

Kurzbericht (Teil I des Schlussberichts)

<i>Projekttitel</i>	Software-Driven Urban plus Rural Area Communication Networks
<i>Kurztitel</i>	SupraCoNeX
<i>Förderkennzeichen</i>	16KIS1197K
<i>Zuwendungsempfänger</i>	Hochschule Nordhausen (NDH)
<i>Projektzeitraum</i>	Geplant: 01.08.2020 - 31.07.2023, Tatsächlich: 01.08.2020 - 31.01.2024

1. Einleitung

Das Forschungsprojekt SupraCoNeX leistet einen Beitrag zur Vision einer anwendungsspezifischen, KI-basierten Ressourcenzuweisung und eines effizienten Managements heutiger 802.11-basierter drahtloser Netzwerke (WLAN) im Heim-, Unternehmens- und Providerkontext. Insbesondere zielte das Projekt darauf ab, statistische und adversarial Multi-Armed Bandit (MAB) Algorithmen für ein breites Spektrum von Anwendungen in drahtlosen Netzen mit spezifischen Anforderungen an die Kommunikationsqualität hinsichtlich Durchsatz, Latenz, Sicherheit und Robustheit zu entwickeln und in produktiven Kommunikationsnetzen mit realem Internetverkehr zu evaluieren.

WiFi-Ressourcenallokation (RA) beinhaltet die dynamische Auswahl von Parametern zur Optimierung einer Metrik, wie z.B. des Durchsatzes, als Reaktion auf einen sich ändernden Kanal. Auf der Senderseite umfasst die Parameterliste die Modulations- und Codierungsschemata (MCS), die Sendeleistung und die Rahmenaggregationslänge. Da der Raum für die Parametersuche mit jeder neuen IEEE 802.11-Änderung größer wird und die Netzwerkdicke ständig zunimmt, bleibt die Aufgabe der WiFi-RA entscheidend für die Gewährleistung einer optimalen Quality of Service (QoS).

Ziel dieser Arbeit war es, den Mangel an Algorithmen für die mehrdimensionale Ressourcenallokation und das effiziente Management heutiger WLAN-Systeme zu beheben. Darüber hinaus war die Einbeziehung praktischer Systemparameter heutiger kommerzieller WLAN-Hardware ein wesentlicher Bestandteil unserer Beiträge.

2. Ablauf des Vorhabens

Im Rahmen des Projekts wurden mehrere wichtige Meilensteine im Hinblick auf die Projektziele erreicht. Wir begannen mit einer gründlichen Untersuchung des Stands der Technik im Bereich WiFi RA, der Implementierung und Evaluierung neuer Algorithmen und ihrer Anwendbarkeit auf COTS-Hardware. Diese Studie zeigt deutlich das Fehlen einer Open-Source-Toolchain für Experimente mit realen Netzwerken. Die Forschung in diesem Bereich wurde bisher hauptsächlich theoretisch oder über Simulationen durchgeführt. Mit diesem Projekt haben wir einen wichtigen Beitrag zu einer echten Open-Source-Forschungs-Toolchain geleistet, die die Entwicklung anspruchsvoller RA-Algorithmen und das Experimentieren mit realen Netzwerken ermöglicht.

Die Open-Source Resource Control API für reale IEEE 802.11 Netzwerke (ORCA) ist das Kernelement unserer Toolchain. ORCA ist eine Linux-Kernel-Benutzer-API mit Funktionen zur Überwachung und Steuerung von WiFi-RA-Parametern. Kontrollparameter, einschließlich der Modulations- und Kodierungsschemarate (MCS) und der Sendeleistung, können pro Station kommentiert werden. Die Überwachung erfolgt auf Frame-Basis mit Informationen wie der Frame-Anzahl für Versuche und Erfolg für verwendete MCS-Raten und Sendeleistungen für gesendete Frames sowie den RSSI-Werten pro Antenne für empfangene Frames. Diese feine Granularität ermöglicht eine kundenspezifische RA auf Stationsbasis, was für zukünftige Netzmanagementtechniken von entscheidender Bedeutung ist. ORCA ist eine Erweiterung innerhalb des Linux mac80211-Subsystems und wurde auf OpenWrt-basierten WiFi-Geräte mit IEEE 802.11n- und IEEE 802.11ac-konformen Chipsätzen getestet, die von den Open-Source-Treibern ath9k bzw. mt76 unterstützt werden. Das nächste Element ist der Remote Control and Telemetry Daemon (RCD), der die Handhabung von RA über einen entfernten Knoten

erleichtert. Mit Hilfe des RCD kann ein zentraler Controller RA über mehrere APs auf einer pro-STA-Basis handhaben. ORCA und RCD sind in C geschrieben. Um RA über mehrere Knoten effizient und gleichzeitig zu verarbeiten, haben wir das Python-Paket RateMan entwickelt.

RateMan verwaltet die Verbindung zu einer gegebenen Liste von APs. Für jede verbundene Station sammelt es Überwachungsinformationen und führt Parametereinstellungen durch. Die Auswahl der Parameter wird durch den RA-Algorithmus vorgenommen. Die Algorithmen sind als unabhängige Python-Pakete abstrahiert, die RateMan als Schnittstelle verwenden, um die Funktionen von ORCA über das RCD zu nutzen. Unter Verwendung der ORCA-RCD-RateMan-Toolchain haben wir mehrere RA-Algorithmen für den Anwenderbereich implementiert. Die bekanntesten davon sind Py-Minstrel-HT, Minstrel-HT-Blues und das Thompson Sampling-basierte RA.

3. Wesentliche Ergebnisse

Was die wichtigsten Beiträge und Ergebnisse betrifft, so ist einer der Wesentlichen die WiFi RA Toolchain auf der Grundlage von ORCA. Diese erste Open-Source-Entwicklungs- und Experimentier-Toolchain ihrer Art ermöglicht es Forschern, neue RA-Algorithmen effektiv zu implementieren und zu testen.

Um unsere Toolchains zu verifizieren, als Zusammenarbeit zwischen NDH und HHI wurde der Minstrel-HT-Algorithmus im Userspace implementiert. Der „Py-Minstrel-HT“ ermöglicht im Gegensatz zur Kernel-Implementierung eine dynamische Neukonfiguration von Parametern wie Aktualisierungsintervall, Abtastintervall und Abtastsequenz. Darüber hinaus unterstützt es das Vorladen von Ratenstatistiken als a priori-Wissen, wenn sich eine Station neu anmeldet, und kann den Suchraum mithilfe des „Rate Ordering“-Schemas verkleinern. Der Minstrel-HT-Algorithmus wird auch auf die Leistungssteuerung für die gemeinsame Ressourcenallokation auf der Grundlage der Minstrel-Blues-Implementierung erweitert. Der Minstrel-HT-Blues-Algorithmus kann die Leistung erheblich reduzieren und gleichzeitig den optimalen Durchsatz beibehalten.

4. Zusammenarbeit mit anderen Forschungseinrichtungen

Während der Projektlaufzeit gab es eine regelmäßige und nachhaltige Zusammenarbeit zwischen den Projektpartnern. Die Stärke des Konsortiums, akademische Einrichtungen wie die Hochschule Nordhausen und die TU Braunschweig, angewandte Forschungsinstitute wie das Fraunhofer HHI und Industriepartner wie NewMediaNet und BISDN zusammenzubringen, hat sich als besonders vorteilhaft erwiesen. Darüber hinaus standen wir in engem Austausch mit der Freifunk- und OpenWrt-Community, um unsere Forschungsergebnisse zu verbreiten und wertvolles Feedback von Entwicklern und Forschern zu erhalten.

Ausführlicher Schlussbericht

<i>Projekttitel</i>	Software-Driven Urban plus Rural Area Communication Networks
<i>Kurztitel</i>	SupraCoNeX
<i>Förderkennzeichen</i>	16KIS1197K
<i>Zuwendungsempfänger</i>	Hochschule Nordhausen (NDH)
<i>Projektzeitraum</i>	Geplant: 01.08.2020 - 31.07.2023, Tatsächlich: 01.08.2020 - 31.01.2024

Inhaltsverzeichnis

Inhaltsverzeichnis.....	1
1. Einleitung.....	1
2. Arbeitspakete.....	3
AP 0 – Projektmanagement.....	3
AP 1 – Identifizierung der Systemzusammenhänge und Anforderungen.....	3
1.1 Anwendungsfallanalyse.....	3
1.2 Anforderungsanalyse.....	3
1.3 Identifizierung der Systemrestriktionen von Hard- und Software-Komponenten.....	6
AP 2 – KI/ML Modellbildung und Algorithmenentwicklung für das SupraCoNeX Design.....	7
2.1 Modellbildung.....	7
2.2 Algorithmenentwicklung.....	7
2.3 Ergebnisse in Simulation und Experimenten.....	7
AP 3 – Design, Implementierung und Validierung der SupraCoNeX Systemarchitektur.....	8
3.1 Design der Gesamtarchitektur.....	8
3.2 Implementierung der lokalen Algorithmen zur Ressourcenallokation.....	9
3.3 Implementierung der SDN-Architektur (Control / Dataplane Split).....	10
3.4 Implementierung der SupraCoNeX Analytics Engine.....	10
3.5 Validierung, Evaluierung und System-Performanzanalyse der SupraCoNeX Komponenten.....	10
AP 4 – Experimentalumfeld und Demonstrator.....	11
4.1 Spezifikation der Testszenarien.....	11
4.2 Deployment des SupraCoNeX Systems.....	11
4.3 Evaluation und Performance-Analyse des SupraCoNeX Systems.....	13
4.4 Aufbau und Umsetzung der Demonstratoren.....	13
AP 5 – Dissemination - Verbreitung der Ergebnisse und Dokumentation.....	13
3. Voraussichtlicher Nutzen und Verwertbarkeit des Ergebnisses.....	14
3.1 Diskussion über den erwarteten Nutzen des Projekts und die Verwertbarkeit der Ergebnisse.....	14
3.2 Konkrete Planungen für die nächste Zukunft, insbesondere im Rahmen des fortgeschriebenen Verwertungsplans.....	14
4. Zusammenfassung und Ausblick.....	14
5. Veröffentlichungen des Ergebnisses.....	16
6. Wichtige Positionen des zahlenmäßigen Nachweises.....	17
7. Literaturverzeichnis.....	18

1. Einleitung

Dieser Bericht bietet einen Überblick über das Projekt **Software-Driven Urban plus Rural Area Communication Networks (SupraCoNeX)** und die wichtigsten vom Hochschule Nordhausen (NDH) erzielten Ergebnisse.

