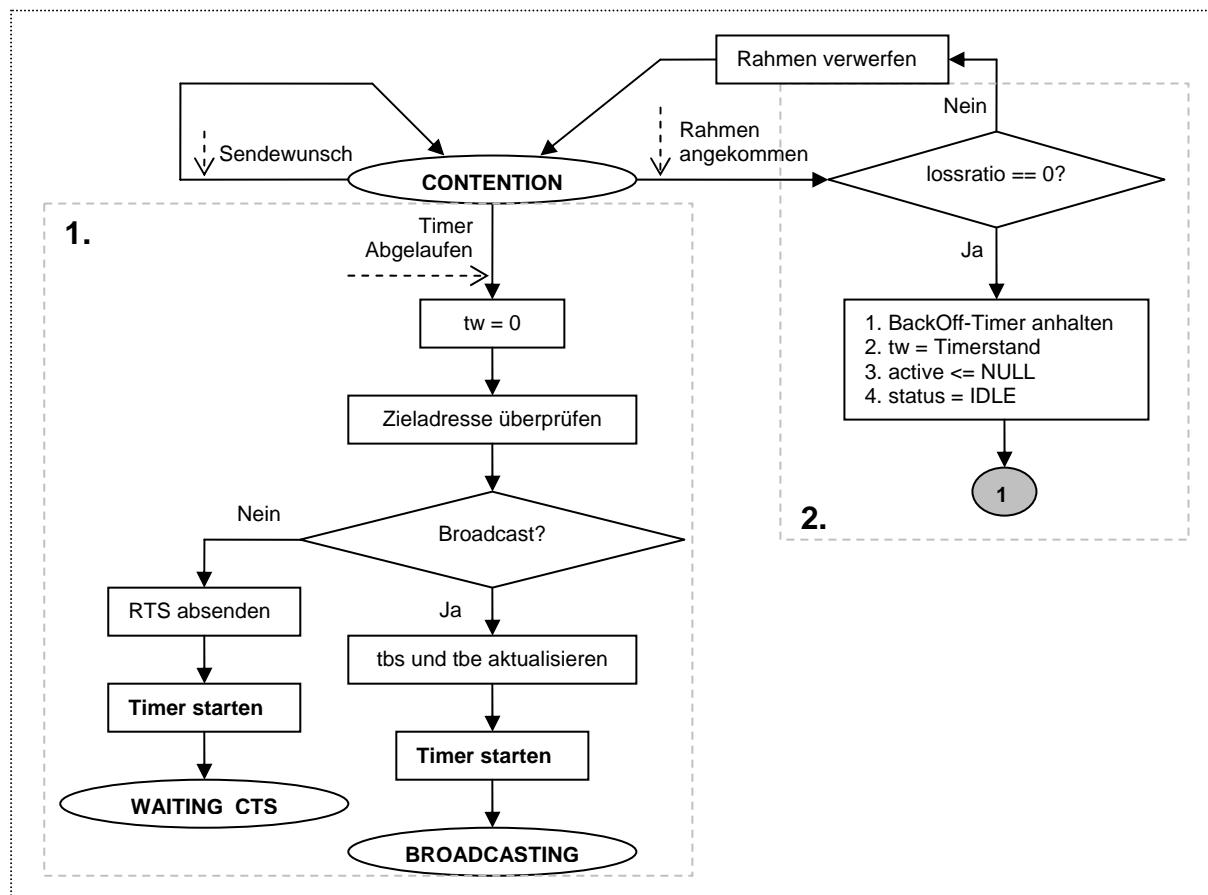


Abbildung 4-19 Zustandsdiagramm – „CONTENTION“

Nun befindet eine Station im Zustand „CONTENTION“ (Wettbewerb). In diesem Zustand darf nicht gesendet werden.

Erklärungen zum Zustand „CONTENTION“:

1. Ist der Timer abgelaufen, heißt es, dass der Wettbewerb abgeschlossen ist, und die Station darf auf den Kanal zugreifen. „tw“ muss auf 0 gesetzt werden, um anzugeben, dass die BackOff-Wartezeit für nächstes Senden erneut generiert werden muss. (Weitere Schritte siehe die Erklärung zur Abbildung 4-11, Punkt 6.)
2. Ist ein Rahmen während Wettbewerb angekommen, muss die Gültigkeit dieses Rahmens getestet werden. Wenn ja, hält der Timer sofort an, d.h. der Wettbewerb-Vorgang wird abgebrochen. Danach wechselt die Station zum Zustand „IDLE“ und beginnt mit dem

Empfang dieses neuen gekommenen Rahmens (siehe die Abbildung 4-15 und die Erklärung dazu).

4.4.2.4 „LISTENING“

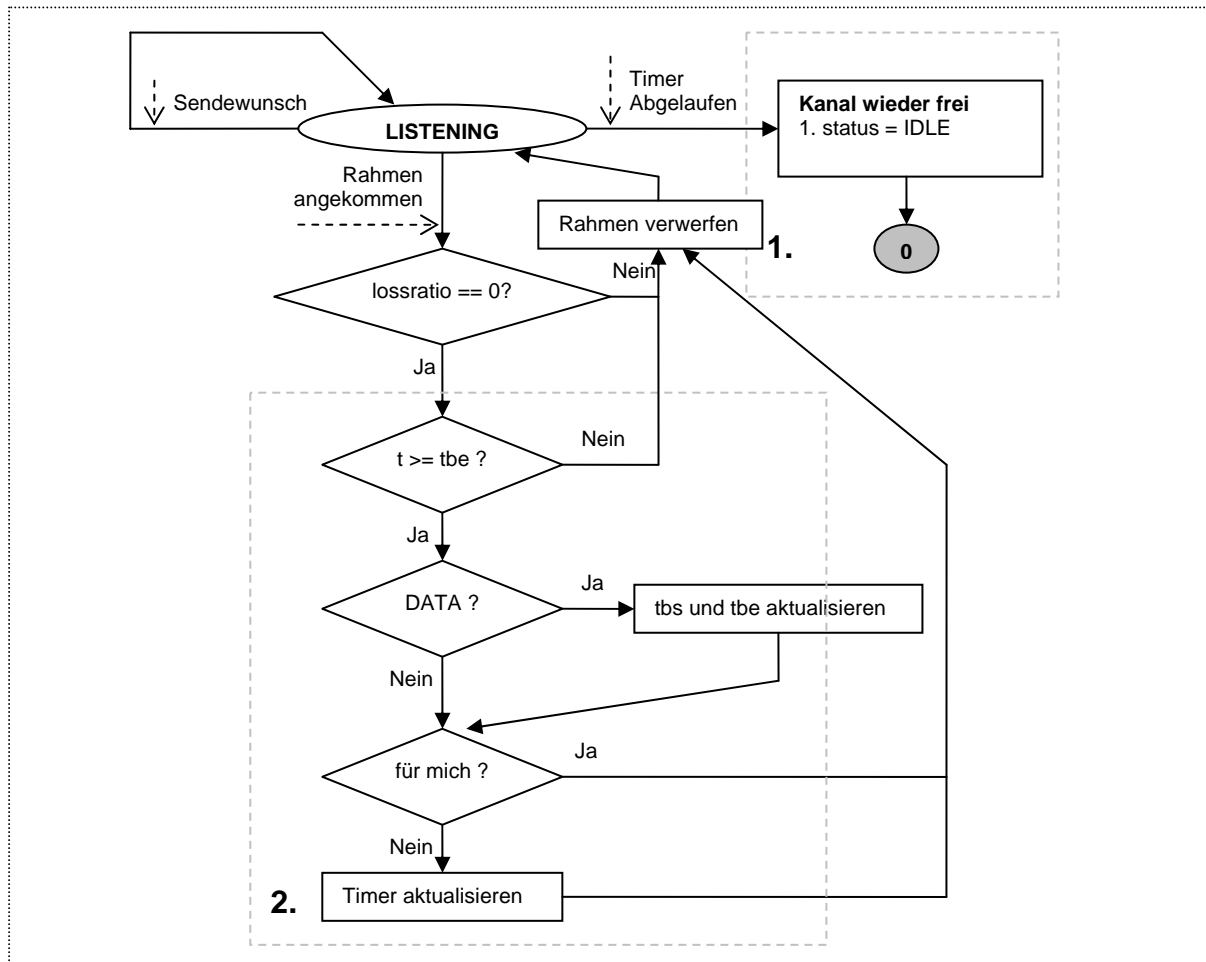


Abbildung 4-20 Zustandsdiagramm – „LISTENING“

Nun befindet eine Station im Zustand „LISTENING“. In diesem Zustand hat Station keinen Zugriff auf das Medium.

Erklärungen zum Zustand „LISTENING“:

1. Ist der Timer abgelaufen, heißt es, dass der Kanal wieder frei ist. Station überprüft danach, ob ein Rahmen in der Warteschlange steht. Wenn nein, geht die Station in den Zustand „IDLE“. Wenn ja, beginnt sie sofort mit dem Senden (siehe Abbildung 4-11, Block 3 und 4).
2. Ist ein gültiger Rahmen (nach Fehlerrateliste) empfangen wurde, muss noch überprüft werden, ob dieser Rahmen kollidiert worden ist. Wenn die Empfangszeit „t“ kleiner als „tbe“ ist, heißt das, dass der Kanal im Moment belegt ist und dieser Rahmen nicht richtig empfangen werden kann. Wenn es nicht der Fall ist, ist dieser Rahmen erfolgreich empfangen worden. Die Zieladresse und der Typ dieses Rahmens muss noch danach geprüft werden. Der Timer wird durch die in MAC-Kopf enthaltene „Dauer“ aktualisiert, wenn der Rahmen ein vorbeilaufender (Zieladresse dieses Rahmens ist nicht der MAC-Adresse der Empfangsstation.)

Steuerungsrahmen ist. Bei Datenrahmen werden die Variable „tbs“ und „tbe“ neben der Aktualisierung des Timers auch noch gesetzt (vgl. die Abbildung 4-15 und Abbildung 4-17).

