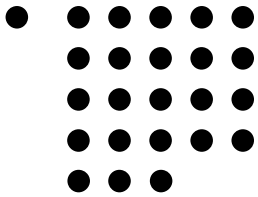


Fachhochschule Köln
University of Applied Sciences Cologne

Fakultät 07
Informations-, Medien- und Elektrotechnik

Institut für Nachrichtentechnik
Labor für Datennetze

Studiengang
Bachelor of Science in Information Engineering

Bachelor-Thesis

Analyse des WMM-Verfahrens zur Unterstützung von QoS-Mechanismen im WLAN

Student: Christoph Michael Obst

Referent: Prof. Dr. Andreas Grebe

Korreferent: Dipl.-Ing. Achim Marikar

Abgabedatum: 30. Juni 2006

Hiermit versichere ich, dass ich die Bachelor-Thesis selbstständig angefertigt und keine anderen als die angegebenen und bei Zitaten kenntlich gemachten Quellen und Hilfsmittel benutzt habe.

_____

Christoph Michael Obst

Inhaltsverzeichnis

Abbildungsverzeichnis 5

Tabellenverzeichnis 5

1 Einleitung 6

2 Grundlagen 7

2.1 Wireless LAN nach IEEE 802.11 7

2.1.1 Rückblick 7

2.1.2 Inhalte des 802.11-Standards 8

2.1.3 MAC-Layer nach 802.11 9

2.2 Quality of Service 18

2.2.1 Notwendigkeit von QoS 18

2.2.2 QoS auf OSI-Schicht 2 19

2.2.3 QoS auf OSI-Schicht 3 20

2.2.4 Umsetzungsproblem bei Shared Media 21

2.3 QoS im WLAN mittels Wireless Multimedia (WMM) 22

2.3.1 Entstehung von WMM 22

2.3.2 Arbeitsweise von WMM 22

3 Vorarbeiten 29

4 Aufbau einer Testumgebung 30

4.1 Anforderungen und Festlegungen 30

4.2 Marktrecherche 32

4.3 Installation und Konfiguration 33

4.3.1 WRT54G 34

4.3.2 Allnet-PCI-WLAN-Karten ALL0281(A) mit MadWifi-Treiber 34

4.4 Gesamtübersicht der Testumgebung 36

5 Test und Analyse der WMM-Funktionalität 38

5.1 Software für Messungen und Analyse 38

5.1.1 Ethereal 38

5.1.2 Ping 38

5.1.3 Iperf..... 39
5.1.4 Shell-Skripte..... 42
5.2 Messungen..... 43
5.2.1 Mapping von TOS-Werten zu WMM-Kategorien..... 45
5.2.2 Durchsatzmessungen ohne Konkurrenzsituation..... 49
5.2.3 Durchsatzmessungen mit Konkurrenzsituation..... 56
5.2.4 Client-interne Priorisierung..... 63
5.2.5 Weitere Beobachtungen und aufgetretene Probleme..... 70
6 Schlussbetrachtung..... 72
Anhang A: Shell-Skripte..... 73
Anhang B: MadWifi-Codeausschnitt..... 79
Anhang C: Messdiagramme..... 82
Anhang D..... 88
Abkürzungsverzeichnis..... 91
Quellenverzeichnis..... 94

Abbildungsverzeichnis

Abbildung 1: Ablauf einer Übertragung (entnommen aus [3])..... 12
Abbildung 2: Backoff-Mechanismus (entnommen aus [3])..... 14
Abbildung 3: Hidden-Node-Problem..... 15
Abbildung 4: WMM AC Timing (z.T. entnommen aus [7])..... 25
Abbildung 5: AC-Warteschlangen (z.T. entnommen aus [7])..... 26
Abbildung 6: interner Aufbau des Linksys WRT54G..... 30
Abbildung 7: Übersicht der Testumgebung..... 37
Abbildung 8: Kanalbelegung von WLAN im 2,4 GHz-Band in Deutschland..... 44
Abbildung 9: Datenpaket mit WMM-Markierung..... 49
Abbildung 10: Diagramm 4b..... 52
Abbildung 11: Diagramm 2b..... 54
Abbildung 12: Diagramm 1a..... 55
Abbildung 13: Diagramm 10b..... 58
Abbildung 14: Diagramm 11b..... 59
Abbildung 15: Diagramm 11c..... 61
Abbildung 16: Diagramm 11d..... 63
Abbildung 17: Diagramm 21b..... 65
Abbildung 18: Aufbau Client-interne Priorisierung in MadWifi..... 68
Abbildung 19: Ethereal-Mitschnitt Client-interne Priorisierung in MadWifi..... 69