Das SupraCoNeX-Projekt befasste sich mit dem Thema Ressourcenallokation in WiFi-Netzwerken. Die WiFi-Ressourcenallokation (RA) ist ein Parameteranpassungsprozess der MAC-Schicht, der darauf abzielt, einen Parametersatz zur Optimierung der Netzwerkleistung basierend auf einer Metrik wie dem Durchsatz auszuwählen. Auf der Senderseite umfasst der Parametersatz unter anderem Modulationstyp, Codierungsrate, Schutzintervall, Anzahl der räumlichen Ströme, Kanalbandbreite und Sendeleistung. Bei einem dynamischen drahtlosen Kanal mit Mobilität der Clients erhält ein Zugangspunkt ständig Netzwerkfeedback, anhand dessen eine Auswahl des Parametersatzes für die nächste Übertragungsmöglichkeit getroffen wird.

Mit der weit verbreiteten Nutzung von IEEE 802.11-basierten WiFi-Netzwerken besteht ein dringender Bedarf an neuen Techniken und Frameworks zur Verbesserung der Gesamtnetzworleistung. Mit den Fortschritten bei den IEEE 802.11-Standards und der zunehmenden Dichte moderner drahtloser Netzwerke wächst die Komplexität der Ressourcenallokationsaufgabe entsprechend weiter. Obwohl sich eine große Menge an Forschung und Entwicklung mit diesem Thema befasst, basieren die modernsten Algorithmen zur RA auf Heuristiken. Darüber hinaus war der Großteil der Forschung in diesem Bereich hauptsächlich theoretisch oder wurde durch Simulationen ausgewertet [13]. Innerhalb von SupraCoNeX konzentrierte sich unsere Arbeit auf die praktische Implementierung neuartiger Algorithmen zur RA und deren Auswertung mit COTS-Hardware. Unsere Forschungsergebnisse sind wie folgt:

- **Open-Source Research Control API (ORCA):** Eine Linux-Kernel-Userspace-API zur Überwachung und Steuerung von WLAN-MAC-Layer-Parametern. Sie ermöglicht eine neuartige Toolchain für die Forschung an fortschrittlichen ML-Algorithmen zur Lösung des RA-Problems. Die Fähigkeit, RA je Client zu handhaben, ermöglicht Verkehrsformung und feinkörniges Netzwerkmanagement. Der Remote Control and Telemetry Daemon (RCD) ermöglicht die Nutzung der Funktionen von ORCA über einen zentralen Knoten, der RA über OBSSs mit mehreren APs ermöglicht.
- **RateMan:** Python-Paket für die Userspace-Handhabung von RA über mehrere APs auf STA-Basis. Seine benutzerfreundlichen Schnittstellen ermöglichen es Forschern, neue RA-Algorithmen schnell zu implementieren und zu bewerten.
- **Experimentelles Testbed:** Einfaches Schreibtisch-Setup für schnelle Tests, kontrollierte Testumgebung mit HF-isolierten WLAN-Knoten für reproduzierbare Experimente und Knoten mit drahtlosem Produktions-ISP-Netzwerk.

Der Bericht ist wie folgt aufgebaut. Abschnitt 2 beschreibt die geplanten Arbeitspakete, die Aufgabenerfüllung und die wichtigsten Ergebnisse. Abschnitt 3 diskutiert den konkreten Nutzen unserer Forschung und skizziert die Richtung für weitere Studien. Abschnitt 4 bietet eine knappe Zusammenfassung und einen Ausblick. Abschnitt 5 bietet einen Überblick über eingereichte und geplante Forschungsarbeiten. Abschnitt 6 enthält die Bibliographie und Abschnitt 7 schließlich beschreibt den Finanzplan und die Ausgaben während SupraCoNeX.

2. Arbeitspakete

AP 0 – Projektmanagement

Um den Erfolg des Projekts sicherzustellen, wurden mehrere Schlüsselstrategien umgesetzt. Regelmäßige Meetings wurden abgehalten, um eine konsistente Kommunikation zwischen den Partnern aufrechtzuerhalten und eine Plattform zu bieten, um den Fortschritt zu besprechen, etwaige Probleme anzusprechen und sich auf Projektziele zu einigen. Es wurden themenspezifische Workshops durchgeführt, um tiefer in bestimmte Bereiche des Projekts einzutauchen. Diese Workshops ermöglichten es den Teammitgliedern, bestimmte Aspekte besser zu verstehen und so Fachwissen und Fähigkeiten innerhalb des Teams zu fördern. Zusätzlich wurden persönliche Meetings organisiert, um die direkte Interaktion zu erleichtern, die Zusammenarbeit zu verbessern und stärkere Arbeitsbeziehungen aufzubauen.

AP 1 – Identifizierung der Systemzusammenhänge und Anforderungen

1.1 Anwendungsfallanalyse

Unsere Netzwerktopologie besteht aus mehreren Overlapping Basic Service Set (OBSS) - *überlappenden Basisdienstgruppen*. Jedes Basic Service Set (BSS) besteht aus einem Access Point (AP), der Dienste für mehrere Clients oder Stationen (STAs) bereitstellt. Diese Zugangspunkte sind über ein Backbone-Netz miteinander verbunden, das ihnen den Anschluss an das Internet ermöglicht. In dieser Netzkonfiguration wird jede Übertragung zwischen verschiedenen BSS als Interferenz behandelt, die die Gesamtleistung des Netzes beeinträchtigen kann und sorgfältige Verwaltungs- und Entschärfungsstrategien erfordert.

In diesem Projekt liegt unser Schwerpunkt auf der Verwendung von Open-Source-Hardware und Software-Tools, um sicherzustellen, dass die Ergebnisse unserer Forschung für eine breitere Gemeinschaft zugänglich und reproduzierbar sind. Um dieses Ziel zu erreichen, haben wir WiFi-Zugangspunkte eingesetzt, die auf OpenWrt basieren. OpenWrt ist ein Linux-basiertes Open-Source-Betriebssystem, das für seine Robustheit und die aktive Beteiligung einer engagierten Entwicklergemeinschaft bekannt ist. Diese Plattform bietet eine flexible und anpassbare Umgebung für die Entwicklung und das Testen von drahtlosen Netzwerklösungen.

Was die WiFi-Chipsätze betrifft, so stützt sich unsere Forschung auf diejenigen, die von den Treibern ath9k [10] und mt76 [4] unterstützt werden. Insbesondere haben wir uns für die Atheros AR900x- und MediaTek MT76XX-Chipsatzfamilien [5] entschieden. Diese Chipsätze sind mit den Standards IEEE 802.11n/a und IEEE 802.11ac kompatibel, die in modernen drahtlosen Netzwerken weit verbreitet sind. Die ath9k und mt76 Treiber gehören zu den zuverlässigsten und am besten unterstützten Open-Source-Treibern auf dem Markt und sind daher für unsere Zwecke ideal.

Was die Softwareentwicklung angeht, so haben wir Patches für den Linux-Kernel in der Programmiersprache C entwickelt. Diese Low-Level-Programmierung ist unerlässlich, um die Kernfunktionen des Betriebssystems zu ändern und zu erweitern, um unsere spezifischen Anforderungen zu unterstützen. Darüber hinaus werden die Benutzeroberflächen und die Toolchain mit Python implementiert, einer vielseitigen und weit verbreiteten Programmiersprache. Python eignet sich aufgrund seiner umfangreichen Bibliotheken und seiner Benutzerfreundlichkeit für die Entwicklung benutzerfreundlicher Schnittstellen und Automatisierungsskripte, wodurch der Forschungs- und Entwicklungsprozess rationalisiert wird.

Insgesamt zeigt unser Projekt, dass es möglich ist, Open-Source-Ressourcen zu nutzen, um die Forschung im Bereich der drahtlosen Netzwerke voranzutreiben. Indem wir unsere Methoden und Tools mit der Community teilen, hoffen wir, zum kollektiven Wissen beizutragen und weitere Innovationen in diesem Bereich zu ermöglichen.