4.4.2.5 „WAITING_CTS“

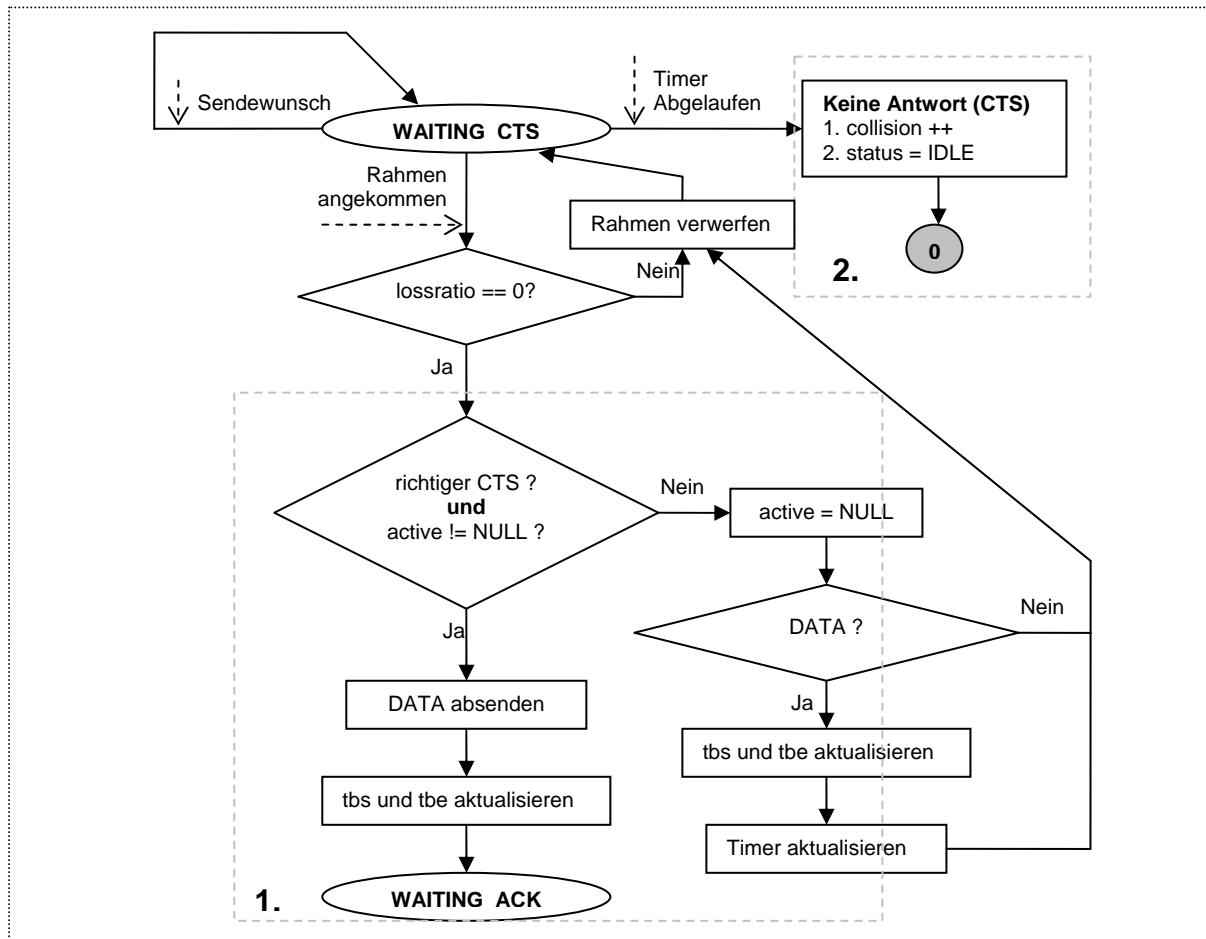


Abbildung 4-21 Zustandsdiagramm – „WAITING_CTS“

Erklärungen zum Zustand „WAITING_CTS“:

1. Wie die Abbildung 4-14 zeigt, wird ein Timer nach dem Absenden von RTS gestartet. Innerhalb dieses Zeitraums ist ein Rahmen außer dem „richtigen“ CTS angekommen, kann festgestellt werden, dass eine Kollision auftritt. Wenn eine Kollision passiert, verweist der Zeiger „active“ auf NULL. Das gibt an, dass eine Kollision vorkommt. Ein „richtiger“ RTS muss Folgendes erfüllen:
 - a. Der RTS ist an diese Station geschickt worden (richtige MAC-Adresse).
 - b. Der Kanal ist zum Zeitpunkt des Empfangs von RTS frei (vergleichen mit „tbe“).

Nach der oben erklärten Vereinfachung, sendet die Station den Datenrahmen sofort ab, wenn der „richtige“ RTS angekommen ist und der Zeiger „active“ nicht auf NULL aufweist, d.h. ist keine Kollision passiert. Danach werden der Timer und die Variablen „tbs“ und „tbe“ aktualisiert. Die Abbildung 4-22 veranschaulicht das Zeitdiagramm. Anschließend geht die Station in den Zustand „WAITING_ACK“ über.

2. Wenn der Timer angelaufen ist, liegt eine erfolglose Sendung vor. Die Variable „collision“ wird erhöht. Danach fängt die Station mit dem erneuten Senden an.

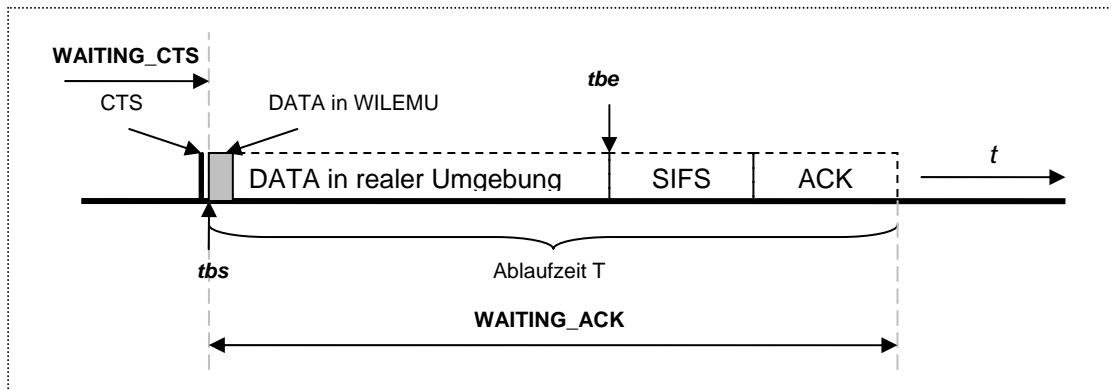


Abbildung 4-22 Zeitdiagramm - Empfang des CTS und Absenden von DATA

4.4.2.6 „WAITING_ACK“

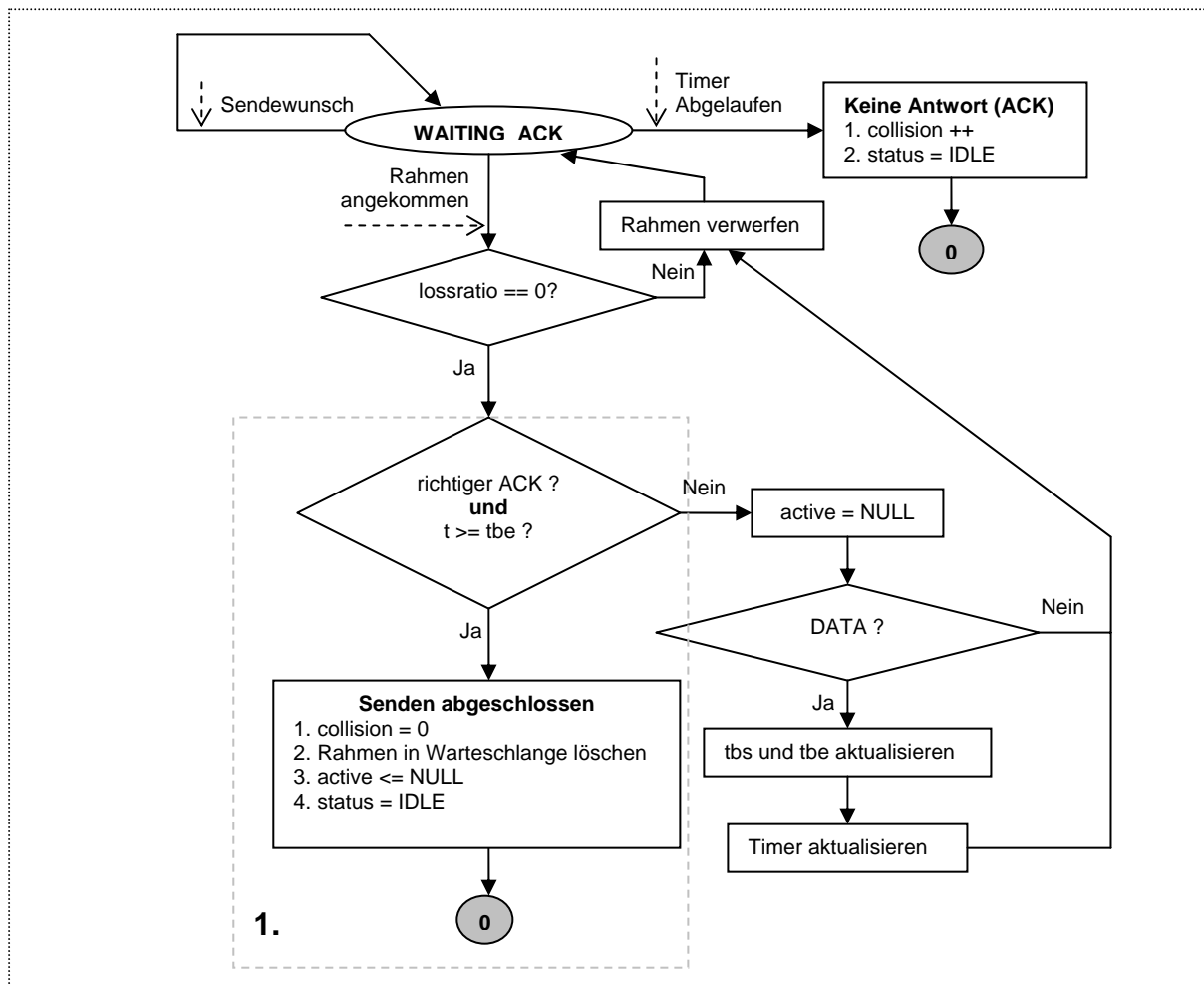


Abbildung 4-23 Zustandsdiagramm – „WAITING_ACK“

Erklärungen zum Zustand „WAITING_ACK“:

Ähnlich wie bei dem Zustand „WAITING_CTS“, ist der Timer abgelaufen, erhöht sich die Variable „collision“. Danach ist eine erneute Sendung erforderlich.

1. Wenn die Station einen richtigen ACK empfangen hat, liegt eine erfolgreiche Übertragung vor. Wie die Abbildung 4-14 zeigt, wird die Variable „collision“ dann zurückgesetzt. Der Rahmen, auf dem „active“ verweist, wird in der Warteschlange gelöscht. Die Station wird danach noch die Warteschlange überprüfen, ob es noch Rahmen in der Warteschlange zu senden gibt (vgl. die Abbildung 4-11).

4.4.2.7 „WAITING_DATA“

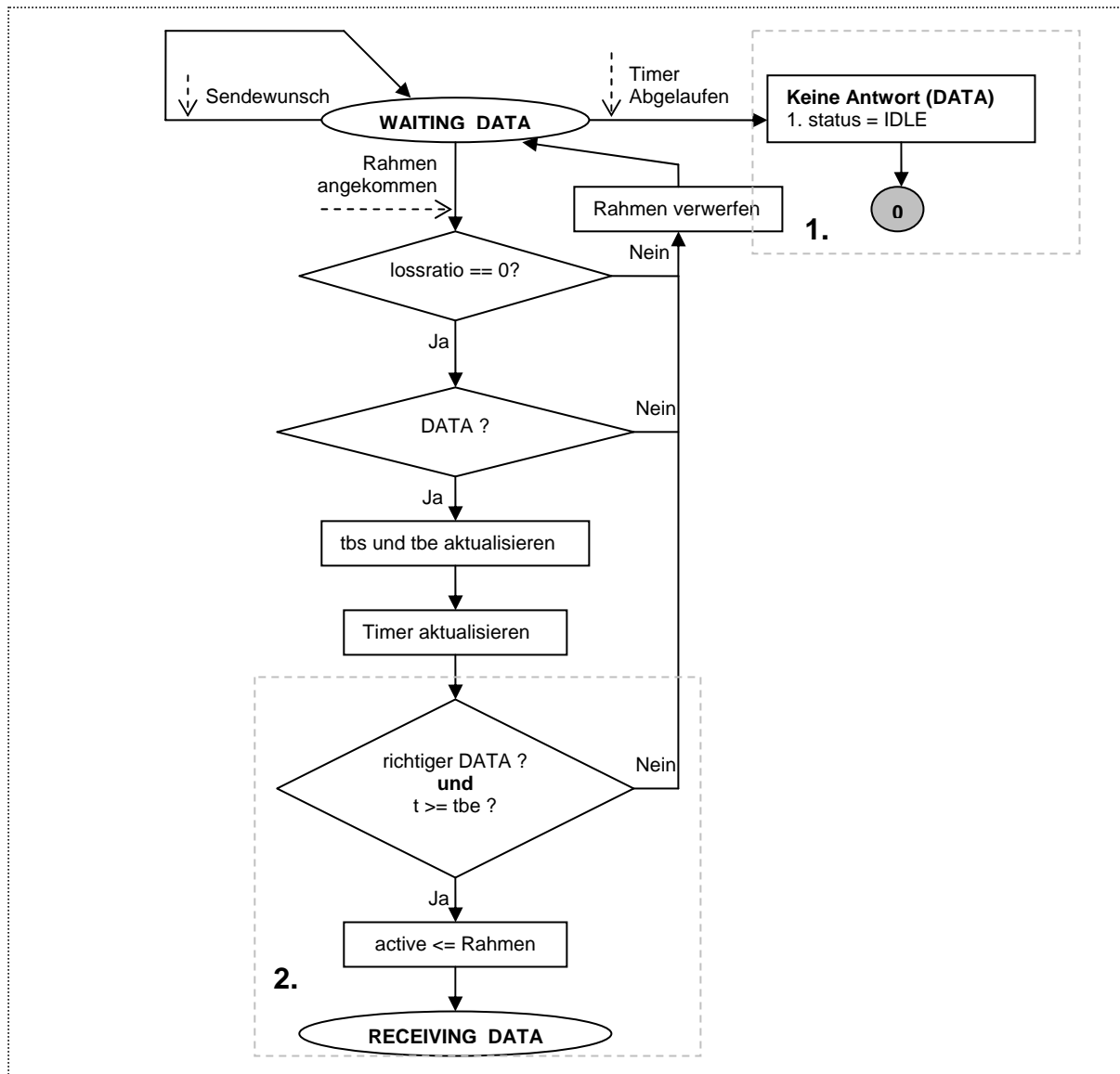


Abbildung 4-24 Zustandsdiagramm – „WAITING_DATA“

Erklärungen zum Zustand „WAITING_DADA“:

1. Ist der Timer bei „WAITING_DATA“ abgelaufen, heißt das, dass der Datenrahmen von dem Kommunikationspartner nach dem Senden von CTS nicht empfangen wurde. Die Station wird dann in den Zustand „IDLE“ wechseln und überprüft die Warteschlange.
2. Wenn die Station einen richtigen DATA-Rahmen empfangen hat, antwort die Station mit einem ACK, aber nicht sofort. Da die Übertragungszeit in Emulationsumgebung deutlich kürzer als in der Realität ist, und ein Datenrahmen eine Länge von bis zu 1500 Bytes haben kann, wird eine künstliche Verzögerung hier eingefügt: die Station geht in den Zustand „RECEIVING_DATA“ (vgl. die Abbildung 4-11). Die Variable „active“ verweist nun auf den gerade empfangenden Rahmen.

4.4.2.8 „RECEIVING_DATA“

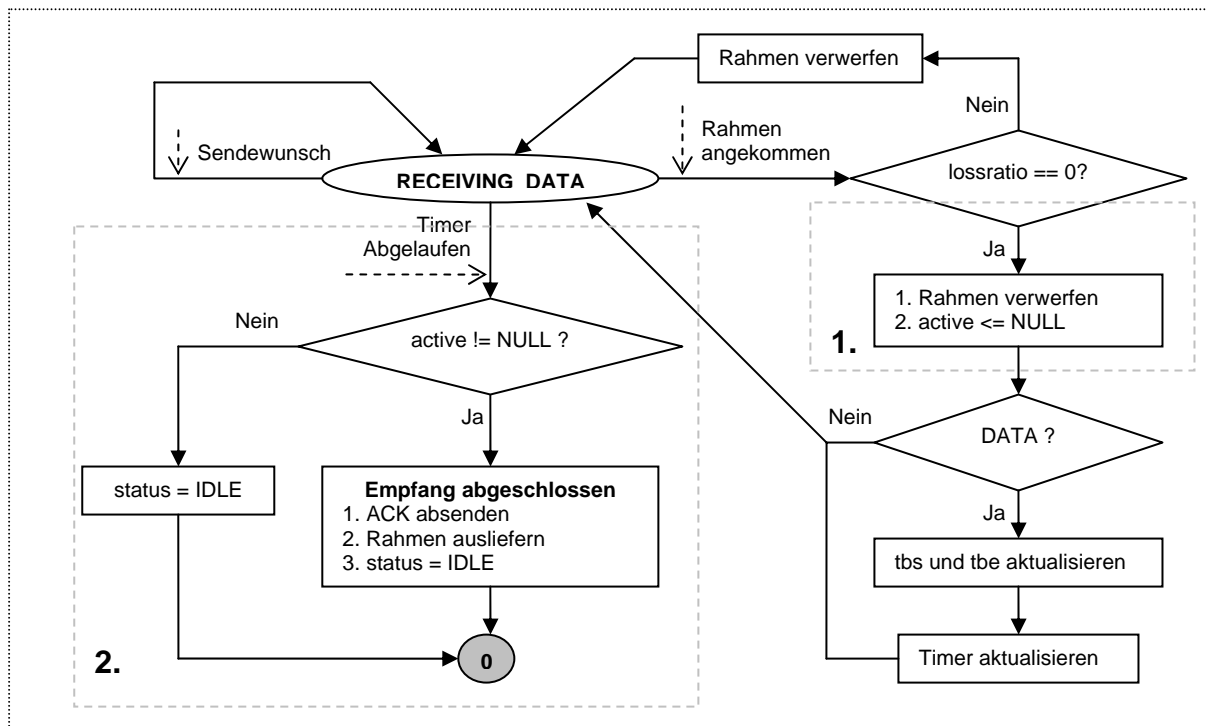


Abbildung 4-25 Zustandsdiagramm – „RECEIVING_DATA“

Erklärungen zum Zustand „RECEIVING_DATA“:

Nach dem Empfang des Datenrahmens in Emulationsumgebung muss noch getestet werden, ob der Rahmen während der realen Übertragungszeitdauer durch eine andere Übertragung gestört worden ist.

1. Ist ein Rahmen angekommen, heißt es, dass der Datenrahmen („active“) durch diesen gerade empfangenden Rahmen kollidiert worden ist. Die beiden Rahmen werden weggeworfen, und „active“ wird NULL. Der Empfangsvorgang wird nicht abgebrochen, sondern läuft weiter bis der Timer abgelaufen ist. Die Ablaufzeit ist die maximale reale Übertragungszeit von den empfangenen Rahmen.
2. Ist der Timer bei „RECEIVING_DATA“ abgelaufen, überprüft die Station, ob „active“ auf NULL verweist. Wenn ja, liegt ein erfolgloser Datenempfang vor. Die Station wechselt seinen Zustand auf „IDLE“ und überprüft die Warteschlange. Wenn nein, heißt es, dass der Datenempfang erfolgreich abgeschlossen ist. Die Station antwortet sofort mit einem ACK-Rahmen und liefert den Datenrahmen an höheren Schicht. Anschließend geht die Station in den Zustand „IDLE“ und überprüft die Warteschlange.

4.4.2.9 „BROADCASTING“

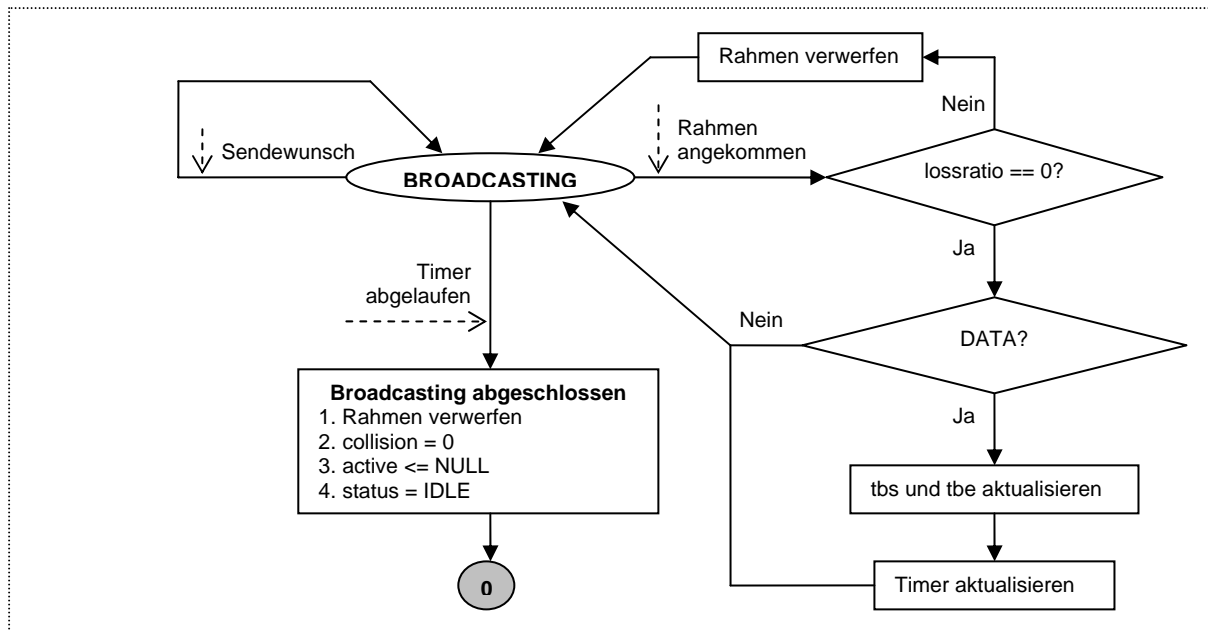


Abbildung 4-26 Zustandsdiagramm – „BROADCASTING“

Erklärungen zum Zustand „BROADCASTING“:

Nach der Eigenschaft des Wireless LAN kann eine Station während des Sendens keine Kollision detektieren. Im Zustand „BROADCASTING“, wenn ein anderer Rahmen angekommen ist, wird dieser weggeworfen. Für einen Datenrahmen, dessen reale Übertragungsdauer länger als die Ablaufzeit des Timers ist, wird der Timer aktualisiert. Wenn der Timer abgelaufen ist, ist das „Broadcasting“ abgeschlossen. Die Station geht in den Zustand „IDLE“. Anschließend wird die Warteschlange überprüft.

4.4.2.10 „RECEIVING_BC“

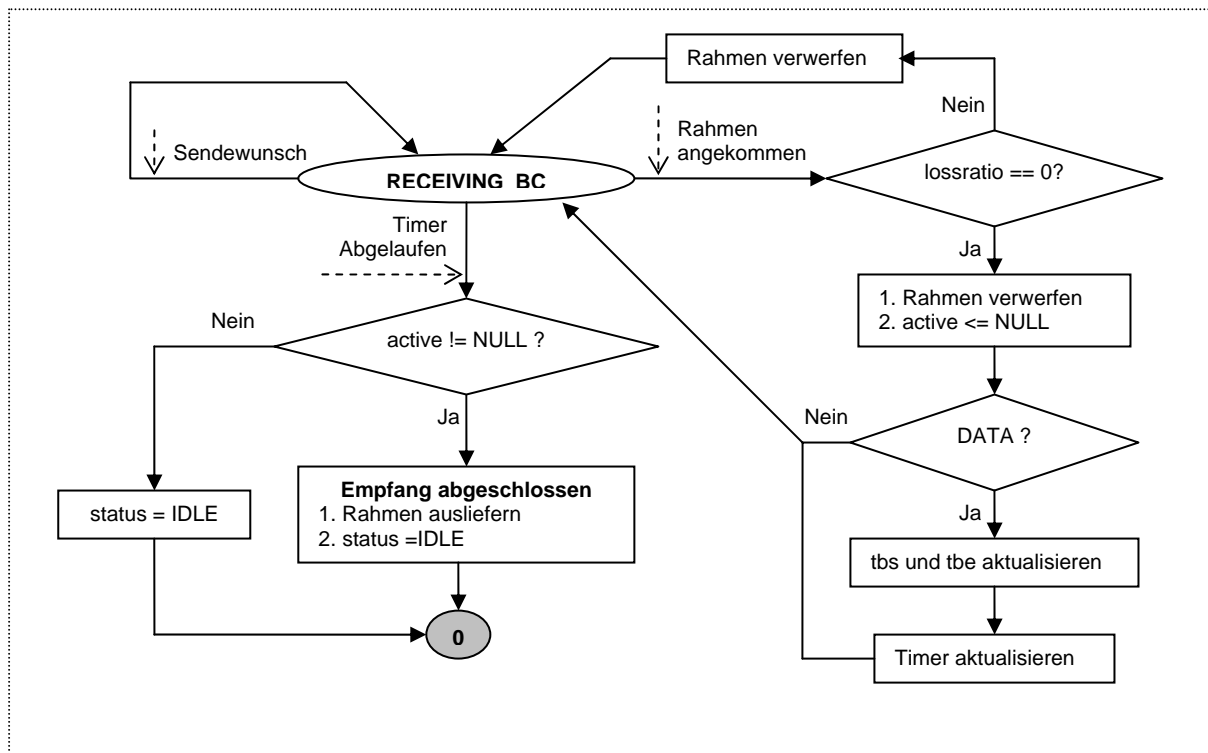


Abbildung 4-27 Zustandsdiagramm – „RECEIVING_BC“

Erklärungen zum Zustand „RECEIVING_BC“:

Ähnlich wie beim Zustand „RECEIVING_DATA“ befindet sich eine Station im Zustand „RECEIVING_BC“, wenn sie einen Rahmen mit „Broadcast“-Adresse empfangen hat. Der Unterschied ist, nach dem Ablauf des Timers liefert die Station diesen Rahmen an die höhere Schicht, ohne einen ACK-Rahmen zu senden (vgl. Abbildung 4-25).