Tabellenverzeichnis

Tabelle 1: Übersicht IEEE 802.11-Standards..... 7
Tabelle 2: WMM-Kategorien..... 24
Tabelle 3: EDCA-Werte..... 25
Tabelle 4: Kanalbelegung von WLAN im 2,4 GHz-Band in Deutschland [1]..... 44
Tabelle 5: TOS-WMM-Mapping..... 47

1 Einleitung

Sowohl im privaten als auch im geschäftlichen Umfeld kommen zur Vernetzung von Computern und anderen Geräten immer häufiger Wireless LANs (kurz WLAN) zum Einsatz. Diese ersparen die oftmals teure und schwierige Verkabelung. Außerdem ermöglichen sie den schnurlosen und mobilen Betrieb diverser Geräte. Per WLAN kann man mit dem Laptop auf das Internet zugreifen, den PDA-Terminkalender abgleichen oder neuerdings mit WLAN-Telefonen Voice over IP (kurz VoIP) nutzen. Des Weiteren sind auch Audio- und Video-Streaming-Clients zu nennen, die Musik und Videos drahtlos aus dem Internet empfangen und auf Fernseher oder Stereoanlage ausgeben können.

Gerade bei Multimedia-Daten wie Musik oder Video ist es aber von besonderem Interesse des Nutzers, dass diese ohne Unterbrechung am Zielgerät ankommen und somit ein „fließender“ Betrieb möglich ist. Bei Videos sind Unterbrechungen in Form von „stehenden“ Bildern bis zu einem gewissen Maße noch tolerierbar. Bei Ton, Musik oder Telefonaten hingegen sind auch seltene Aussetzer absolut störend und fallen dem Nutzer meist sofort auf.

Deshalb ist es nötig, auch im WLAN Möglichkeiten zur Qualitätssicherung bereitzustellen. Nur so können Multimedia-Daten gegenüber sonstigen Daten bevorzugt behandelt und über die Luftschnittstelle übertragen werden.

Im Rahmen dieser Bachelor-Thesis mit dem Titel „Analyse des WMM-Verfahrens zur Unterstützung von QoS-Mechanismen im WLAN“ soll untersucht werden, wie das Wireless Multimedia-Verfahren (kurz WMM) arbeitet, welche Möglichkeiten es bietet und ob es die gewünschte Qualitätssicherung (in der Praxis) erfüllen kann. WMM ist eine Teil-Erweiterung des IEEE 802.11-Standards. Deshalb werden in dieser Bachelor-Thesis ausschließlich WLANs nach diesem Standard betrachtet.

Bei Lesern wird Grundlagenwissen im Fachbereich Datennetze vorausgesetzt.

2 Grundlagen

In diesem Kapitel wird auf wichtige Grundlagen eingegangen, die für die Bearbeitung des Themas von Belang sind. Dazu gehört der WLAN-Standard 802.11, Quality of Service im Allgemeinen und im Speziellen im Rahmen von WMM.

2.1 Wireless LAN nach IEEE 802.11

Bei 802.11 handelt es sich um eine Gruppe von Standards für WLANs, die durch das Institute of Electrical and Electronics Engineers (kurz IEEE) entwickelt wurden und werden.