1.2 Anforderungsanalyse

In einem typischen Wi-Fi-Netzwerk sind die drahtlosen Kanalbedingungen dynamisch, insbesondere aufgrund der Mobilität der STAs. Wenn sich Benutzer bewegen, unterliegt das Signal Pfadverlusten, Mehrwegeausbreitung und Dopplerverschiebung, was zu Schwankungen der Signalstärke und -qualität führt. Diese Veränderungen

können zu Schwankungen in der Zuverlässigkeit und Leistung der Verbindung führen. Ein AP muss sich an diese Bedingungen anpassen, indem er die Übertragungsparameter dynamisch anpasst, um eine stabile Verbindung aufrechtzuerhalten. Im Folgenden finden Sie eine Liste der Übertragungsparameter, die angepasst werden müssen, um eine Kennzahl wie den Durchsatz zu optimieren.

- **Modulationsart:** Die Modulationsart bezieht sich auf die Methode, mit der Daten auf dem Trägersignal kodiert werden. Zu den bei Wi-Fi gebräuchlichen Modulationsverfahren gehören die binäre Phasenumtastung (BPSK), die Quadraturphasenumtastung (QPSK) und die Quadraturamplitudenmodulation (QAM). Modulationen höherer Ordnung wie 64-QAM oder 256-QAM ermöglichen höhere Datenraten, indem sie mehr Bits pro Symbol kodieren, erfordern jedoch ein besseres Signal-Rausch-Verhältnis (SNR). Die dynamische Auswahl des geeigneten Modulationstyps auf der Grundlage der Kanalbedingungen trägt zu einem ausgewogenen Verhältnis zwischen Datenrate und Zuverlässigkeit bei.
- **Kodierungsrate:** Die Kodierungsrate bestimmt den Anteil der übertragenen Daten, der aus tatsächlichen Informationen und nicht aus fehlerkorrigierendem Code besteht. Eine Rate von 3/4 bedeutet, dass 75 % der Bits aus Daten und 25 % aus Fehlerkorrektur bestehen. Niedrigere Kodierungsraten bieten mehr Fehlerkorrektur, was die Zuverlässigkeit in verrauschten Umgebungen verbessert, während höhere Kodierungsraten den Durchsatz bei guten Kanalbedingungen erhöhen. Wi-Fi-Standards wie 802.11n und 802.11ac verwenden Techniken wie Faltungscodierung und LDPC (Low-Density Parity-Check) zur Verbesserung der Fehlerkorrektur.
- **Räumliche Datenströme (Spatial Streams):** Spatial Streams (räumliche Datenströme) beziehen sich auf die Verwendung mehrerer Antennen sowohl auf der Sender- als auch auf der Empfängerseite, um mehrere Datenströme gleichzeitig zu senden. Diese Technik, die als Multiple Input Multiple Output (MIMO) bezeichnet wird, erhöht die Datenrate und die Gesamtkapazität des Wi-Fi-Netzwerks. 4x4 MIMO bedeutet zum Beispiel, dass vier räumliche Datenströme gesendet und empfangen werden können. Durch die Verwendung von Spatial Streams wird die verfügbare Bandbreite effektiv vervielfacht und die Zuverlässigkeit verbessert, insbesondere in Umgebungen mit hoher Gerätedichte.
- **Schutzintervall (Guard Interval):** Das Schutzintervall ist eine kurze Zeitspanne, die zwischen den übertragenen Symbolen eingefügt wird, um Intersymbolinterferenzen (ISI) zu verhindern, die aufgrund von Mehrwegeausbreitung auftreten. Standardmäßig beträgt das Schutzintervall 800 Nanosekunden, aber in einigen Wi-Fi-Standards werden kürzere Intervalle von 400 Nanosekunden verwendet, um die Datenraten zu erhöhen. Kürzere Schutzintervalle sind jedoch in Umgebungen mit starken Reflexionen anfälliger für ISI, so dass die Wahl von den spezifischen Bedingungen des Wi-Fi-Netzwerks abhängt.
- **Bandbreiten:** Die Bandbreite in Wi-Fi bezieht sich auf die Breite des für die Übertragung verwendeten Frequenzkanals. Zu den Standard-Kanalbandbreiten gehören 20 MHz, 40 MHz, 80 MHz und sogar 160 MHz bei den neuesten Standards wie 802.11ac und 802.11ax. Größere Bandbreiten ermöglichen höhere Datenraten, da mehr Unterträger untergebracht werden können, aber sie erfordern auch ein sauberes, weniger überlastetes Spektrum. Die dynamische Anpassung der Bandbreite auf Grundlage der Kanalbedingungen und der Netzwerküberlastung kann die Leistung optimieren.
- **Sendeleistung:** Die Sendeleistung (TX Power) ist die Stärke des vom Wi-Fi-Sender gesendeten Signals. Eine höhere Sendeleistung erhöht die Reichweite und die Durchdringung des Signals, kann aber zu stärkeren Interferenzen mit anderen Netzen führen und verbraucht mehr Energie. Eine geringere Sendeleistung hingegen reduziert Störungen und spart Energie, schränkt aber die Reichweite ein. Adaptive Algorithmen passen die Sendeleistung auf der Grundlage der Entfernung zwischen dem Zugangspunkt und dem Client-Gerät sowie der Umgebungsbedingungen an, um ein Gleichgewicht zwischen Reichweite, Interferenzen und Stromverbrauch zu gewährleisten.
- **Frame Aggregation:** Frame Aggregation ist eine Technik, die in Wi-Fi verwendet wird, um mehrere Datenrahmen in einer einzigen Übertragungseinheit zu kombinieren. Dadurch wird der mit jedem Rahmen verbundene Overhead, wie z. B. Präambel und Bestätigung, reduziert, was die Effizienz des Netzes erhöht. Es gibt zwei Haupttypen der Frame-Aggregation in Wi-Fi: Aggregierte MAC-Protokoll-dateneinheit (A-MPDU) und Aggregierte MAC-Dienst-dateneinheit (A-MSDU). A-MPDU aggregiert mehrere MPDUs, während A-MSDU mehrere MSDUs zu einer MPDU zusammenfasst. Durch die Verringerung der Anzahl der gesendeten Frames trägt die Frame-Aggregation zur Verbesserung des Durchsatzes und zur Verringerung der Latenz bei, insbesondere in Umgebungen mit hohem Verkehrsaufkommen.
- **Kanalfrequenz:** Die Kanalfrequenz bezieht sich auf die spezifische Funkfrequenz, die für die Wi-Fi-Kommunikation verwendet wird. Wi-Fi arbeitet in erster Linie in den Bändern 2,4 GHz und 5 GHz, wobei neuere Standards wie Wi-Fi 6E auch das 6-GHz-Band nutzen. Jedes Band ist in mehrere Kanäle

unterteilt, wobei Kanalbreiten von 20 MHz, 40 MHz, 80 MHz und 160 MHz zur Verfügung stehen. Die Wahl der Kanalfrequenz wirkt sich auf Störungen, Reichweite und Datenrate aus. Höhere Frequenzen (z. B. 5 GHz und 6 GHz) bieten mehr Kanäle und höhere Datenraten, haben aber im Vergleich zum 2,4-GHz-Band eine geringere Reichweite und eine geringere Wanddurchdringung. Die dynamische Auswahl der optimalen Kanalfrequenz trägt dazu bei, Störungen durch andere Netze und Geräte zu vermindern und die Gesamtleistung des Netzes zu verbessern.

Diese Übertragungsparameter werden von modernen Wi-Fi-Systemen dynamisch angepasst, um die Leistung auf der Grundlage von Echtzeit-Kanalbedingungen und Nutzeranforderungen zu optimieren und eine robuste und effiziente drahtlose Kommunikation zu gewährleisten.

Modulations- und Kodierungsschema (MCS) Tabelle

Die MCS-Tabelle [14] definiert verschiedene Kombinationen von Modulation, Kodieraten und räumlichen Datenströmen, um unterschiedliche Datenraten zu erreichen. Sie ist eine Nachschlagetabelle für die theoretischen Übertragungsdatenraten, die bei einer bestimmten Parameterkonfiguration zu erwarten sind. Die Tabellengröße wurde schrittweise von 128 möglichen Raten im 802.11n-Standard auf 620 im 802.11ac-Standard und auf mehr als 1000 im 802.11ax-Standard erweitert. Der MCS-Index ermöglicht es einem WiFi-Gerät, sich an die unterschiedliche Qualität des drahtlosen Kanals anzupassen. Unter guten Bedingungen werden höhere MCS-Indizes (mit höheren Modulations- und Kodierungsraten) verwendet, was zu einem höheren Durchsatz führt. Unter schlechten Bedingungen werden niedrigere MCS-Indizes verwendet, um eine zuverlässige Verbindung aufrechtzuerhalten. Verschiedene Geräte unterschiedlicher Hersteller können nahtlos zusammenarbeiten, wenn sie sich an die MCS-Definitionen halten. Die MCS-Tabelle ist genormt, damit die Geräte effizient kommunizieren können.

Beispiel MCS-Tabelle für WiFi 5 (802.11ac) für Einstellung mit einzelner räumlicher Datenstrom, mit einer Bandbreite von 80 MHz und einem Schutzintervall von 0,4 Mikrosekunden.