Kapitel 5 Implementierung

5.1 Architektur

Die entwickelte Emulationssoftware ist in der Programmiersprache C geschrieben und funktioniert als Modul im Linux-Kernel. Die eingesetzte Linux-Kernel ist eine leicht veränderte Version 2.4, die den APIC-Timer unterstützt. Zur Übersetzung der Programme ist der Compiler „gcc“, Version 3 erforderlich. Entsprechend der Hardwareumgebung in NET sind die Programme für die Intel-Plattform übersetzt.

| | |
|--------------------|--|
| Programmiersprache | Standard C |
| Kernel | Linux-Kernel mit APIC-Timer Patch, Version >=2.4 |
| Plattform | Intel Pentium (i386) |
| Übersetzer | Gnu gcc, Version >= 3.3 |

Tabelle 5-1 Entwicklungsumgebung

Die WILEMU Software besteht aus folgenden Dateien:

| Dateiname | Beschreibung |
|---------------|---|
| wilemu.h | Formate der Fehlermeldungen |
| wilemu.c | Funktionen zur Steuerung des Moduls und des Netzwerkgeräts |
| if_wilemu.h | Definitionen von lokalen Variablen |
| wilemu_func.h | Schnittstelle für den Zugriff auf die Funktionen in wilemu_func.c |
| wilemu_func.c | Kernfunktionen, die den konzipierten Algorithmus realisieren |

Tabelle 5-2 Bestandteile von WILEMU

Die Software wird als in einen Linux-Kernel-Modul übersetzt („wemodule.o“). WILEMU funktioniert im Linux-Kernel ähnlich wie ein Netzwerktreiber. Wie schon in Abschnitt 2.7.2 vorgestellt, arbeitet WILEMU grundsätzlich zwischen der zweiten und dritten Schicht des Kommunikationsstapels in Linux. Abbildung 5-1 veranschaulicht die Kommunikationsstapel für die Emulation.

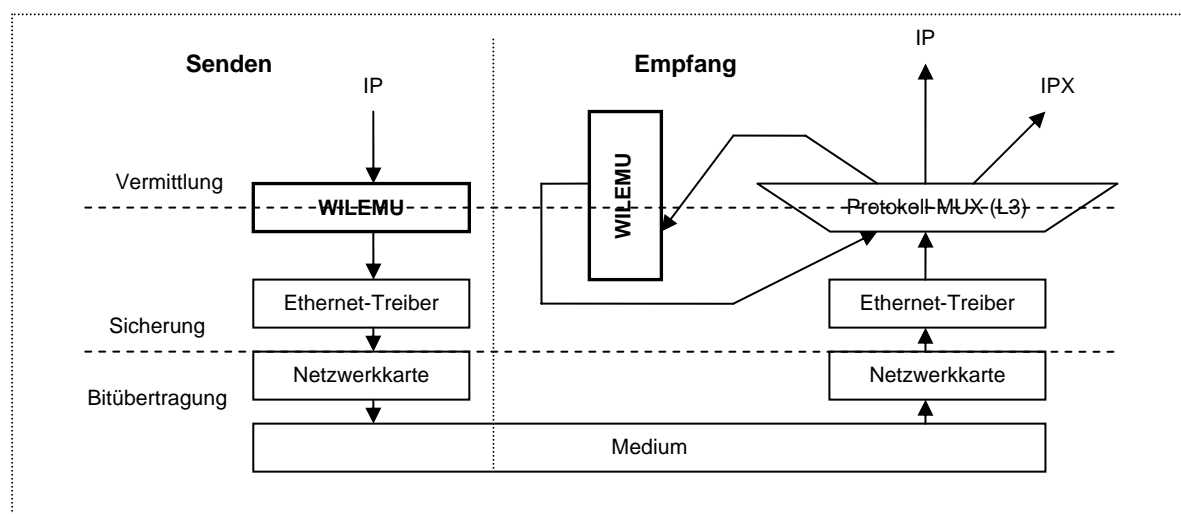


Abbildung 5-1 WILEMU im Kommunikationsstapel

5.2 Funktionsweise

5.2.1 Senden und Empfang

Für die obere Schicht ist WILEMU durch eine IP-Adresse identifiziert. Ebenfalls ist WILEMU mit einem realen Ethernet-Gerät gebunden. D.h. WILEMU besitzt die gleiche MAC-Adresse wie die darunter liegende Netzwerkkarte.

Wie in Abbildung 5-1 veranschaulicht, wird eine Nachricht mit einer bestimmten IP-Adresse an WILEMU dadurch geschickt, dass die Funktion „dev_hard_start_xmit()“ an der Schnittstelle für das Senden aufgerufen wird. Die Funktion „dev_hard_start_xmit()“ ist eine Standard-Funktion jedes Netzwerktreibers. In WILEMU leitet die „dev_hard_start_xmit()“ an die internen Funktionen weiter, die den Emulations-Algorithmus implementieren. Der genaue Programmablauf wird in Abschnitt 5.3 erklärt. Nach der Bearbeitung des Rahmens in dem WILEMU-Modul ruft WILEMU die Funktion „dev_queue_xmit()“ auf, damit der bearbeitete Rahmen an die Schnittstelle-Funktion des Treibers der realen Netzwerkkarte weitergeleitet wird. Damit wird ein Rahmen mittels der Netzwerkkarte in NET abgeschickt. Abbildung 5-2 veranschaulicht die Schnittstelle-Funktionen der WILEMU.

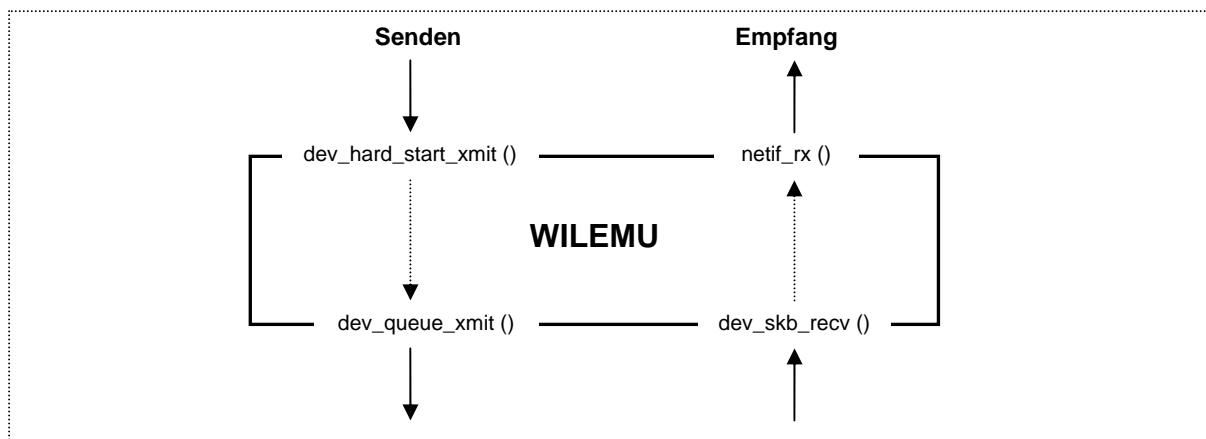


Abbildung 5-2 Schnittstelle-Funktionen der WILEMU

Da der Standard-Treiber der Netzwerkkarte in der Empfangsrichtung nicht geändert werden soll, liefert der Treiber der Netzwerkkarte in NET den empfangenen Rahmen an die Protokoll-Multiplexer auf Schicht drei. Dieser Protokoll-Multiplexer entscheidet dann, wem dieser Rahmen weiter geliefert werden soll. Dafür muss ein spezieller Protokolltyp für WILEMU in Protokoll-Multiplexer eingetragen werden:

```
static struct packet_type wilemu_packet_type =
{
    type: __constant_htons(ETH_P_WILEMU),
    dev: NULL, /* receive on all devices */
    func: wilemu_dev_skb_recv,
    data: (void *)(-1),
    next: NULL
};
```

Der Protokolltyp allen Rahmen, die über WILEMU verschickt werden, müssen nach diesem Typ (ETH_P_WILEMU) gekennzeichnet werden (in Ethernet-Rahmenkopf einzutragen). Wenn der Protokoll-Multiplexer einen Rahmen mit dem Protokolltyp für WILEMU bekommt, ruft er dann die

Funktion „dev_skb_recv()“ auf, die in WILEMU definiert ist. Damit wird ein Rahmen an WILEMU ausgeliefert. Nach der Bearbeitung dieses Rahmens in WILEMU, z.B. die Überprüfung der Fehlerrateliste, wenn der Rahmen ausgeliefert werden soll, ruft WILEMU die Funktion „netif_rx()“ auf, damit der Rahmen wieder an den Protokoll-Multiplexer gesendet wird, wie Abbildung 5-2 zeigt. Danach wird dieser Rahmen ohne MAC_Kopf durch die Entscheidung des Protokoll-Multiplexers weiter nach oben geliefert.

5.2.2 Modulaktivierung und -deaktivierung

Zur Aktivierung und Deaktivierung des Moduls zur Laufzeit stehen die Standard-Programme „insmod“ und „rmmod“ zur Verfügung. Dafür sind „root“-Rechte erforderlich. Da die WILEMU APIC-Timer verwendet, muss das APIC-Timer-Modul (uka_apic_timer.o) zuerst aktiviert werden.

```
insmod uka_apic_timer.o      /* Modulaktivierung (APIC-Timer) */
insmod wemodule.o          /* Modulaktivierung */

rmmod wemodule              /* Moduldeaktivierung */
rmmod uka_apic_timer        /* Moduldeaktivierung (APIC-Timer) */
```

Mit der Hilfe von „ifconfig“ kann eine IP-Adresse an WILEMU zugewiesen werden. Das Kommando „up“ schaltet das Gerät an:

```
ifconfig wilemu0 inet up 10.1.2.25    /* Beispiel */
```

5.2.3 Steuerung von WILEMU und Änderung von Parametern

In Linux dient der Systemruf „ioctl“ zur allgemeinen Steuerung von Ein- und Ausgabegeräten. Da WILEMU als ein logisches Netzwerkgerät funktioniert, kann sie auch mittels „ioctl“ gesteuert werden. Hierzu wird das „ioctl“-Kommando SIOCDEVPRIVATE für Netz-Sockets in Verbindung mit dem Selektor „ifr_data“ der Datenstruktur „struct ifreq“ (aus <linux/if.h>) verwendet [maie02]. Über diesen Selektor wird die eigene Konfigurations-Datenstruktur „struct wilemu_conf“ übergeben. Die Struktur ist wie folgt definiert:

```
struct wilemu_conf {
    enum wilemu_conf_cmds cmd; /* structure holding wilemu device
                                configuration data for ioctl */
    union {
        char      name[IFNAMSIZ]; /* real dev's name */
        __u32     speed;          /* speed [bit/s] */
        __u32     MTU;           /* MTU [bytes] */
        __u32     slottime;      /* slot time [us] */
        __u32     DIFS;          /* DIFS [us] */
        __u32     SIFS;          /* SIFS [us] */
        struct nodelink link;
    } param;
};
```