2.1.1 Rückblick

Der ursprüngliche Standard 802.11 wurde ab 1991 von der IEEE entwickelt und 1997 verabschiedet. Er erlaubte Datenübertragungen mit ein und zwei MBit/s. Nur wenige Monate später begannen bereits die Arbeiten an Erweiterungen und Optimierungen des Grundstandards. Im Jahr 1999 wurden dann die Erweiterungen 802.11a und 802.11b verabschiedet, wobei 2001 eine Korrektur von 802.11b folgte. Mit dem a-Standard sind seitdem Datenraten von bis zu 54 MBit/s möglich. Die b-Erweiterung erlaubt lediglich bis zu 11 MBit/s. Dass in Deutschland trotzdem hauptsächlich 802.11b zum Einsatz kam, liegt daran, dass es wie der Grundstandard im ISM-Band[1] bei 2,4 GHz betrieben wird. Dieses Frequenzband darf weltweit ohne Lizenzen oder Genehmigungen genutzt werden. In Deutschland gibt es nur eine Beschränkung der Sendeleistung auf 100mW. 802.11a nutzt dagegen Frequenzen im 5-GHz-Bereich, welche in Deutschland erst seit November 2002 zur Nutzung freigegeben sind, wenn bestimmte Auflagen erfüllt werden. Dafür wurde im September 2003 die Standarderweiterung 802.11h verabschiedet. Einige Monate früher wurde

Standard	*Fertig-stellung*	*Frequenz-band*	*Max. Datenraten*
802.11	1997	2,4 GHz	1 und 2 MBit/s
802.11a	1999	5 GHz	54 MBit/s
802.11b	1999	2,4 GHz	11 MBit/s
802.11g	2003	2,4 GHz	54 MBit/s
802.11e	2005	-	-

Tabelle 1: Übersicht IEEE 802.11-Standards

1 ISM – Abk. für Industrial, Scientific, Medical

aber bereits der Standard 802.11g fertiggestellt, der ebenfalls Datenraten von bis zu 54 MBit/s ermöglicht. Der Vorteil dieser Erweiterung und wohl auch der Grund der heutigen weiten Verbreitung von 802.11g-Geräten ist die Nutzung des 2,4-GHz-Bandes wie bei 802.11b.[2]

Zur Zeit sind bei der IEEE über 20 Erweiterungen des 802.11-Standards gelistet, an denen gearbeitet wird oder die bereits fertiggestellt wurden. Einige wurden bereits erwähnt. Ein weiterer ist 802.11e, mit dem unter anderem QoS-Mechanismen im WLAN bereitgestellt werden sollen. Teile von 802.11e, die in WMM eingeflossen und deshalb von besonderem Interesse in dieser Bachelor-Thesis sind, werden im Kapitel 2.3 näher erläutert.

2.1.2 Inhalte des 802.11-Standards

Im 802.11-Standard sind MAC- und PHY-Layer definiert. Der Physical Layer ist die unterste Schicht im OSI-Schichtenmodell und wird auch Bitübertragungsschicht genannt. Dort ist unter anderem definiert, wie das jeweilige Übertragungsmedium genutzt wird, wie physikalische Verbindungen auf- und abgebaut werden und wie einzelne Bits übertragen werden. Im Fall des Grundstandards von 802.11 werden Informationen entweder optisch[3] oder mit elektromagnetischen Wellen durch den Raum übertragen. Der MAC-Layer ist ein Sublayer der zweiten OSI-Schicht. Diese wird Data Link Layer oder Sicherungsschicht genannt. Sie steuert den Datenfluss und soll eine möglichst fehlerfreie Übertragung sicherstellen.

Für die speziellen Anforderungen in LANs hat die IEEE die zweite OSI-Schicht in die Sublayer MAC und LLC unterteilt. Der MAC-Layer regelt den Zugriff auf das Medium, da dieses im Falle von LANs gemeinsam genutzt wird. Konkurrierende Stationen können nicht zeitgleich darauf zugreifen, da sonst am Empfänger überlagerte und somit nicht mehr auswertbare Signale ankämen. Der LLC-Layer regelt die Sicherungsfunktionen der eigentlichen zweiten OSI-Schicht. Sowohl bei Ethernet als auch bei WLAN wird über dem MAC der gleiche LLC-Layer genutzt.[22]

2 [1], Kap. 1.2.2; [2]

3 802.11 sah im Grundstandard auch die Möglichkeit der optischen Übertragung vor.