MCS Index	Modulationsart	Kodierungsrate	Theoretische Datenrate (Mbit/s)
0	BPSK	1/2	32.50
1	QPSK	1/2	65.00
2	QPSK	3/4	97.50
3	16-QAM	1/2	130.00
4	16-QAM	3/4	195.00
5	64-QAM	2/3	260.00
6	64-QAM	3/4	292.50
7	64-QAM	5/6	325.00
8	256-QAM	3/4	390.00
9	256-QAM	5/6	433.33

Konzepte im Zusammenhang mit der Anpassung von Parametern der Wi-Fi-MAC-Schicht für praktische Systeme [9]:

Bei einem dynamischen drahtlosen Kanal muss die Anpassung der Parameter pro Rahmen erfolgen, um präzise auf die Änderungen der Kanalbedingungen reagieren zu können. Die Implementierung einer solchen Anpassung erfordert eine Hardware mit extrem niedriger Latenzzeit. Dies ist in praktischen Netzen mit COTS-Hardware nicht machbar. Hardware mit niedriger Latenz erhöht zwangsläufig die Herstellungskosten. Die Kohärenzzeit für ein typisches Heim-WLAN liegt zwischen einigen zehn Millisekunden und einer Sekunde. Dies ermöglicht die Anpassung von Parametern über mehrere Frames hinweg und nicht auf Frame-Basis.

Aktualisierungsintervall (Update Interval): Die Zeitspanne, nach der die Übertragungsparameter der Wi-Fi-MAC-Schicht aktualisiert werden, wird als Aktualisierungsintervall bezeichnet. Die meisten Algorithmen

verwenden daher ein Aktualisierungsintervall, in dem Kanalstatistiken gesammelt werden und nach jedem Zeitraum eine Entscheidung zur Parameterkonfiguration getroffen wird.

Multi-Rate-Retry (MRR) Kette: Mehrere WiFi-Chipsätze unterstützen die Aufzeichnung einer Reihe von Raten anstelle einer einzelnen Rate. Dieser Mechanismus ermöglicht die Anpassung innerhalb eines Aktualisierungsintervalls, ohne dass eine Ableitung durchgeführt werden muss, um die besten MCS-Raten zu finden, was CPU-Nutzung erfordert. Die MRR-Kette besteht in der Regel aus einer Reihe von Raten und der entsprechenden maximalen Anzahl von Wiederholungen. Die maximale Anzahl der Wiederholungen legt fest, wie oft ein Rateausprobiert werden sollte, bevor der nächste Rate in der Kette verwendet wird. Die spezifische Anzahl der Rateneinträge (Stufen) in der Kette und die zulässige Anzahl der Wiederholungsversuche hängt von den Spezifikationen des Chipsatzherstellers und des Treibers ab. Für die Atheros AR9000-Chipsatzfamilie, die den ath9k-Treiber verwendet, stehen beispielsweise vier Stufen zur Verfügung, und die maximale Summe der Wiederholungen kann 15 betragen.

1.3 Identifizierung der Systemrestriktionen von Hard- und Software-Komponenten

Die meisten Forschungsarbeiten im Bereich der Anpassungsalgorithmen für die MAC-Schicht wurden mit Simulationswerkzeugen durchgeführt. Experimentelle Forschung, die neue Algorithmen auf WiFi-Geräten in komplexen realen Funkumgebungen evaluiert, gibt es in der akademischen Welt praktisch nicht. Die Gründe dafür sind ziemlich offensichtlich. Für Experimente in realistischen Umgebungen müssen neue Lösungen auf COTS-Geräten entweder in bestehenden Produktionsnetzen oder in groß angelegten Testumgebungen mit vergleichbarer Komplexität eingesetzt werden. Die Implementierung neuer Algorithmen für COTS-WiFi-Geräte erfordert jedoch die Modifizierung von Chipsatztreibern und Firmware, doch ein großer Teil der WiFi-Chipsätze ist Closed-Source mit proprietärer Firmware. Das OpenWrt-Ökosystem bietet einige Möglichkeiten für Open-Source-Entwicklung auf Linux-basierten Geräten. Innerhalb des Linux-Kernels ist die Entwicklung jedoch auf die Verwendung von ANSI-C und Festkomma-Arithmetik beschränkt. Bis zum Zeitpunkt dieses Projekts gibt es also keine quelloffene, einfach zu verwendende Toolchain, die ein effizientes Prototyping von WiFi-RA Algorithmen ermöglicht.

Begrenzter Zugang zu WiFi-Chipsatz-Firmware

Die Handhabung von RA variiert zwischen den verschiedenen Hardware-Plattformen, wobei der am weitesten verbreitete Ansatz die Implementierung in der Firmware von WiFi-Chipsätzen ist. Die meisten Anbieter betrachten ihren RA-Mechanismus als sensibles geistiges Eigentum, das geheim gehalten werden sollte. Die Atheros-Chipsätze der AR5000- und AR9000-Serie sind eine bemerkenswerte Ausnahme. Diese verwenden den völlig freien und quelloffenen ath9k-Treiber, der seit 2008 Teil des Linux-Kernels ist [10]. Aufgrund der offen zugänglichen Dokumentation des Chips und seines Open-Source-Treibers ist er die Plattform der Wahl für die meisten akademischen RA-Forschungen. Nachfolgende Chipsätze, die nach der Übernahme von Atheros durch Qualcomm im Jahr 2011 eingeführt wurden, verwalten RA hingegen über proprietäre Firmware, die Closed-Source ist. Der MT7615-Chipsatz von MediaTek, der IEEE 802.11ac mit bis zu vier räumlichen Streams unterstützt, bietet einen Open-Source-Treiber für Linux-basierte Geräte namens mt76, obwohl er mit einer Closed-Source-Firmware arbeitet. Der Hybrid-RA Modus deaktiviert den RA-Algorithmus der Firmware und erlaubt es dem Treiber, MCS-Raten zu konfigurieren. Infolgedessen ist die Forschung zu WiFi-RA auf der MAC-Schicht unter Verwendung von COTS-Hardware auf die Verwendung der IEEE 802.11a/n/ac-Standards beschränkt. Unser Beitrag in diesem Papier basiert daher auf Forschungen, die an Chipsätzen mit den WiFi-Treibern ath9k und mt76 durchgeführt wurden. Wir verwenden in dieser Arbeit sowohl den Atheros AR9280, der den 802.11a/n-Standard erfüllt und MIMO mit zwei räumlichen Streams unterstützt, als auch den MediaTek MT7615-Chipsatz. Unser Ziel ist die umfassende Nutzung der Steuerungsmöglichkeiten von Chipsätzen, die die ath9k- und mt76-Treiber verwenden.

Verwendung von OpenWrt-basierten WiFi-Geräten

OpenWrt [2] ist ein vielseitiges, Linux-basiertes Betriebssystem, das auf eingebettete Geräte, insbesondere Router, zugeschnitten ist. Sein Open-Source-Charakter fördert eine große Gemeinschaft von Entwicklern und Nutzern, die kontinuierlich zur Weiterentwicklung beitragen, die Sicherheit verbessern, neue Funktionen hinzufügen und eine breite Kompatibilität sicherstellen. Da OpenWrt die werkseitige Firmware ersetzt, können Benutzer die Einschränkungen der Standard-Router-Firmware überwinden, was es zu einer beliebten Wahl für alle macht, die mehr Kontrolle über ihre Netzwerkhardware haben wollen.

Wir haben OpenWrt als das zuverlässigste Open-Source-Betriebssystem für die WiFi-Geräte in unseren experimentellen Testumgebungen identifiziert.

AP 2 – KI/ML Modellbildung und Algorithmenentwicklung für das SupraCoNeX Design

Bei der drahtlosen Kommunikation im WLAN haben wir im Gegensatz zur drahtgebundenen Kommunikation eine Übertragung in einem freien Medium, der Luft, die von vielen Menschen mit ihren Geräten genutzt werden kann, die wir nicht kontrollieren können. Für die Übertragung im WLAN sind nur bestimmte Frequenzbänder vorgesehen. Bekannt sind die Frequenzbänder 2,4 GHz, 5 GHz und 6 GHz, die je nach unterstütztem Standard von Endgeräten und WLAN-Routern unterstützt werden. Allen Bändern gemeinsam ist, dass die Frequenzen begrenzt sind und zwischen den drahtlosen WLAN-Geräten in der Nähe aufgeteilt werden. Im Gegensatz zu zellularen Netzen arbeiten WLAN-Router dezentral und haben keine aktive Kommunikation mit anderen WLAN-Routern und deren Geräten. Dies führt zu Unsicherheiten im WLAN, da nicht bekannt ist, welche Geräte wann und wie lange den Funkkanal nutzen wollen. Außerdem können, je nach Frequenzband, Geräte mit anderen Technologien, wie Bluetooth, Mikrowellen im 2,4 GHz Band oder Medienkonverter im 5 GHz Band, von den WLAN-Geräten nicht erkannt werden. Dies sind alles Unsicherheitsquellen, die dazu führen können, dass bestimmte WLAN-Senderaten nur im Moment optimal sind, sich aber in den nächsten Sekunden ändern können. Dies ist ein Optimierungsproblem mit unbekanntem (sich ändernden) Parametern, die die WLAN-Leistung beeinflussen. Für solche Probleme können statistische Multi-Armed Bandit (MAB) Modelle eingesetzt werden, die Unsicherheiten in der drahtlosen Übertragung durch Interferenzen anderer Geräte und Kanalrauschen in drahtlosen Netzwerken mit Hilfe von Wahrscheinlichkeitsverteilungen modellieren.

2.1 Modellbildung

Die Optimierung des Algorithmus erfordert zudem eine Anpassung an das WLAN-Framework, um die Senderaten korrekt einstellen zu können. Dies bedeutet, dass die Ein- und Ausgänge des MAB-Algorithmus mit denen des WLAN-Systems übereinstimmen müssen. Die Spezifikationen des WLAN und des verwendeten Systems inklusive seiner Hardware und Firmware legen diese Anforderungen fest.