Der Datentyp des Parameters „link“ ist „nodelink“. Ein „link“ entspricht einem Eintrag in der Fehlerrateliste und besteht aus zwei Teilen: MAC-Adresse und Fehlerrate. Die Struktur „nodelink“ ist wie folgt definiert:

```
struct nodelink {
    unsigned char nl_hwaddr[ETH_ALEN];    /* MAC-address */
    unsigned loss;                        /* loss ratio */
};
```

Die in WILEMU definierte „cmd“ sind in Tabelle 5-3 aufgelistet.

| cmd | Wert | Beschreibung |
|---------------------|--------|---|
| WILEMU_SET_DEV | 0x0000 | Bindet WILEMU an reales Netzwerkgerät |
| WILEMU_DEL_DEV | 0x0001 | Trennt die Bindung zwischen WILEMU und realem Netzwerkgerät |
| WILEMU_GET_DEV | 0x0002 | Liefert den Namen des realen Netzwerkgeräts |
| WILEMU_SET_SPEED | 0x0003 | Setzt die Übertragungsrate [Bit/s] |
| WILEMU_GET_SPEED | 0x0004 | Liefert die Übertragungsrate |
| WILEMU_SET_MTU | 0x0005 | Setzt die MTU (maximale Länge der Netzdaten) [Bytes] |
| WILEMU_GET_MTU | 0x0006 | Liefert den Wert der MTU |
| WILEMU_SET_SLOTTIME | 0x0007 | Setzt den Wert des Zeitschlitzes [Mikrosekunden] |
| WILEMU_GET_SLOTTIME | 0x0008 | Liefert den Wert des Zeitschlitzes |
| WILEMU_SET_DIFS | 0x0009 | Setzt den Wert des DIFS [Mikrosekunden] |
| WILEMU_GET_DIFS | 0x000A | Liefert den Wert des DIFS |
| WILEMU_SET_SIFS | 0x000B | Setzt den Wert des SIFS [Mikrosekunden] |
| WILEMU_GET_SIFS | 0x000C | Liefert den Wert des SIFS |
| WILEMU_ADD_NODELINK | 0x000D | Fügt einen Eintrag in die Fehlerrateliste ein |
| WILEMU_DEL_NODELINK | 0x000E | Löscht einen Eintrag aus der Fehlerrateliste |
| WILEMU_GET_NODELINK | 0x000F | Liefert einen Eintrag der Fehlerrateliste |

Tabelle 5-3 Definitionen von „cmd“

Zur manuellen Einstellung von WILEMU in der Kommandozeile steht das Programm „weconfig“ zur Verfügung:

```
./weconfig <wilemu device name>                /* zum Beispiel wilemu0 */
    -d <real network device name>                /* attach/bind */
    -u                                           /* detach/unbind */
    -b <network bandwidth in bps>                /* set speed */
    -m <MTU in Bytes>                            /* set MTU */
    -c <slot time in microseconds>              /* set Slot time */
    -w <Wireless Lan DIFS in microseconds>      /* set DIFS */
    -o <Wireless Lan SIFS in microseconds>      /* set SIFS */
    -a <MAC-address>,<loss ratio, 0 or 1>        /* add entry */
    -r <MAC-address>                             /* remove entry */
    -l                                           /* list settings */
```

5.3 Algorithmus

Entsprechend dem in Kapitel 4 diskutierten Entwurf, ist der Algorithmus der Emulationssoftware WILEMU in Abbildung 5-3 zu sehen.

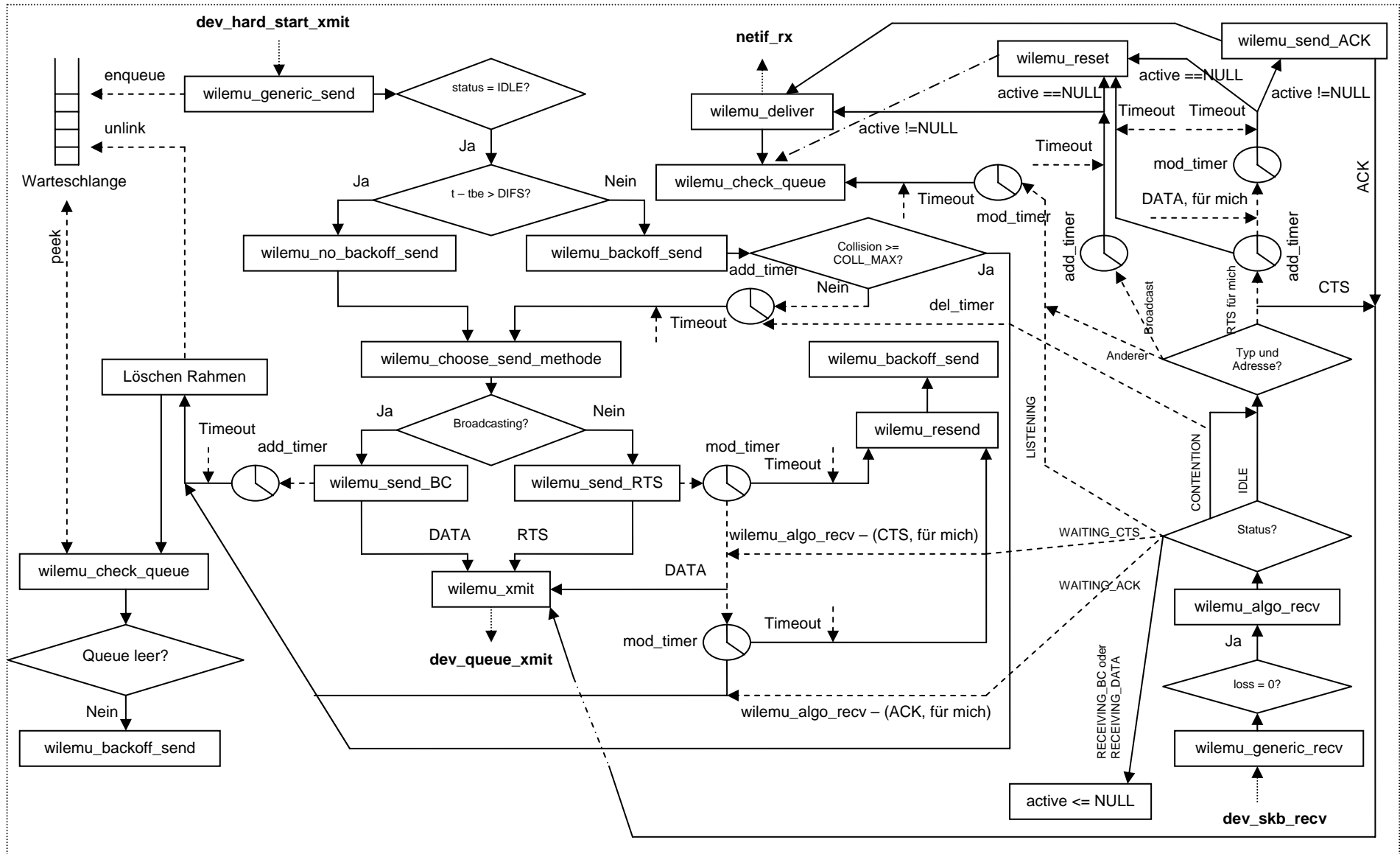


Abbildung 5-3 Vereinfachter Algorithmus von WILEMU

Kapitel 6 Testmessung

6.1 Test-Werkzeug

Zum Durchsatz-Testen der Implementierung wird das Programm „netio“ verwendet:

```
./netio
```

```
NETIO - Network Throughput Benchmark, Version 1.13  
(C) 1997-2001 Kai Uwe Rommel
```

```
Usage: netio [-s] [-p <port>] [<server>]
```

```
-s          run server side of benchmark (otherwise run client)  
-p <port>  use this port instead of the default (18767)  
<server>   if the client side of the benchmark is running,  
           a server host name is required
```

Als Server wird „./netio -s“ bei einer Station aufgerufen. Mit der Option „-p“ können mehrere Server auf einer Station zum Durchsatz-Testen installiert werden. Damit werden mehrfache Verbindungen realisiert. Auf der Client-Seite, wird der Test durch das Kommando „./netio <IP-Nummer des Servers: Port-Nummer>“ gestartet. „netio“ misst den Durchsatz jeweils für Paketgrößen von 1KB, 2KB, 4KB, 8KB, 16KB bis zu 32KBytes.

Um die „Round-Trip-Time“ und damit den Overhead zu testen, wird das Standard-Programm „Ping“ benutzt.

6.2 Test-Szenarien und Messergebnisse

Für die in IEEE 802.11b spezifizierten Brutto-Übertragungsraten (1Mbit/s, 2Mbit/s, 5.5 Mbit/s, 11Mbit/s) werden jeweils eine (Test 1) und zwei (Test 2) gleichzeitige Übertragungen getestet. In Test 3 werden die „Round-Trip“ Zeiten ebenfalls für die unterschiedlichen Übertragungsraten getestet.

6.2.1 Test 1

Im diesem Durchsatztest sind 2 Stationen beteiligt, wie Abbildung 6-1 zeigt.

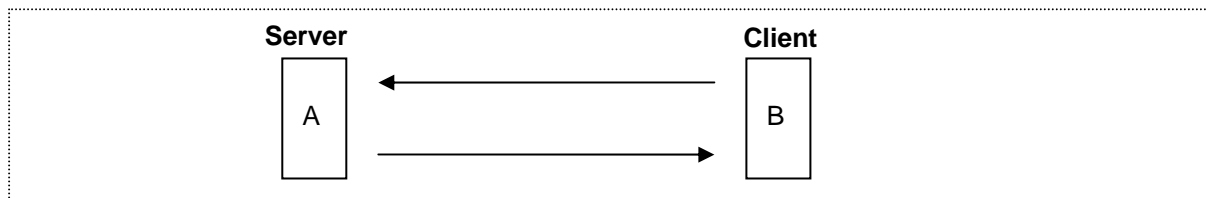


Abbildung 6-1 Test 1

Station A dient als Server, Station B schickt durch „netio“ Pakete an A. Die Messergebnisse sind in Tabelle 6-1 aufgelistet. Abbildung 6-2 ist eine graphische Veranschaulichung jeweils mit dem Durchschnittswert über die Paketgrößen.

| Übertragungsrate \ Paketgröße | Paketgröße | | | | | |
|-------------------------------|------------|----------|----------|----------|-----------|-----------|
| | 1 KBytes | 2 KBytes | 4 KBytes | 8 KBytes | 16 KBytes | 32 KBytes |
| 11 (MBit/s) | 5.98 | 6.01 | 6.08 | 6.22 | 6.05 | 6.07 |
| 5.5 (MBit/s) | 4.64 | 4.79 | 4.59 | 4.62 | 4.76 | 4.40 |
| 2 (MBit/s) | 1.93 | 1.91 | 1.76 | 1.90 | 1.88 | 1.92 |
| 1 (MBit/s) | 1.30 | 1.18 | 1.26 | 1.22 | 1.22 | 1.26 |