2.1.3 MAC-Layer nach 802.11[4]

Im Folgenden wird genauer auf den MAC-Layer des 802.11-Standards eingegangen. Dieser definiert die zwei grundsätzlichen Zugriffsmethoden DCF und PCF. Bei der Distributed Coordination Function (DCF) handelt es sich um einen dezentralen Ansatz, der die Basismethode darstellt. Jede Station ist dabei selbst verantwortlich für den Medienzugriff. Bei der Point Coordination Function (PCF) hingegen gibt es einen so genannten Point Coordinator, der den Medienzugriff zentral verwaltet und die Stationen durch Polling zum Senden auffordert beziehungsweise die Sendeerlaubnis erteilt. PCF ist optional und baut auf DCF auf.

Der Zeitraum, während dem in einer Funkzelle nach der DCF gearbeitet wird, heißt „Contention Period“. Der Zeitraum mit PCF-Mechanismus wird „Contention Free Period“ genannt. Übersetzt bedeutet das, dass sich eine Phase mit Konkurrenzsituation abwechselt mit einer, in der es keinen Wettstreit um den Medienzugriff gibt.

CSMA/CA-Verfahren

Bei der Basiszugriffsmethode wird von den Stationen das CSMA/CA-Verfahren eingesetzt, welches starke Ähnlichkeiten zu jenem vom leitungsgebundenen Ethernet aufweist. Dort wird bekanntlich CSMA/CD verwendet. Beiden Verfahren gemeinsam ist das CSMA, also „Carrier Sense Multiple Access“. Wenn eine Station senden möchte, hört sie zunächst auf dem Übertragungsmedium, ob dieses frei ist. Erst wenn dies der Fall ist, wird darauf zugegriffen. „Multiple Access“ drückt den Sachverhalt aus, dass möglicherweise mehrere Stationen gleichzeitig zugreifen möchten. Da dies aber zu Kollisionen und somit fehlerhaften Übertragungen führen würde, kommt im Ethernet „Collision Detection“ zum Einsatz. Während des Sendevorganges hört die sendende Station gleichzeitig das Medium ab und kann damit feststellen, ob sie das empfängt, was sie auch gesendet hat. Ist dies nicht der Fall, muss von einer Kollision ausgegangen werden. Der Sendevorgang wird dann abgebrochen und nach erneutem Carrier Sense wiederholt. Im WLAN ist ein solcher Mechanismus nicht mit vertretbarem Aufwand umsetzbar. Eine Station würde unabhängig voneinander arbeitende Sende- und Empfangseinheiten benötigen um zeitgleich senden und abhören zu können. Deshalb nutzt man im WLAN „Collision

4 [1] Kap. 4.2; [3] Kap. 9.2

Avoidance“, also eine Kollisionsvermeidung. Wie diese funktioniert wird nachfolgend erklärt.

NAV-Wert

Im WLAN besteht ein Datenaustausch zwischen zwei Stationen immer aus mehreren Frames zwischen denen Sendepausen auftreten. Deshalb nutzt man im WLAN nicht ausschließlich die „Carrier Sense“-Funktion des Physical-Layers. Diese signalisiert dem MAC in einer Sendepause einer laufenden Operation anderer Stationen ein freies Medium. Daraufhin würde eine sendewillige Station die Operation durch ihre Aussendung unterbrechen. Deshalb ist im MAC-Layer eine virtuelle Medienabhörfunktion implementiert. Diese nutzt einen Timer, der als „Network Allocation Vector“ (NAV) bezeichnet wird. Jede Station verwaltet diesen NAV-Wert für sich selbst und weiß durch diesen, wie lange das Medium noch anderweitig belegt sein wird. Erst wenn der NAV-Wert auf Null zurückgezählt ist und der Physical-Layer ein freies Medium signalisiert, beginnt eine Station zu senden. Der NAV-Wert wird durch die Angabe im Duration/ID-Feld des Frame-Headers mitgeteilt (siehe 802.11-Frameformat, S.16). Eine sendende Station teilt dadurch allen anderen mit, wie lange der begonnene Datenaustausch inklusive aller Pausen und abschließendem Acknowledgement (ACK) andauern wird. Alle empfangenden Stationen lesen den Wert aus dem Header aus und aktualisieren ihren NAV-Wert, wenn dieser größer ist als der vorhandene. Damit in dem Moment, wenn der NAV-Wert in allen Stationen auf Null zurückgezählt ist, nicht alle gleichzeitig beginnen zu senden, gibt es zusätzlich noch den Backoff-Mechanismus. Wenn eine Übertragung abgeschlossen ist, wählt eine sendewillige Station eine zufällige Backoff-Wartezeit. Solange durch Carrier Sense das Medium als frei erkannt wird, wird die Backoff-Zeit heruntergezählt. Ist die Wartezeit abgelaufen, darf die entsprechende Station senden. Durch diese Verfahren werden Kollisionen zum großen Teil vermieden, ganz auszuschließen sind sie aber nicht.