Es wird zwischen notwendigen und optionalen Nebenbedingungen unterschieden. Die Erfüllung notwendiger Nebenbedingungen ist Voraussetzung dafür, dass MAB-Algorithmen überhaupt zur Auswahl der Senderate im WLAN eingesetzt werden können. Eine zwingende Nebenbedingung ist beispielsweise, dass die Senderate nur für bestimmte Zeitintervalle eingestellt werden kann und nicht nach jedem einzelnen Paket. Optionale Nebenbedingungen beinhalten zusätzliche Informationen oder Expertenwissen, die die Explorationszeit des Algorithmus verkürzen und ihm helfen, die optimale Senderate schneller zu finden.

2.2 Algorithmenentwicklung

Die Wahl der optimalen Senderate spielt in der dynamischen Welt der WLAN-Technologie eine entscheidende Rolle für die Netzwerkeffizienz. Wie bereits im Punkt 2.1 besprochen, sind Multi-Armed Bandit (MAB)-Algorithmen hervorragend geeignet, um die Senderaten in WiFi-Netzwerken zu optimieren.

NDH unterstützte zusammen mit NMN und HHI TUBr bei der Anpassung theoretischer MAB-Modelle für die Verwendung mit COTS-WiFi-Hardware. Das Verständnis der Rechen- und Speicherbeschränkungen von praktischem WLAN ist für die Implementierung von MAB-Algorithmen von entscheidender Bedeutung.

2.3 Ergebnisse in Simulation und Experimenten

Bei der Auswahl der Simulationsparameter wurden von NDH Fachwissen und Unterstützung bereitgestellt.

AP 3 – Design, Implementierung und Validierung der SupraCoNeX Systemarchitektur

3.1 Design der Gesamtarchitektur

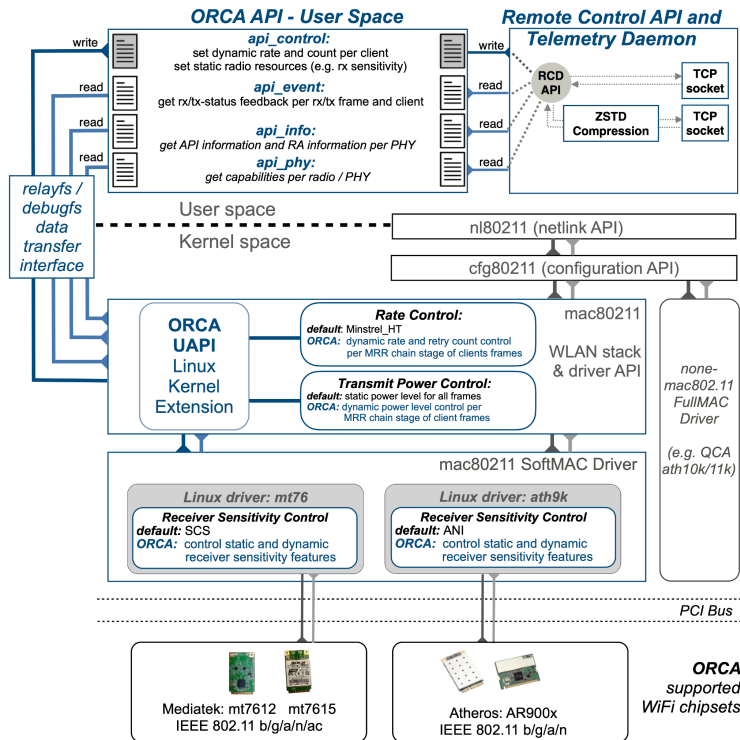


Abbildung 1: Architektur der Linux mac80211-basierten Kernel-User-Space API zur Kontrolle von Ressourcen vom Userspace; Open-Source Ressourcenkontroll-API (engl. open-source resource control API, kurz ORCA) mit Fernsteuerung und Telemetrie Daemon (engl. remote control and telemetry daemon, kurz RCD).

Wir stellen die **Open-Source Resource Control API (ORCA)** als Schlüsselkomponente für die Forschung zur Entwicklung von Algorithmen zur Ressourcenallokation für reale WiFi-Netzwerke vor.

ORCA ist eine leistungsstarke und gut abstrahierte Kernel-User-Space-API im Linux-Kernel zur Überwachung und Steuerung von WiFi-Ressourcen-Kontrollparametern verschiedener WiFi-Chipsätze von verschiedenen Herstellern. Wie in Abb. 1 dargestellt, ist es als Erweiterung des mac80211-Subsystems des Linux-Kernels implementiert. ORCA nutzt Minstrel-HT [7], das in mac80211 enthaltene, standardmäßig verwendete RA-Modul im Linux-Kernel, um Informationen zu erhalten, die für die Überwachung der RA pro Frame und pro Client notwendig sind.

ORCA liefert statische und dynamische Informationen über die Fähigkeiten und Konfigurationen von drahtlosen Access Points und Clients. Dazu gehören eine Liste der unterstützten MCS-Raten, eine Beschreibung des Formats für die verschiedenen Datensätze, die erfasst werden können, sowie mögliche Kommandos zur Steuerung der Parameter. Darüber hinaus sind für jede drahtlose Netzwerkschnittstelle Informationen zu Treiber, virtuellen Schnittstellen, dem Bereich der unterstützten Sendeleistungsstufen und weitere Funktionen und Parameter verfügbar. Darüber hinaus werden für verbundene Clients die unterstützten Raten pro Client zur Verfügung gestellt.

Dynamische Überwachungsinformationen werden über verschiedene Traces erfasst, wobei die txs- und rxs-Zeilen die wichtigsten sind. Eine txs-Zeile meldet das Ergebnis einer ausgehenden Übertragung eines Datenframes. Für jede STA werden die in der MRR-Kette verwendeten MCS-Raten, die jeweiligen Übertragungsversuche und die Sendeleistungspegel angegeben. Die MRR-Kette ist bis zu der Stufe gefüllt, in der die Übertragung erfolgreich war. Zusätzlich wird die Anzahl der aggregierten Sub-Frames innerhalb des Datenframes und die Anzahl der vom Empfänger bestätigten Frames angegeben. Ein Flag zeigt an, ob die Rate in der ersten Stufe der MRR-Kette verwendet wurde, um andere Parameter als die vom RC-Algorithmus ermittelte beste Kombination zu untersuchen. rxs-Zeilen melden den erfolgreichen Empfang eines Datenframes.

Sie enthalten einen aggregierten RSSI-Wert (Received Signal Strength Indicator) und je nach Unterstützung des WiFi-Chips einen RSSI-Wert für bis zu vier einzelne Antennen. Der aggregierte Wert wird normalerweise in dBm-Einheiten dargestellt.

Remote Control and Telemetry Daemon

ORCA ermöglicht lediglich die lokale Steuerung und Überwachung aus dem Userspace. COTS-Hardware verfügt jedoch in der Regel nur über begrenzte Rechen- und Speicherressourcen und bietet oft keine Unterstützung für praktische Skriptsprachen wie Python. Um diese Einschränkungen zu überwinden, führen wir den RCD ein, der die Fähigkeiten von ORCA über einen TCP-Socket zugänglich macht. Somit ermöglicht der Einsatz des RCD in einem Netzwerk die Überwachung und Steuerung mehrere ORCA-fähiger Geräte von einem zentralisierten und leistungsfähigeren Knoten aus. Darüber hinaus haben wir eine ZSTD-basierte [3] Komprimierung zur kompakten Erfassung von Überwachungsinformationen implementiert.

Eine detailliertere Beschreibung unserer WiFi-Ressourcenallokations-Forschungs-Toolchain finden Sie in unserem wissenschaftlichen Papier [ORCA-WiNTECH]. Diese Entwicklung wurde zusammen mit NMN und HHI durchgeführt.

3.2 Implementierung der lokalen Algorithmen zur Ressourcenallokation

Die in Abschnitt 3.1 besprochene ORCA-API stellt Informationen im Userspace über Datenleitungen bereit. Bei direkter Verwendung dieser API müssen alle Anwendungen im Userspace die Verbindung zu ORCA unter Verwendung des RCD verwalten und die generischen Datenleitungen analysieren. Daher haben wir ein Python-Paket namens **Rate Manager (RateMan)** entwickelt, das als Vermittler zwischen der ORCA und den Python-RC-Algorithmen fungiert.

RateMan-Schnittstellen

RateMan bietet eine robuste höhere Abstraktion und verbirgt die Feinheiten der ORCA-API vor Anwendungen in Python. Während RCD einen Fernzugriff auf die ORCA-Funktionen eines einzelnen APs bietet, ermöglicht RateMan den Nutzern die gleichzeitige Durchführung von RC über mehrere APs auf einer Basis pro Station. Durch die benutzerfreundlichen Funktionen müssen sich die Benutzer nicht mit den zugrunde liegenden Linux-Modulen und -Subsystemen auseinandersetzen. Abb. 2 zeigt eine Beispielkonfiguration, die die Verwendung von RateMan veranschaulicht. Ein Controller führt RateMan aus und verbindet sich mit Instanzen von RCD, die auf AP1 und AP2 laufen. Die APs bedienen jeweils zwei angeschlossene STAs.