Tabelle 6-1 Messergebnis bei einem Sender/Empfänger

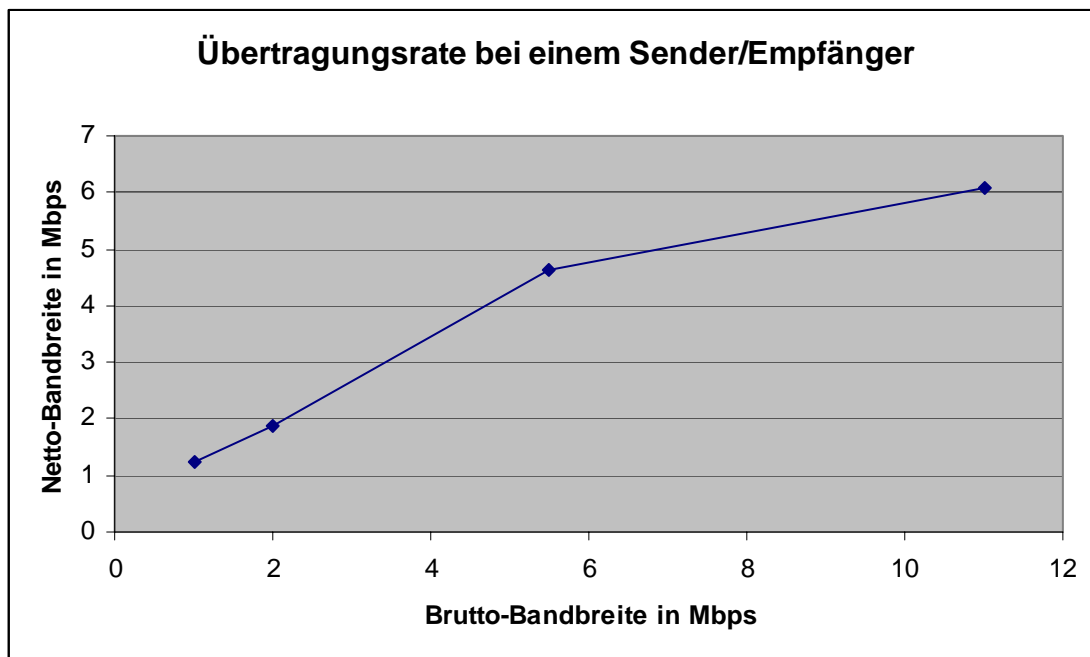


Abbildung 6-2 Durchsatz bei einem Sender/Empfänger

6.2.2 Test 2

In diesem Test sind 3 Stationen beteiligt, wie Abbildung 6.3 zeigt.

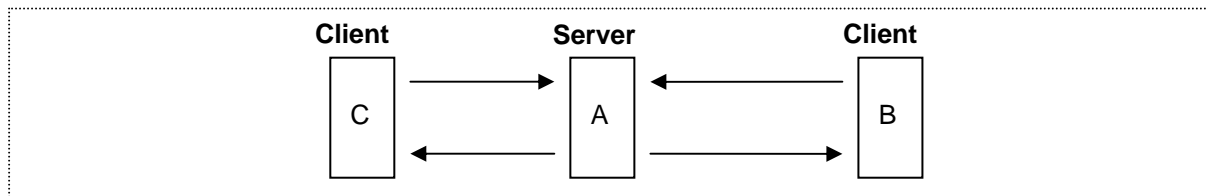


Abbildung 6-3 Test 2

Station A dient als Server, Clients B und C bauen Verbindungen mit A auf. Die Testläufe werden gleichzeitig auf den Clients B und C gestartet. Die Messergebnisse sind in Tabelle 6-2 aufgelistet. Abbildung 6-4 ist eine graphische Veranschaulichung, jeweils mit dem Durchschnittswert über die Paketgröße.

| Übertragungsrate Paketgröße | | Übertragungsrate | | | |
|--------------------------------|---|------------------|--------------|------------|------------|
| | | 11 (MBit/s) | 5.5 (MBit/s) | 2 (MBit/s) | 1 (MBit/s) |
| 1 KBytes | B | 2.64 | 2.32 | 0.98 | 0.60 |
| | C | 2.33 | 1.60 | 1.26 | 0.77 |
| 2 KBytes | B | 2.25 | 1.86 | 1.02 | 0.66 |
| | C | 2.12 | 1.90 | 1.02 | 0.59 |
| 4 KBytes | B | 2.26 | 1.89 | 1.02 | 0.72 |
| | C | 2.40 | 1.71 | 1.07 | 0.61 |
| 8 KBytes | B | 1.92 | 1.94 | 1.10 | 0.62 |
| | C | 2.43 | 1.66 | 0.86 | 0.65 |
| 16 KBytes | B | 1.97 | 1.94 | 0.94 | 0.62 |
| | C | 2.41 | 1.75 | 1.19 | 0.67 |
| 32 KBytes | B | 2.13 | 1.90 | 0.94 | 0.53 |
| | C | 2.80 | 2.05 | 1.15 | 0.82 |

Tabelle 6-2 Messergebnis bei zwei Sendern und einem Empfänger

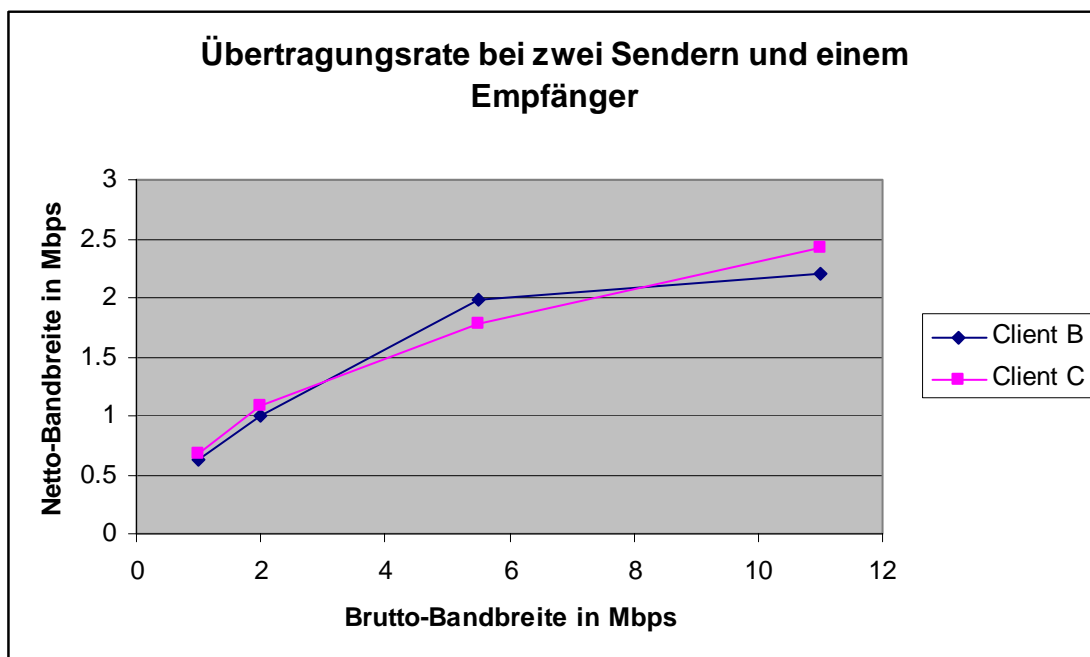


Abbildung 6-4 Durchsatz bei zwei Sendern und einem Empfänger

6.2.3 Test 3

Im diesem „Round-Trip-Time“-Test sind 2 Stationen beteiligt, wie Abbildung 6-5 zeigt.

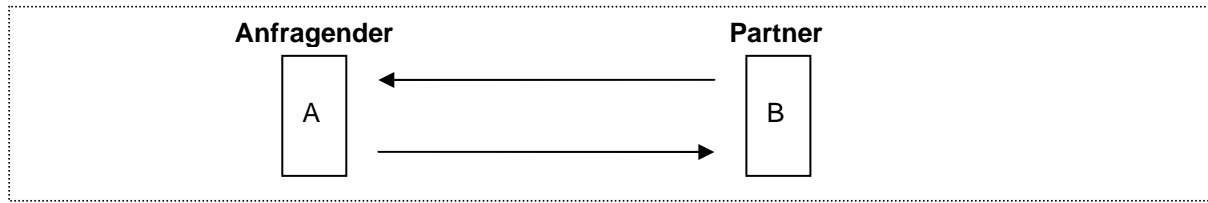


Abbildung 6-5 Test 3

Station A sendet „Ping“-Anfragen an die Station B. Die Messergebnisse mit den Durchschnittswerten des „Round-Trip“-Timers von 10 Anfragen sind in Tabelle 6-3 aufgelistet. Abbildung 6-6 ist eine graphische Veranschaulichung.

| Übertragungsrate | 11 MBit/s | 5.5 MBit/s | 2 MBit/s | 1 MBit/s |
|------------------|-----------|------------|----------|----------|
| Dauer | 0.952 ms | 1.061 ms | 1.772 ms | 2.238 ms |

Tabelle 6-3 Messergebnis „Ping“

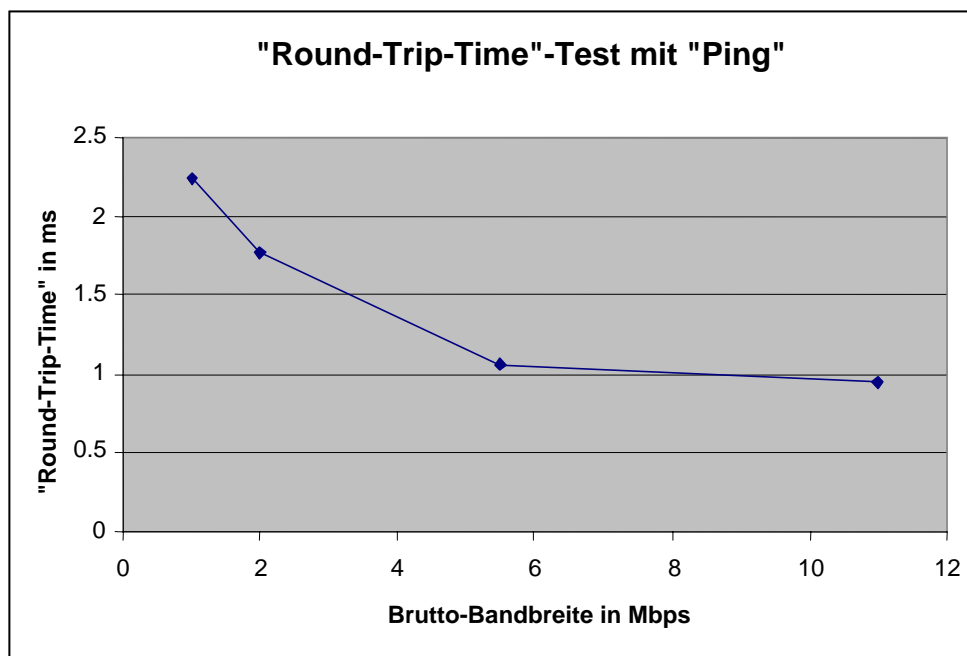


Abbildung 6-6 „Round-Trip-Time“ mit „Ping“

6.3 Diskussion der Testergebnisse

Die Testergebnisse zeigen, dass die emulierte Bandbreite von WILEMU realistische Werte ergibt. Eventuell ist die Netto-Bandbreite durch die Vereinfachungen in der Emulation etwas zu groß.