Acknowledgement

Um am Sender erkennen zu können, ob Datenrahmen bei der Übertragung durch Kollisionen oder andere äußere Einflüsse zerstört wurden, werden im WLAN ACK-Frames eingesetzt. Jedes korrekt empfangene Frame, welches eine Bestätigung

erfordert, führt zur Aussendung eines ACKs durch eine bestimmte Station. Entweder ist dies der Access Point oder die empfangende Station. Dies gilt jedoch nur für Unicast-Frames. Broadcast- oder Multicast-Frames bleiben unbestätigt. Sinn würde es ohnehin nicht machen, dass jeder Empfänger diese Frames bestätigt. Erstens käme es zu einer möglicherweise extremen Sendeflut von ACKs und zweitens wüsste der Empfänger der ACKs nicht, wer diese versendet hat. Im ACK-Frame ist nämlich keine Absender-Angabe vorgesehen.

Interframe Space

Der folgende Abschnitt ist für den späteren Vergleich mit der Erweiterung 802.11e beziehungsweise dem WMM-Verfahren von besonderer Wichtigkeit.

Nachdem eine Station einen Unicast-Frame gesendet hat, folgt, wie bereits erwähnt, eine kurze Sendepause. Anschließend bestätigt der Empfänger mit einem ACK. Vor einem weiteren Frame befindet sich wieder eine Sendepause. Zwischen jedem Frame gibt es diese mit Interframe Space (IFS) bezeichnete Zeit. Im 802.11-Standard sind vier unterschiedliche IFSs definiert, die die Bezeichnungen SIFS, PIFS, DIFS und EIFS (Erklärung siehe unten) haben. Diese sind nötig um unterschiedliche Prioritäten beim Medienzugriff zu realisieren. Eine kürzere Wartezeit nach dem Sendeende eines Frames bedeutet eine höhere Wahrscheinlichkeit den Medienzugriff zu erlangen.

- SIFS steht für Short Interframe Space und hat die kürzeste Dauer und somit auch die höchste Priorität. Diese Zeit muss vergangen sein, bevor ein zugehöriges ACK-Frame versendet wird. Außerdem kommt SIFS vor CTS-Frames (Clear to Send, siehe unten RTS/CTS-Mechanismus) und vor Antworten auf Polling eines Point Coordinator zum Einsatz.
- PIFS ist der Point (Coordination Function) Interframe Space, der im bereits erwähnten PCF-Modus verwendet wird.
- DIFS ist der im DCF-Modus genutzte Distributed (Coordination Function) Interframe Space. Dies ist der zeitliche Mindestabstand vor der Übertragung von Daten- und Management-Frames. Hinzu kommt jedoch noch die Backoff-Wartezeit (siehe unten).

- EIFS ist die Abkürzung für Extended Interframe Space. Wenn an einer Station vom Physical- an den MAC-Layer der Empfang eines fehlerhaften Frames gemeldet wird, beginnt diese Station EIFS herunterzuzählen. EIFS ist dabei so lang bemessen, dass eine andere Station die Möglichkeit hat, dieses vermeintlich fehlerhafte Frame mit einem ACK zu bestätigen. Empfängt die Station, die sich in der EIFS-Phase befindet, ein korrektes Frame, bricht sie diese ab und setzt mit dem normalen Betrieb fort.