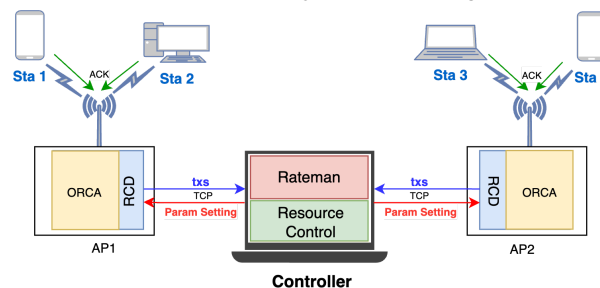


Abbildung 2: Ferngesteuertes RC mit RateMan und zugrundeliegende ORCA mit RCD.

Die erforderliche Parallelität von RateMan wird durch die Pythonbibliothek `asyncio` erreicht, die auf die Entwicklung von Single-Thread-Anwendungen zugeschnitten ist, die viele gleichzeitige blockierende I/O-Operationen verarbeiten. Für jede STA speichert RateMan die Rahmenübertragungsdaten für jede verfügbare Raten-Leistungs-Kombination. Da RateMan seine Statistiken kontinuierlich aktualisiert, kann jeder RA-Algorithmus seine eigene Ratenanpassungsperiode definieren. Entwickler können ihren eigenen RA-Algorithmus als Python-Paket implementieren und dabei die von RateMan bereitgestellten Steuerungs- und Überwachungsschnittstellen nutzen. Es ist auch möglich, einen RA-Algorithmus pro STA zu definieren.

Obwohl es möglich ist, RateMan weiter entfernt vom lokalen Netz zu positionieren, z. B. in einer Cloud-basierten Einrichtung, erfordern RA-Schemata eine zeitnahe Reaktion. Bei der Verwendung von RateMan für die Überwachung und Datenerfassung könnte jedoch ein Cloud-Einsatz von Vorteil sein.

Ressourcenallokation im User Space in Python

Unter Verwendung von ORCA, RCD und RateMan haben wir die folgenden Python-Beispielpakete zur Durchführung von RA implementiert. Der Zweck dieser Pakete besteht hauptsächlich darin, die Fähigkeiten von ORCA zu testen und zu demonstrieren. Sie sind ein Ausgangspunkt für die Entwicklung von anspruchsvolleren RC-Algorithmen und RC-Techniken durch gemeinsame Parameteroptimierung in der Zukunft.

Manual MRR Setter

Der Manual MRR Chain Setter (MMRRS) ist ein Paket, das eine rudimentäre Rateneinstellung für ein festes Intervall vornimmt. Es stellt einfach eine feste MRR-Kette für ein bestimmtes Zeitintervall ein. Diese einfache Rateneinstellung kann verwendet werden, um eine beliebige Rate aus der Liste der unterstützten Raten auf einer Rotationsbasis einzustellen, um die schnellste oder niedrigste Rate einzustellen oder um eine zufällig ausgewählte Rate einzustellen. Das MMRRS kann verwendet werden, um den drahtlosen Kanal zu bewerten und Übertragungsmuster zu erkennen.

Py-Minstrel-HT

Das Python-basierte Minstrel-HT (Py-Minstrel-HT) [12] ist ein Python-Paket, das als Userspace-Version des Standard-RA-Algorithmus des Linux-Kernels (Minstrel-HT) implementiert ist. Für jede angeschlossene STA beginnt der Algorithmus mit der niedrigsten MCS-Rate, die von der Verbindung unterstützt wird, und sendet in regelmäßigen Abständen eine Probe-Rate, um schnellere oder robustere Raten zu ermitteln. Das Standard-Abtastintervall beträgt 20 ms. Minstrel-HT verfolgt die Statistiken für jede Rate, indem es eine eindeutige Statistiktafel führt, die alle 50 ms aktualisiert wird. Der gewichtete Durchschnitt der Übertragungserfolgswahrscheinlichkeit pro Rate wird dann verwendet, um die Raten nach dem erwarteten Durchsatz einzustufen. Die MRR-Kette wird dann mit der höchsten geschätzten Durchsatzrate für die ersten drei Stufen und der besten Wahrscheinlichkeitsrate für die vierte Stufe bestückt. Wir haben Py-Minstrel-HT auf der Grundlage der neuesten OpenWrt-Version des Kernels Minstrel-HT implementiert. Py-Minstrel-HT bietet jedoch eine breite Palette von Einstellungen, die leicht konfiguriert werden können, z. B. Aktualisierungs- und Abtastintervall, Abtastreihenfolge und Vorladen. Dies ermöglicht eine adaptive Anpassung des Aktualisierungsintervalls je nach Szenario und Kanalbedingungen. Außerdem lassen sich die Auswirkungen von Änderungen an RC viel leichter beobachten als bei der In-Kernel-Implementierung, bei der der Kernel neu kompiliert und nach jeder Änderung neue Bilder geflasht werden müssten. Die User-Space-Implementierung wurde mit dem Kernel Minstrel-HT in identischer Umgebung validiert.

3.3 Implementierung der SDN-Architektur (Control / Dataplane Split)

NDH unterstützte HHI und BISDN bei der Untersuchung der Entwicklung der SDN-Architektur.

3.4 Implementierung der SupraCoNeX Analytics Engine

Wir haben das Python-Paket *scnx_expositor* entwickelt, um den Experimentierprozess zu optimieren. Dieses Paket nutzt die auf ORCA basierende RA-Toolchain und sammelt und organisiert effizient alle Spuren und Ordner. Diese Organisation erleichtert die Datenanalyse und stellt die Reproduzierbarkeit der Ergebnisse sicher.

Unter Verwendung der von *scnx_expositor* festgelegten Datei- und Ordnerbenennungskonventionen haben wir auch Auswertungsskripte erstellt. Diese Skripte generieren Visualisierungen, die eine detaillierte Analyse und einen Vergleich der Algorithmusleistung ermöglichen.

3.5 Validierung, Evaluierung und System-Performanzanalyse der SupraCoNeX Komponenten

Herausforderungen bei der Verwendung des mt76-Treibers

Während der Validierung von ORCA mit Chipsätzen, die vom mt76-Treiber unterstützt werden, sind wir auf Probleme bei der Konfiguration von RA-Parametern mit unseren User-Space-Algorithmen gestoßen. Wir haben erhebliche Ressourcen und Zeit in die Fehlersuche, Reproduktion und Verifizierung dieser Probleme investiert. Insbesondere war es bisher undokumentiert, dass die Chipsatz-Firmware in bestimmten Fällen Entscheidungen

zur Parameterkonfiguration außer Kraft setzen kann, selbst wenn der Hybridmodus aktiviert ist. Durch die intensive Zusammenarbeit mit unseren Konsortialpartnern - insbesondere NMN, NDH und HHI - konnten wir diese Fehler erfolgreich beheben und ORCA Funktionen hinzufügen, um die mit dem mt76-Treiber verbundenen Besonderheiten zu verwalten. Diese Verbesserungen und ihre Details werden in unserer Veröffentlichung näher erläutert.

AP 4 – Experimentalumfeld und Demonstrator

4.1 Spezifikation der Testszenarien

Zur Demonstration und Bewertung der in AP 3 erörterten Techniken haben wir mit alle Partnern zusammenverschiedene Szenarien betrachtet, die jeweils einem bestimmten Bewertungszweck dienen. Im Folgenden sind die ausgewählten Szenarien aufgeführt

Szenario 1: Einfaches Heim-WiFi

In diesem Szenario betrachten wir die Over-the-Air-Kommunikation innerhalb eines einfachen Heim-WiFi. Konkret werden ein oder mehrere APs mit mehreren STAs betrachtet, die ein OBSS bilden. Der Controller kann ein Laptop oder ein PC mit Rechenkapazitäten sein, um die RA-Aufgabe über mehrere APs zu bewältigen.

Für dieses Szenario wird der Verkehrsfluss zwischen den AP-STA-Verbindungen mit Tools wie iperf3 künstlich erzeugt. Dies ermöglicht eine experimentelle Kontrolle über die Verkehrsparameter sowie die Möglichkeit der Reproduzierbarkeit. Die WiFi-Knoten in diesem Szenario sind Interferenzen von bestehenden, sich überschneidenden BSSs ausgesetzt. Dies gibt Aufschluss über realistische Kanalbedingungen und Anforderungen an das Interferenzmanagement.

Szenario 2: Isoliertes Single-Link-WiFi-Netzwerk

Eine isolierte RF-Verbindung zwischen einem AP und einem STA wird als kontrollierbares Netzwerk betrachtet. In einem solchen Szenario werden die Auswirkungen von Interferenzen durch benachbarte Knoten und sich überschneidende BSSs eliminiert. Dies ist nützlich, um die Leistung verschiedener RA-Algorithmen auf kontrollierte Weise genau zu bewerten. Unterschiedliche Kanalbedingungen können durch den Einsatz eines Dämpfungsglieds emuliert werden, das eine programmierbare Einstellung der Dämpfungspegel über die HF-Verbindungen zwischen dem AP und der STA ermöglicht. Der Verkehrsfluss wird wie in Szenario 1 mit Tools wie iperf3 künstlich erzeugt.

Szenario 3: Produktives Netzwerk innerhalb von Evernet

Die Evernet eG [6] ist ein genossenschaftlicher Internet Service Provider und betreibt ein produktives Breitband-WLAN-Netz im ländlichen Raum in fünf Gemeinden Thüringens. Sie verfügt über 100 OpenWRT Linux WiFi Mesh Router im 5GHz ISM Spektrum und ca. 420 WiFi Access Points im 2,4GHz ISM Spektrum, insgesamt sind 108 Haushalte und 8 Unternehmen an das drahtlose Kommunikationsnetz angeschlossen.