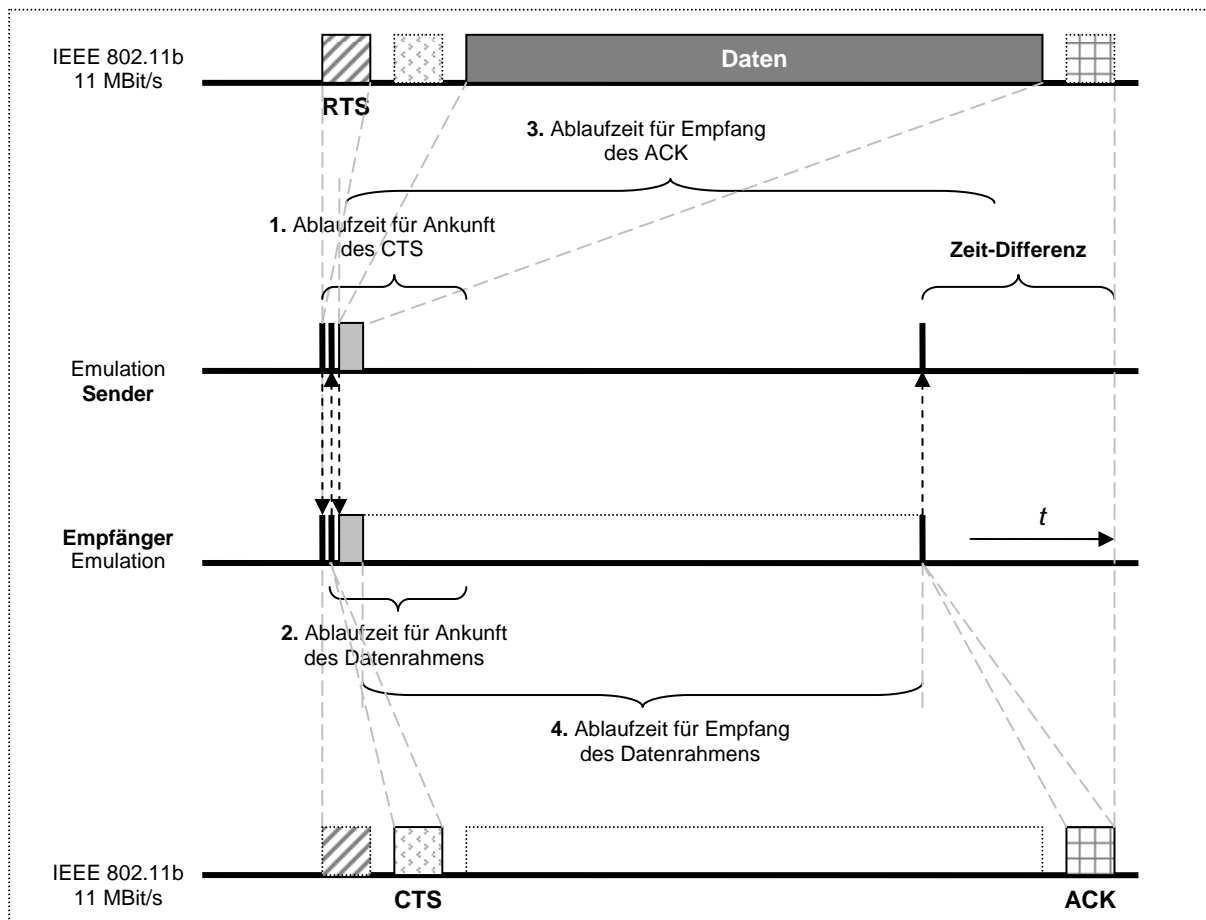


Abbildung 6-7 Zeit-Differenz der Datenübertragung in WILEMU und im realen Wireless LAN

Wie die Abbildung 6-7 zeigt, ist die Zeitdauer für die Übertragung eines Rahmens in der Emulationsumgebung (IEEE 802.11b 11 MBit/s) kürzer als in der realen Umgebung. Um das Verhältnis in Realität genau nachzubilden, muss ein Timer nach dem Rahmensenden oder -empfang gestartet werden. Die Ablaufdauer des Timer beträgt etwa die reale Übertragungszeit dieses Rahmens. Damit werden die Übertragungen in der Emulationsumgebung synchronisiert mit denen in der Realwelt. Wie schon in Abschnitt 4.3.6 erklärt (SIFS), kann die kurze Zeitdauer allerdings nicht ganz exakt behandelt werden, obwohl der APIC-Timer eingesetzt worden ist.

Mit einer niedrigen Übertragungsrate, wie z.B. 1 MBit/s, wird diese Zeit-Differenz (siehe Abbildung 6-7) sogar noch größer. Einerseits ist die Übertragungszeit wegen der kleinen Übertragungsrate länger. Andererseits ist die SIFS größer, die bei der Emulation ignoriert wurde (mit FHSS, vgl. Tabelle 4-10).

Die Ergebnisse der „Ping“-Messung zeigen, dass der von WILEMU eingeführte Overhead nicht größer ist als der Overhead einer realen MAC-Implementierung.

Kapitel 7 Verwandte Arbeiten

7.1 Zentralisierte Methoden

7.1.1 NS – E

NS als ein ereignisgesteuertes System ist einer der bekanntesten NS implementiert die üblichen Netzwerksimulatoren. Internetprotokolle z.B. TCP, UDP usw. Darüber hinaus enthält es einige Spezielle Protokolle wie z.B. zuverlässiges Multicast, TCP selektive acknowledge usw [fall99].

Konzipiert und realisiert durch Universität California, Berkeley beschreibt NS-E die Erweiterung des zentralisierten Netzsimulators NS zu einem Emulator. Hierzu wird der diskrete Ereignisscheduler des Simulators zu einem Echtzeit-Scheduler umgewandelt. Die Verbindung zum realen Netz wird über Schnittstellen hergestellt [fall99].

Wie es bei allen diskreten ereignisgesteuerten Simulatoren der Fall ist, funktioniert der generische NS-Scheduler in virtueller Zeit in einer ereignisgesteuerten Weise, d.h. aktuelle Zeit wird gleich vor der Absendungszeit nächsten hängenden Ereignisses bestimmt. Um die Echtzeit anzunähern, sendet der Basisscheduler dieses Ereignis in der Emulation nicht gleich vor dem Zeitpunkt des nächsten hängenden Ereignisses, sondern zur geeigneten Echtzeit. Das heißt: der Scheduler fängt an, in dem er irgendein altes eingetroffenes Ereignis ausführt. Diese Exekution dauert so lange, bis entweder ein Zukunftsereignis bevorsteht oder keine vergangenen Ereignisse mehr existieren. Wenn ein Zukunftsereignis bevorsteht, wartet dieser Scheduler entweder bis zur Deadline dieses Ereignisses oder zum Eintreffen eines I/O-Ereignisses. Wenn keine Zukunftsereignisse vorliegen, führt dieser Scheduler ein I/O-Ereignis aus und startet neu. Beim Routing und Adressierung werden die IP-Adressen in der Emulation beim Eintreffen und Abfahren der IP-Datagramme abgebildet, so dass sie durch die interne Netzwerktopologie des Simulators umgeleitet werden. Der Zugriff zu den Paketdaten des Netzwerks erfolgt auf Netzwerkobjekte, die auf Live-Sources oder Tracedateien, die vorher gesammelten Netzwerkpakete beinhalten, hinweisen. Der Zugriff zu den Rahmen wird durch die Bibliothek zum Paketabfangen einschließlich Berkeley-Paketfilter bereitgestellt. Damit die Unverträglichkeiten der Adressen zwischen dem Simulator und der Realwelt überwunden werden, werden spezifische Objekte (tap Agents) verwendet, um den Live-Netzwerkverkehr innerhalb Simulatorpaketen zu tunneln. Kombiniert mit einem verbundenen Netzwerkobjekt, kann ein tap Agent den Live-Netzwerkverkehr in verschiedenen Protokollschichten absenden und empfangen [fall99].

Mit dieser Implementierung lässt sich die Emulation des Netzwerksimulators durch den programmierbaren Simulorkern praktisch beliebige Manipulationen am gesamten Netzverkehr vornehmen [maie02].

7.1.2 JEmu

Entwickelt durch Trinity College, Irland, ist Jemu, ein drahtloser Netzwerkemulator eine allein stehende Anwendung, die Verbindungen aus den Jemu-Klienten akzeptiert. Solche Klienten sind normalerweise die niedrigste Schicht in einem NTRG-Stack. Um die Verbindungen zwischen mehreren unabhängigen Stacks und einem Emulator zu ermöglichen, wird die Emulationsengine mit einem Klient/Server-Modell konzipiert: jeder Stack hat eine Radio-Emulationsschicht, die die vorhandene Schicht zur Verbindung zum Radiohardware ersetzt und des Weiteren als Klient zur Verbindung zum Emulatorserver dient. Wenn der Emulator und Klientstack auf derselben Maschine laufen, ist es eine lokale Verbindung. Andernfalls erfolgt ihre Verbindung durch ein TCP/IP-Netzwerk [flto01].

Um ein Paket zur Jemu-Schicht abzuschicken, wird eine Nachricht zur Jemu-Radioschicht geliefert. Diese Schicht stellt die Nachricht mit ID des Quellknotens heran, die anschließend zum Emulationsengine hingeschickt wird. Der Emulator empfängt diese Nachricht und fügt einen Zeitstempel ein. Darüber hinaus schlägt er die Knoten-ID in einer Tabelle nach und fügt die X-, Y-Position und die aktuelle Übertragungsleistung des Knotens zur Nachricht hin. Eine gekennzeichnete Nachricht nennt sich ein Fragment [flto01].

Das Fragment wird zu einem Fragmentpool gefügt. Ein anderer Thread, der geringfügig später als aktuelle Zeit t_e arbeitet, fragt diesen Pool innerhalb eines festgelegten Zeitraums t_e bis t_e+q nach Fragmenten ab, wobei t_e der Anfangszeitpunkt des Zeitraums und q die zu berücksichtigende Zeitraumlänge ist. Der Emulator iteriert die Liste der empfangenen Knoten und prüft, ob sie irgendein im Zeitraum von t_e bis t_e+q gefundenes Fragment empfangen können. Wenn ein Knoten in der Lage ist, gleichzeitig mehr als ein Fragment zu empfangen, ist eine Kollision aufgetreten. Abhängig davon, wie der Benutzer den Emulator eingestellt hat, können die Fragmente entweder zusammengeführt, abgesetzt oder sogar zur Jemu-Radioschicht sequentiell zurückgeliefert werden, als ob die Kollision nicht passierte. Zum Schluß beendet der Emulator alle abgearbeiteten Fragmente bzw. die Fragmente mit Zeitstempeln früher als t_e+q im Pool, während t_e um q erhöht wird [flto01].

Dieses Design lässt Paketkollisionen für versteckte und herausgestellte Knoten zu. Darüber hinaus ermöglicht es, Kollisionen so „auszuschalten“, dass ein Empfänger sowohl versteckte als auch herausgestellte Kollisionen sequentiell erfährt. Basiert drauf, dass der Entwurf des Emulators empfängerorientiert ist, bzw. der Emulator iteriert jeden Empfänger für eine bestimmte Zeitsegmente, werden die Broadcastingübertragungen bedingungslos bearbeitet [flto01].

7.1.3 Bewertung

In den letzten zwei Abschnitten wurden NS-E und Jemu als Beispiele von zentralisierten Methoden vorgestellt. Jedoch ist die Skalierbarkeit dieser Methoden schlecht, weil der gesamte Netzwerkverkehr an eine zentrale Instanz gesendet und bearbeitet werden muss. Dies wird bei großen Szenarien dazu führen, dass die Ankunftsrate der Pakete in der zentralen Instanz so hoch ist, dass sie nicht mehr rechtzeitig bearbeitet werden können.

7.2 Verteilte Methode

7.2.1 EthEmu

EthEmu wurde entwickelt von Steffen Maier an der Abteilung Verteilte Systeme der Universität Stuttgart. EthEmu ist ein Emulationswerkzeug für Netze mit gemeinsamem Medium, speziell für IEEE 802.3 Ethernet. In EthEmu wurde das CSMA/CD-Verfahren in der MAC-Schicht von IEEE 802.3 mit der Übertragungsrate 10MBit/s nachgebildet. EthEmu wurde als Linux-Modul implementiert und kann bei allen Knoten in NET installiert werden.