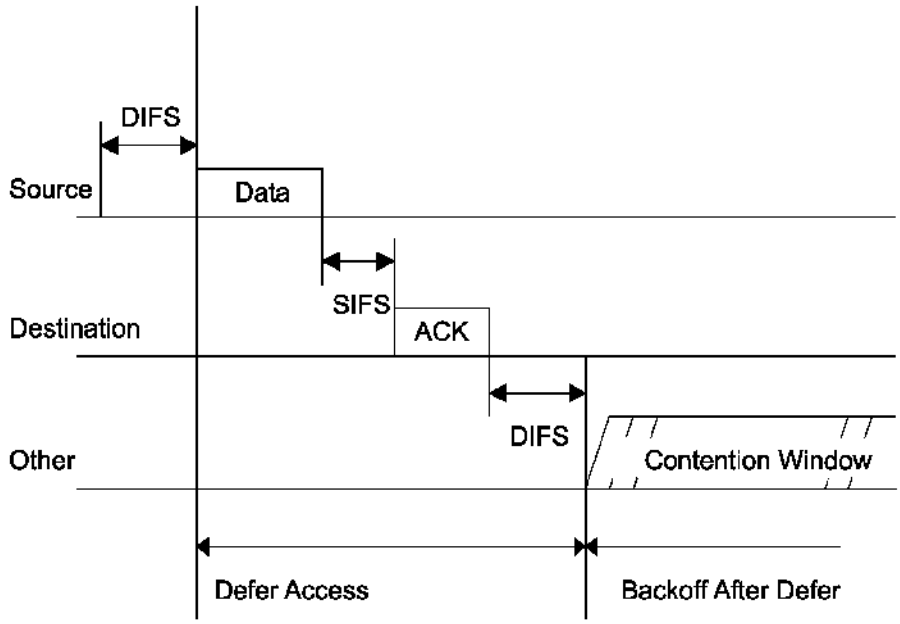


Abbildung 1: Ablauf einer Übertragung (entnommen aus [3])

Für die Zeiten gilt SIFS < PIFS < DIFS < EIFS. Die genaue Länge der IFS-Zeiten hängt davon ab, welcher Physical Layer verwendet wird. Für 802.11 sind drei unterschiedliche PHYs definiert, weitere für 802.11b und 802.11g et cetera. Je PHY ist die Dauer von SIFS und der Slot Time fest (→ Anhang D, S.88). Die anderen Zeiten lassen sich nach folgenden Formeln berechnen:

- PIFS = aSIFSTime + aSlotTime
- DIFS = aSIFSTime + 2 * aSlotTime
- EIFS = aSIFSTime + DIFS + (8 * ACKSize) + aPreambleLength + aPLCPHeaderLength[5]

5 aPreambleLength und aPLCPHeaderLength resultieren aus dem genutzten Physical Layer; siehe Anhang D, S.88

Hierbei ist ACKSize die Größe eines ACK-Frames in Byte. Der Term (8 * ACK-Size) + aPreambleLength + aPLCPHeaderLength wird in Mikrosekunden ausgedrückt. Er beschreibt die Zeit, die für die Übermittlung dieser drei Teile benötigt wird; wobei die niedrigste Datenrate, die der benutzte PHY unterstützen muss, zu Grunde gelegt wird.

Backoff-Zeit und Contention Window

Wenn eine Station senden möchte, muss sie erst einmal abwarten bis das Medium nicht mehr belegt ist. Anschließend beginnt sie mit dem Countdown des Interframe Spaces. Bleibt das Medium weiter unbelegt, wird nach Ablauf des DIFS oder EIFS noch eine Backoff-Wartezeit angehängt, bevor dann gesendet wird. Kommt ihr eine andere Station während der Backoff-Zeit zuvor, so wird die verbleibende Zeit gespeichert und weiter heruntergezählt, wenn das Medium wieder frei ist. So ist sichergestellt, dass jede Station auf jeden Fall irgendwann die Möglichkeit bekommt zu senden.