In diesem Szenario wird eine Reihe von APs zur Bewertung unserer RC-Algorithmen herangezogen. In diesem Fall stellt der Verkehrsfluss die tatsächlichen Kommunikationsdaten zwischen dem AP und den STAs dar. Es werden Benutzer mit unterschiedlichen Mobilitäts- und Datenanforderungen erwartet. Bei diesem Szenario handelt es sich im Wesentlichen um ein echtes WiFi-Netz mit sich überlappenden BSSs im Innen- und Außenbereich. Da der Verkehrsfluss nicht künstlich erzeugt wird, ist die Reproduzierbarkeit der Versuchsergebnisse nicht gewährleistet.

4.2 Deployment des SupraCoNeX Systems

Um reproduzierbare Experimente für verschiedene Testszenarien durchzuführen, haben wir zusammen mit den anderen Partnern die folgenden Testbed-Bereitstellungen entwickelt.

Setup 1: Schreibtisch Setup

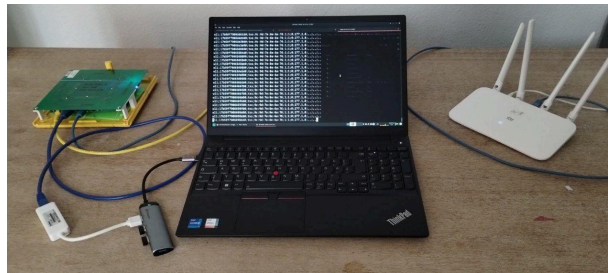


Abbildung 3: Schreibtisch Setup

Die typische Schreibtischkonfiguration für den Einsatz umfasst eine AP-STA-Konfiguration und einen Laptop oder PC, der als Controller dient. Abb. 3 zeigt eine Grundkonfiguration mit einem AP, einem STA und einem Laptop, der als Controller fungiert.

WiFi-Netzwerkgeräte, auf denen das OpenWrt-Betriebssystem läuft, können als APs verwendet werden. Diese Geräte verwenden häufig WiFi-Chipsätze, die auf den Standards IEEE 802.11n und ac basieren. Insbesondere die Atheros AR9280 (IEEE 802.11n) und Mediatek MT7615 (IEEE 802.11n/ac) Chipsätze, die von den ath9k bzw. mt76 Treibern unterstützt werden, werden häufig verwendet. STAs können alle WiFi-fähigen Geräte sein, die mit den IEEE 802.11n/ac-Standards kompatibel sind. Der Controller, bei dem es sich um einen Desktop-Computer, einen Laptop oder einen dedizierten Server handeln kann, muss über eine ausreichend leistungsfähige CPU verfügen, um die Algorithmen im Benutzerbereich ohne Latenz zu verarbeiten. Vorzugsweise wird eine geeignete Linux-Distribution als Controller-Betriebssystem verwendet, um Netzwerk-Namespaces zu nutzen. Die Verkehrserzeugung wird mit dem iperf3-Tool zur Messung der Netzwerkleistung verwaltet.

Linux-Netzwerk-Namespaces ermöglichen das Routing des Datenverkehrs, so dass Quelle und Senke über den Controller realisiert werden, während die drahtlose Übertragung zwischen AP und STA erfolgt. Dieser Ansatz entlastet die CPU bei der Erzeugung von Datenverkehr auf dem AP.

Setup 2: Single-Link Shielding Box Setup

Um ein isoliertes Single-Link-WiFi-Netzwerk einzurichten, verwenden wir einen Aufbau, bei dem die WiFi-Knoten in RF-Abschirmkästen eingeschlossen sind. Die Knoten sind mit HF-Kabeln verbunden, zwischen denen sich ein Dämpfungsglied befindet. Abb. 4 zeigt den experimentellen Testaufbau, bestehend aus einem AP und einem STA, die ein Single-Link-WiFi-Netzwerk bilden. Beide basieren auf dem Banana Pi R64 WiFi Router Board [8], das einen MediaTek MT7622 System-on-Chip mit einem 64-bit Dual-Core ARM Cortex-A53 Prozessor verwendet und mit MT7615 802.11ac Chipsätzen ausgestattet ist.

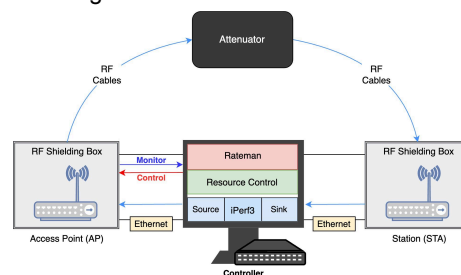


Abbildung 4: Komponentenansicht des RF-Abschirmkasten Experimentaufbau mit einem programmierbaren Dämpfungsglied

Die WiFi-Geräte befinden sich in separaten EMC Pioneer PESB-08 [11] HF-Abschirmgehäusen, um externe Störungen zu verhindern und eine reproduzierbare Leistungsbewertung zu ermöglichen. Diese Abschirmungsboxen blockieren HF-Signale im Frequenzbereich von 0,8 GHz bis 6 GHz und bieten eine Abschirmung von mindestens 70 dB bei 5,8 GHz. Die Antennenanschlüsse von AP und STA sind über HF-Kabel mit einem dazwischen liegenden Dämpfungsglied verbunden. Wir verwenden das programmierbare 4-Kanal-Dämpfungsglied AD-USB4AR36G95 von Adaura (5-6000 MHz) [1]. Die Einstellung der Dämpfungspegel in dB ermöglicht es uns, verschiedene SNR-Pegel des Funkkanals auf kontrollierbare Weise zu emulieren. Als

Controller verwenden wir einen Workstation-PC mit einer Intel Xeon W3530 CPU mit 8 Kernen und einem Basistakt von 2,80 GHz, 12 GB DDR3 RAM und Arch Linux. Der Controller ist für die Erzeugung von künstlichem Datenverkehr mit iperf3 und die Ausführung von Überwachungs- und Steuerungsfunktionen über RateMan zuständig. Er ist über 1-Gbit/s-Ethernet-Links mit dem AP und dem STA verbunden.

Wir nutzen Linux-Netzwerk-Namespaces, um die Netzwerkschnittstellen zu trennen, die für die Verbindungen vom Controller zum AP und STA verwendet werden. Dieses Setup leitet den künstlich erzeugten Datenverkehr über Ethernet an den AP, überträgt ihn dann drahtlos vom AP zum STA und leitet schließlich den empfangenen Datenverkehr über die Ethernet-Verbindung zwischen dem STA und dem Controller zurück an den Controller.

Setup 3: Evernet

Die Evernet e.G. betreibt ein Netzwerk, das fünf Dörfer abdeckt und über fünf aggregierte VDSL-Uplinks rund 100 Haushalten drahtlosen Internetzugang bietet. Der Backhaul des Datenverkehrs erfolgt über 802.11n- und 802.11ac-Mesh-Links sowie 5-GHz- und 60-GHz-Punkt-zu-Punkt-Backbone-Links. Der auf 1G-Ethernet basierende Intra-Household-Link bietet allen angeschlossenen APs im Haushalt Internetzugang. Innerhalb eines BSS ist ein AP über WLAN mit mehreren STAs verbunden. Der Controller ist über eine 10-G-Glasfaserverbindung mit dem Mesh-Gateway verbunden. Über diese Verbindung kann der Controller die APs in jedem Haushalt überwachen und steuern.

4.3 Evaluation und Performance-Analyse des SupraCoNeX Systems

Wir haben unsere ORCA-basierte Toolchain erfolgreich im Evernet-Netzwerk bereitgestellt. Frühe Experimente mit Py-Minstrel-HT zeigten die Wirksamkeit der Verwendung von User Space-Algorithmen zur RA. Aufgrund von Verzögerungen bei der Behebung aller Fehler und Probleme mit dem mt76-Treiber konnten wir jedoch keine umfassenden Experimente und Bewertungen von RA mit MAB-basierten Algorithmen durchführen.

4.4 Aufbau und Umsetzung der Demonstratoren

Der geplante mobile Demonstrator wurde durch Setup 1 realisiert. Dieses Setup, bestehend aus COTS-WiFi-Knoten und typischerweise einem Laptop als Controller, wurde verwendet, um unsere Arbeit auf verschiedenen Veranstaltungen zu präsentieren, unter anderem beim Freifunk Wireless Community Weekend 2022 in Weimar, Deutschland und beim Wireless Battle Mesh 2023 in Barcelona, Spanien.

AP 5 – Dissemination - Verbreitung der Ergebnisse und Dokumentation

Nachdem wir wichtige Meilensteine erreicht hatten, ergriffen wir zahlreiche Maßnahmen, um die Ergebnisse unserer Arbeit zu verbreiten. Im Januar 2024 veranstalteten wir eine öffentliche Veröffentlichungsveranstaltung, bei der die ORCA-Toolchain offiziell veröffentlicht wurde. Die Veranstaltung brachte erfolgreich mehrere namhafte Mitglieder der OpenWrt- und Freifunk-Communitys zusammen. Sie trug dazu bei, Interesse und Bekanntheit für dieses Projekt zu wecken. Es gab eine lebhafte Diskussion über zukünftige Möglichkeiten und Richtungen zur Implementierung neuartiger Algorithmen für WiFi RA.