Jede Instanz von EthEmu verwaltet zwei Timers und zwei Warteschlangen jeweils für das Senden und den Datenempfang. Die Kanalbelegung wird durch lokale Variablen „ts“ (Anfang der Belegung) und „tsr“ (Ende der Kanalbelegung) realisiert. Bei jedem Senden bzw. Empfang werden die beiden Variablen aktualisiert. Ein Zeiger „aktiv“ verweist auf den gerade „aktiven“ Rahmen. Das senden kann nur gestartet werden, wenn die lokale Zeit $t \geq$ „tsr“ ist. Da die Übertragungszeit eines Rahmens in NET (1GBit/s) wesentlich kürzer als in normalem Ethernet ist, muss ein Rahmen künstlich

verzögert gesendet bzw. empfangen werden. Wenn der Kanal als frei erkannt wurde, wird der zu sendende Rahmen zuerst in der Send-Warteschlange gespeichert und der Zeiger „aktiv“ auf ihn gesetzt. Gleichzeitig wird der Send-Timer gestartet. Wenn das Senden innerhalb der realen Übertragungsdauer dieses Rahmens nicht durch den Empfang eines anderen Rahmens gestört wird, liegt eine erfolgreiche Sendung vor. Danach darf der Rahmen aus der Send-Warteschlange weggeworfen werden. Wenn nicht, ist Kollision aufgetreten. Wie bei CSMA/CD wird das Senden sofort abgebrochen. Ein erneuter Sendeversuch ist zum „tsr“ geplant (BackOff ist erforderlich.).

EthEmu arbeitet auf der niedrigsten möglichen Software-Schicht im Protokollstapel und erlaubt es daher, Protokolle und Anwendungen auf allen darüber liegenden Netzschichten zu testen [maie02]. Im Gegensatz zu den im letzten Abschnitt vorgestellten zentralen Emulationsmethoden ist EthEmu eine völlig verteilte Lösung, die prinzipiell die Emulation beliebig großer Szenarien ermöglicht. Allerdings benutzt EthEmu die Standard-Linux-Timer mit 10 ms-Granularität und erlaubt damit eine realistische Trägererkennung nur bei sehr geringen Bandbreiten.

Kapitel 8 Zusammenfassung und Ausblick

8.1 Zusammenfassung

In dieser Diplomarbeit wurde das in IEEE 802.11 beschriebene MAC-Protokoll detailliert analysiert. Nach dem Entwicklungsstand des NET-Projekts und den Bedürfnissen der Abteilung Verteilte Systeme an der Universität Stuttgart ist der „CSMA/CA mit RTS/CTS“-Algorithmus zur Nachbildung ausgewählt worden. Wegen der begrenzten Granularität des Linux-Kernel-Timers wurde der APIC-Timer zur Entwicklung benutzt. Damit erreicht die Zeitgenauigkeit den Bereich von Mikrosekunden.

Die in dieser Arbeit entwickelte Software „WILEMU“ ist ein funktionsfähiges Werkzeug zur Emulation der Medienzugriffsschicht in Wireless LAN. Das in IEEE 802.11 spezifizierte Zugriffsverfahren „CSMA/CA mit RTS/CTS“ wurde mit den in Kapitel 4 erläuterten Vereinfachungen und Annahmen nachgebildet. WILEMU ist das erste verteilte Werkzeug zur Emulation der MAC-Schicht für drahtlose Netze. Durch den komplett verteilten Emulationsansatz von WILEMU ist es nun möglich, auch große Szenarien mit vielen WLAN-Knoten realistisch zu emulieren. Durch den erstmaligen Einsatz der APIC-Timer in einem Emulationswerkzeug funktioniert die virtuelle Trägererkennung auch bei hohen Bandbreiten.

8.2 Ausblick

In dieser Diplomarbeit wurde das Zugriffsverfahren „CSMA/CA mit RTS/CTS“ nachgebildet. Dieses funktioniert sowohl im Infrastruktur-Modus, als auch in Ad-hoc-Netzwerken. Das in IEEE 802.11 ebenfalls beschriebene PCF Verfahren, das nur in Infrastruktur-Netzwerken zum Einsatz kommt, könnte als Erweiterung ebenfalls nachgebildet werden.

Als Option bietet IEEE 802.11 MAC die Fragmentierung von Rahmen. Zum vollständigen Nachbilden der MAC-Schicht in Wireless LAN sollten zukünftige Arbeiten auch Rahmenfragmentierung unterstützen.

Literaturverzeichnis

- [beck99] M. Beck, „*Linux-Kernel-Programmierung : Algorithmen und Strukturen der Version 2.2*“, Addison-Wesley, 1999
- [bewk00] B. Bensaou, Y. Wang und CC. Ko, „*Fair media access in 802.11 based wireless ad-hoc networks*“, in *Proceedings of the IEEE/ACM MobiHOC*, (Boston, MA.), Aug. 2000
- [boce01] D. P. Bovet und M. Cesati, „*Understanding the Linux Kernel*“, O'Reilly, 2001
- [cass02] D. Cavin, Y. Sasson und A. Schiper, „*On the Accuracy of MANET Simulators*“, in *Proceedings of the ACM Principles of Mobile Computing POMC 2002*, October 30-31, Toulouse, France 2002
- [cacg98] F. Cali, M. Conti, E. Gregori, „*IEEE 802.11 Wireless LAN: Capacity and Protocol Enhancement*“, in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, p. 142, San Francisco, California, March/April 1998
- [dena95] A. Desimone und S. Nanda, „*Wireless data: Systems, standards, services*“, *Wireless Networks ACM/Baltzer Wireless Networks Journal*, October 1995
- [dlls96] M. Deininger, H. Lichter, J. Ludewig und K. Schneider, „*Studien-Arbeiten*“, VDF und Teubner Stuttgart, 1996
- [dudk02] D. Dudkowski, „*Emulationskonzepte für Mobilfunk- und Ad-hoc-Netze*“, Diplomarbeit Nr. 2004, Abteilung Verteilte Systeme, Universität Stuttgart, 2002
- [fall99] K. Fall, „*Network Emulation in the Vint/NS Simulator*“, In *Proceedings of the Fourth IEEE Symposium on Computers and Communications (ISCC'99)*, Pages 244-250, Red Sea, Egypt, July 6-8, 1999
- [flto01] J. Flynn, H. Tewari, D. O'Mahony, „*A Real Time Emulation System for Ad-hoc Networks*“, In *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, pages 115-120, San Antonio, Texas, January 27-31, 2002
- [haar98] J. Haartsen, „*Bluetooth: the universal radio interface for ad hoc, wireless connectivity*“, *Ericsson Review* 3, pages 110-117, 1998
- [hbei01] J. Heidemann, N. Bulusu, J. Elson, C. Intanagonwiwat, K. Lan, Y. Xu, W. Ye, D. Estrin, R. Govindan, „*Effects of details in wireless network simulation*“, In *Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2001)*, pages 3-11, Phoenix, Arizona, USA, January 2001
- [hemr03] D. Herrscher, S. Maier, K. Rothermel, „*Distributed Emulation of Shared Media Networks*“, In *Proceedings of the 2003 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2003)*, pages 226-233, Montréal, Quebec, Canada, July 20-24, 2003

- [hemr04] D. Herrscher, S. Maier, J. Tian, and K. Rothermel. "A Novel Approach to Evaluating Implementations of Location-Based Software." In *Proceedings of the 2004 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS '04)*, San Jose, CA, USA, July 25–29 2004, to appear.
- [hero02] D. Herrscher, K. Rothermel, „A Dynamic Network Scenario Emulation Tool“, In *Proceedings of the 11th International Conference on Computer Communications and Networks (ICCCN 2002)*, pages 262-267, Miami, October, 2002
- [herr01] D. Herrmann, „Effektiv Programmieren in C und C++“, Vieweg Verlag, 2001
- [hett01] A. B. Hettich, „Leistungsbewertung der Standards HIPERLAN/2 und IEEE 802.11 für drahtlose locale Netze“, Wissenschaftsverlag Mainz, 2001
- [IEEE811] „Wireless LAN medium access control (MAC) and physical layer (PHY) specifications“, ANSI/IEEE Std 802.11, 1999 Edition
- [IEEE811b] „Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: High-speed physical layer extension in the 2.4 GHz band“, IEEE Std 802.11b-1999, 1999,
<http://standards.ieee.org/getieee802/download/802.11b-1999.pdf>
- [iser99] R. Isernhagen, „Softwaretechnik in C und C++“, Carl Hanser Verlag, 1999
- [john99] D. B. Johnson, „Validation of Wireless and Mobile Network Models and Simulation“, in *Proceedings of the DARPA/NIST Workshop on Validation of Large-Scale Network Models and Simulation*, Fairfax, VA, May 1999
- [jups03] J. Jun, P. Peddabachagari, M. Sichitiu, „Theoretical Maximum Throughput of IEEE 802.11 and its Applications“, in *Proceedings of the Second IEEE International Symposium on Network Computing and Applications*, Cambridge, April 2003
- [kgms01] M. Kojo, A. Gurtov, J. Manner, P. Sarolahti, T. Alanko, K. Raatikainen, „Seawind: A wireless network emulator“, In *Proceedings of the 11th GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB 2001)*, Aachen, Germany, September 11-14, 2001
- [kwfl03] Y. Kwon, Y. Fang, H. Latchman, „A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs“, in *Proceedings of the IEEE INFOCOM*, San Francisco, April 2003
- [lisc03] R. Lischner, „C++ in a Nutshell“, O'Reilly, 2003
- [maie02] S. Maier, „Emulationskonzepte für Netze mit gemeinsamem Medium“, Diplomarbeit-Nr. 2000, Abteilung Verteilte Systeme, Universität Stuttgart, 2002
- [mast00] N. Matthew and R. Stones, „Linux-Programmierung“, MIT Press, 2000

- [nemg01] E. Nett, M. Mock, M. Gergeleit, „*Das drahtlose Ethernet*“, Addison-Wesley, 2001
- [peni03] Z. Pei und L. M. Ni, „*EMPOWER: A Network Emulator for Wireline and Wireless Networks*“, In *Proceedings of the Conference on Computer Communications (INFOCOM-2003)*, pages 1933-1942, San Francisco, March 30-April 3, 2003
- [rizz97] L. Rizzo, „*Dummysnet: A simple approach to the evaluation of network protocols*“, *ACM Computer Communication Review*, pages 31-41, January 1997
- [roth01] K. Rothermel, „*Skript zur Vorlesung Rechnernetze I*“, Universität Stuttgart, 2001
- [roth02] J. Roth, „*Mobile Computing: Grundlagen, Technik und Konzepte*“, Dpunkt Verlag, 2002
- [ruco01] A. Rubini and J. Corbet, „*Linux Device Drivers*“, O'Reilly, 2nd edition, June 2001
- [schi00] J. Schiller, „*Mobile Communications*“, Addison-Wesley, 2000
- [shar02] S. Sharma, „*Analysis of 802.11b MAC: A QoS, Fairness and Performance Perspective*“, Technical Report TR-126, Experimental Computer Systems Lab, Department of Computer Science, State University of New York, 2002
- [siko01] A. Sikora, „*Wireless LAN*“, Addison-Wesley, 2001
- [stro98] B. Stroustrup, „*Die C++ Programmiersprache*“, Addison-Wesley, 1998
- [tane97] A. S. Tanenbaum, „*Computernetzwerke*“, Prentice Hall, 1997
- [vekr93] H. Velayos, G. Karlsson, „*Techniques to Reduce IEEE 802.11b MAC Layer Handover Time*“, KTH Technical Report TRITA-IMIT-LCN R 03:02, ISSN 1651-7717, ISRN KTH/IMIT/LCN/R-03/02--SE, Stockholm, Sweden. April 2003
- [whlg02] B. White, J. Lepreau, S. Guruprasad, „*Lowering the Barrier to Wireless and Mobile Experimentation*“, In *First Workshop on Hot Topics in Networks (Hotnets-I)*, Princeton, New York, USA, October 28-29, 2002
- [wlsr02] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, A. Joglekar, „*An Integrated Experimental Environment for Distributed Systems and Networks*“, In *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI-'02)*, pages 255-270, Boston, MA, December 2002
- [wprm02] K. Wehrle, F. Pahlke, H. Ritter, D. Muller, M. Bechler, „*Linux Netzwerkarchitektur: Design und Implementierung von Netzwerkprotokollen im Linux-Kern*“, Addison-Wesley, 2002

Ich versichere, dass ich diese Arbeit selbständig verfasst und nur die angegebenen Hilfsmittel verwendet habe.

Stuttgart, den

Zhenxiang Yang