Die Backoff-Zeit ist zufällig gewählt und soll Kollisionen weitestgehend verhindern. Ohne diese weitere Wartezeit würden möglicherweise viele Stationen gleichzeitig nach dem Ablauf von DIFS oder EIFS anfangen zu senden. Kollisionen wären vorprogrammiert.

Die Backoff-Zeit berechnet sich nach der Formel

BackoffTime = Random() * aSlotTime,

wobei aSlotTime vom benutzten PHY abhängt. Random() ist ein Integer-Wert aus dem Intervall [0, CW]. CW ist das sogenannte Contention Window, dessen Wert wiederum vom PHY abhängt und im Bereich aCWmin bis aCWmax liegt. Es gilt $aCWmin \leq CW \leq aCWmax$. CW kann außerdem ausschließlich die Werte der Potenzen von 2, dekrementiert um 1 annehmen.

Beim ersten Sendeversuch wird CW auf aCWmin gesetzt. Jeder nicht erfolgreiche Sendeversuch bewirkt ein exponentielles Ansteigen des CW-Wertes, bis dieser schließlich den maximalen Wert aCWmax erreicht und diesen Wert so lange behält, bis er zurückgesetzt wird. Ein Zurücksetzen erfolgt nach einer erfolgreichen Übertragung.

Zusätzlich werden in jeder Station noch die Werte SSRC (Station Short Retry Count) und SLRC (Station Long Retry Count) verwaltet. Schlägt der Sendeversuch eines langen Frames fehl (länger oder gleich dem RTS-Grenzwert[6]), wird SLRC inkrementiert, andernfalls SSRC. Wenn diese Parameter ihren Maximalwert erreichen, wird CW auf aCWmin zurückgesetzt und die Aussendung des Frames als endgültig fehlgeschlagen deklariert. Der Frame wird verworfen.

In Abbildung 2 ist die Wirkungsweise des Backoff-Mechanismus mit mehreren Stationen zu sehen.

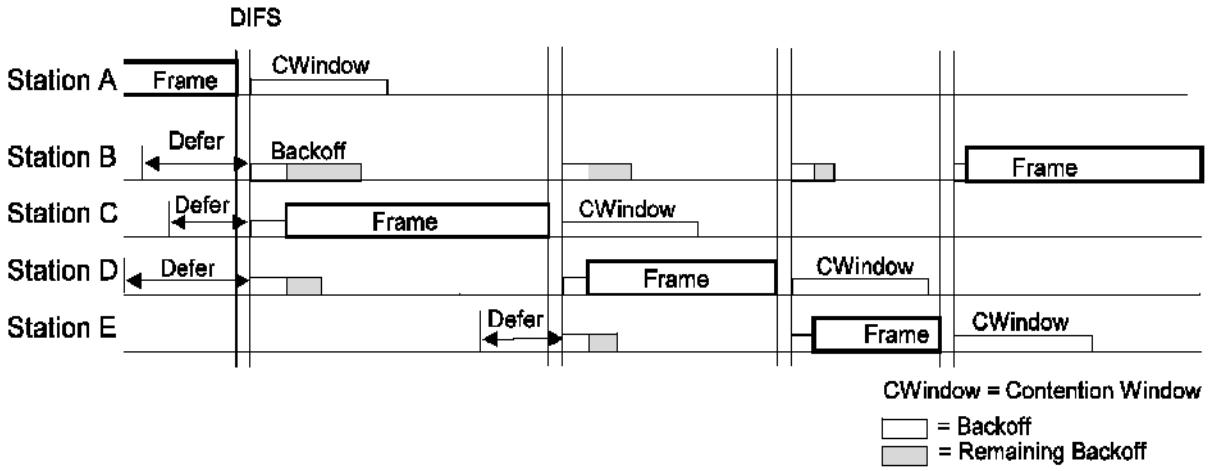


Abbildung 2: Backoff-Mechanismus (entnommen aus [3])

RTS/CTS-Mechanismus

In bestimmten Situationen kann es sinnvoll sein, dass WLAN-Geräte den Request to Send / Clear to Send-Mechanismus nutzen. Wie der englische Name schon sagt, sendet eine Station, die Daten übertragen möchte, vor der eigentlichen Datenübertragung zuerst ein RTS-Frame aus, mit dem sie sozusagen um Sendeerlaubnis bei der Empfängerstation bittet. Die Empfängerstation sendet daraufhin ein CTS-Frame zurück, mit dem sie die Sendeerlaubnis erteilt. Sinnvoll ist dieser Mechanismus beispielsweise zur Lösung des Hidden-Node-Problems. Hierbei gibt es zum Beispiel drei Stationen, von denen aber nur jeweils zwei im gleichen Empfangsbereich liegen.