Derzeit haben wir zwei Artikel eingereicht. Der erste ist ein technischer Artikel, in dem ORCA, RCD und RateMan beschrieben werden. Dieser Artikel soll einen umfassenden Überblick über die ORCA-basierte WiFi RA-Toolchain bieten und zeigen, wie Forscher diese zur Implementierung ihrer eigenen RA-Algorithmen nutzen können. Dieser Artikel wurde bei WiNTECH 2024 Konferenz eingereicht. Der zweite Artikel ist ein Forschungsartikel über den Einsatz von Multi-Armed Bandits für die Auswahl von Senderaten im WLAN. Hierbei wurden die Tools ORCA, RCD und RateMan für die Implementierung der Algorithmen auf realer WLAN-Hardware genutzt. Dieser Artikel wurde bei der IEEE CQR 2024 Konferenz eingereicht.

3. Voraussichtlicher Nutzen und Verwertbarkeit des Ergebnisses

3.1 Diskussion über den erwarteten Nutzen des Projekts und die Verwertbarkeit der Ergebnisse

Das im Rahmen des Projekts entwickelte System bietet eine Schnittstelle, die es Anwendern ermöglicht, die Kontrolle über WLAN-Ressourcen wie Modulations- und Kanalcodierungsparametern, Anzahl der genutzten Antennen sowie die Sendeleistung zu übernehmen. Dies war bisher nur in Simulationen oder mit tiefem Wissen in der Kernel-C-Programmiersprache möglich, die in der WLAN-Firmware und -Treiber verwendet wird. Jetzt jedoch können diese Steuerungen durch die Verwendung der weitverbreiteten und benutzerfreundlichen Programmiersprache Python realisiert werden. Diese Schnittstelle hat das Potenzial, die Akzeptanz und Weiterentwicklung von WLAN-Kontroll- und Optimierungsalgorithmen erheblich zu erleichtern und zu erweitern, da sie leicht auch auf andere Programmiersprachen angepasst werden kann.

3.2 Konkrete Planungen für die nächste Zukunft, insbesondere im Rahmen des fortgeschriebenen Verwertungsplans

In der nächsten Phase des Projekts stehen zwei Hauptziele im Vordergrund:

Ausbau und Erweiterung des WLAN-Shielding Box Testbed Setup: Durchführung von Experimenten unter realistischeren Bedingungen in einer kontrollierten Umgebung.

Diese Erweiterung wird die Untersuchung der Leistungsfähigkeit von WLAN-Algorithmen unterstützen, indem sie eine präzisere Kontrolle und Analyse der Beziehung zwischen einstellbaren Parametern und der Netzwerkperformance ermöglicht. Dadurch wird es möglich, fundierte Erkenntnisse zu gewinnen, die zur Entwicklung und Feinabstimmung neuer, leistungsfähigerer Algorithmen genutzt werden können.

Erweiterung von ORCA für die Nutzung mit IEEE 802.11ax Chipsets: Um die Nutzung moderner COTS-Hardware zu ermöglichen, die mit IEEE 802.11ax mit der ORCA-basierten WiFi RA-Toolchain kompatibel ist.

Die Live-Einsätze der MAB-Algorithmen bieten eine wichtige Möglichkeit, die theoretischen und experimentellen Ergebnisse unter realen WLAN-Bedingungen zu bestätigen. Dies ist entscheidend, um die praktische Anwendbarkeit und Effizienz der Algorithmen zu demonstrieren, was letztlich die Grundlage für eine breitere Implementierung in der Praxis darstellen kann.

4. Zusammenfassung und Ausblick

Im Rahmen des SupraCoNeX-Projekts haben wir eine umfassende und wirklich quelloffene Toolchain für die Forschung zur WiFi-RA entwickelt. Diese Toolchain umfasst die ORCA Linux-Kernel-Benutzer-API, den Fernsteuerungs- und Telemetrie-Daemon sowie das RateMan-Python-Paket. Diese Tools unterstützen gemeinsam fortgeschrittene Experimente und Forschungen zur WiFi-RA.

Darüber hinaus haben wir eine praktische Modellierung der WiFi-RA als MAB-Problem durchgeführt. Dies umfasste die Implementierung von auf Thompson Sampling basierenden Ratenanpassungsalgorithmen sowie die Entwicklung einer User-Space-Version des Minstrel-HT-Algorithmus. Unsere Forschung konzentrierte sich auf praktische Experimente mit COTS-Hardware, um sicherzustellen, dass unsere Erkenntnisse und Tools relevant und auf reale Szenarien anwendbar sind. Diese Beiträge bieten eine solide Grundlage für die zukünftige Implementierung und Nutzung von auf maschinellem Lernen basierenden Algorithmen und Techniken zur Lösung des RAproblems in WiFi-Netzwerken.

Mit Blick auf die Zukunft wird sich unsere zukünftige Arbeit auf die Erweiterung der Forschungs-Toolchain konzentrieren, um Experimente mit WiFi-Netzwerken zu unterstützen, die mit dem IEEE 802.11ax-Standard kompatibel sind. Da es derzeit keine Open-Source-Treiber für diesen Standard gibt, erwägen wir Partnerschaften mit Branchenteilnehmern, insbesondere mit Herstellern von WLAN-Chipsätzen. Angesichts der zunehmenden Dichte und Komplexität moderner drahtloser Netzwerke sind die Beherrschung von Interferenzen und die

Verbesserung der Energieeffizienz entscheidende Bereiche für unsere künftige Forschung. Wir gehen davon aus, dass unsere Erfahrung bei der Implementierung von Algorithmen auf Basis maschinellen Lernens für echte WLAN-Netzwerke bei der Bewältigung dieser Herausforderungen von unschätzbarem Wert sein wird.

5. Veröffentlichungen des Ergebnisses

Liste der eingereichten wissenschaftlichen Arbeiten

- [ORCA-WiNTECH] Pawar et al., “Open-source Resource Control API for real IEEE 802.11 Networks”, auf Konf. WiNTECH 2024, Washington, D.C, USA.
- [MAB-TS-CQR] Le et al., “Multi-Armed Bandits in IEEE 802.11ac: Efficient Algorithms and Testbed Experiments”, auf ‘IEEE International Communications Quality and Reliability Workshop’ (IEEE CQR 2024), Seattle, WA, USA.

Liste der geplanten Veröffentlichungen

- Performance Analysis of Minstrel-HT Variants for COTS WiFi Networks
- Optimizing WiFi Rate Adaptation through Multi-Armed Bandits and Exploitation of Multi-Rate Retry Chains

6. Wichtige Positionen des zahlenmäßigen Nachweises

1. Projektbudget und -kosten: Es gibt keine Abweichungen zur ursprünglichen Kostenplanung.
2. Zeitplan und Termine:
 - a. Geplante Projektdauer: 01.08.2020 - 31.07.2023
 - b. Tatsächliche Projektdauer: 01.08.2020 - 31.01.2024
 - c. Geplante Meilensteine: 6
 - d. Tatsächlich erreichte Meilensteine: 6
3. Abweichungsanalyse:
 - a. Einsparung bei Sachkosten und Reisekosten wegen Änderungen der Hardwareanforderungen und Corona.
 - b. Längere Projektdauer wegen Corona.

7. Literaturverzeichnis

- [1] Adura Technologies. 4-Channel 5-6000MHz Programmable RF Attenuator (AD-USB4AR36G95) Specifications. <https://adauratech.com/product/ad-usb4ar36g95/>.
- [2] R. Brown. Openwrt project. <https://openwrt.org/>, Jun 2023.
- [3] Y. Collet and M. Kucherawy. Zstandard compression and the application/zstd media type. Technical report, Facebook, 2018.
- [4] O. contributors. mt76: mac80211 driver for MediaTek MT76x0e, MT76x2e, MT7603, MT7615, MT7628 and MT7688. github.com/openwrt/mt76.
- [5] MediaTek Incorporated. MediaTek MT7615 WiFi Chip Specifications. <https://www.mediatek.com/products/broadband-wifi/mt7615>.
- [6] Evernet eG. Evernet eG. <http://evernet-eg.de/>
- [7] F. Fietkau. minstrel-ht: new rate control module for 802.11n. lwn.net/Articles/376765/, 2010.
- [8] Guangdong Bipai Technology and Taiwan Hon Hai Technology (Foxconn). BananaPi Banana Pi BPI-R64 Router Board Specifications. <https://www.banana-pi.org/en/bananapi-router/28.html>.
- [9] W. Yin, P. Hu, J. Indulska, M. Portmann, and Y. Mao. Mac-layer rate control for 802.11 networks: A survey. *Wireless Networks*, 26(5):3793–3830, 2020.
- [10] Linux Wireless. ath9k wireless driver. <https://wireless.wiki.kernel.org/en/users/drivers/ath9k>, October 2023.
- [11] PIONEER EMC LTD. EMC Pioneer PESB-06 RF Shielding Box Product Page. <https://www.emcpioneer.com/RF-Box-pd43750874.html>.
- [12] Prashiddha Dhoj Thapa. Evaluation of Resource Allocation in WiFi Networks using Minstrel HT in User space. Bachelor's Thesis, Jacobs University Bremen. 2022.
- [13] S. Szott, K. Kosek-Szott, P. Gawlowicz, J. T. Gómez, B. Bellalta, A. Zubow, and F. Dressler. Wi-Fi Meets ML: A Survey on Improving IEEE 802.11 Performance with Machine Learning. *IEEE Communications Surveys & Tutorials*, 24(3):1843–1893, 2022.
- [14] F. Vergès. Mcs index - MCS index table, modulation and coding scheme index 11n, 11ac, and 11ax. <https://mcsindex.com/>.