6 RTS-Threshold – für das Senden von Paketen, die größer als der Grenzwert sind, wird das Request to Send / Clear to Send-Verfahren (RTS/CTS) angewendet

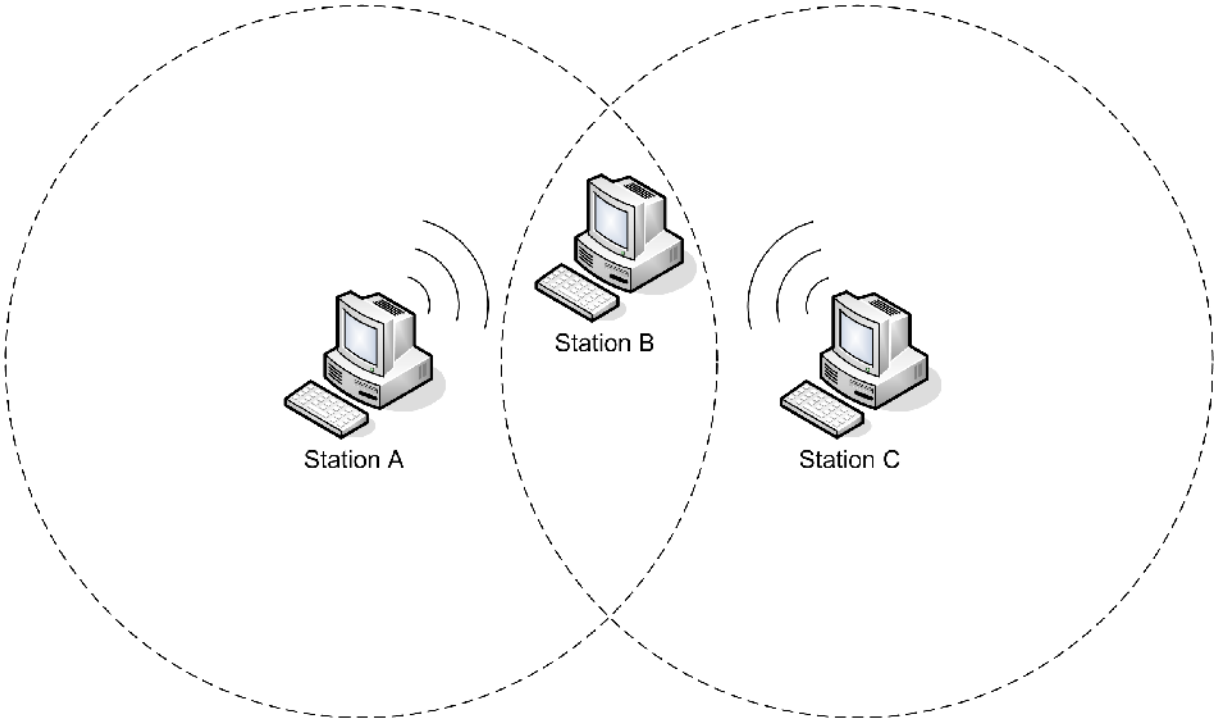


Abbildung 3: Hidden-Node-Problem

Station A möchte an B etwas senden. Station C liegt außerhalb des Empfangsbereichs von A, kann jedoch von B erreicht werden. Würde A nun ohne RTS/CTS an B senden, so würde C dies nicht mitbekommen und gegebenenfalls die Übertragung durch eigene Sendeversuche stören, weil sie annimmt, das Medium sei frei. Um dies zu verhindern sendet A also zuerst ein RTS-Frame, in dem das Duration/ID-Feld (siehe NAV) für die geplante Datenübertragung gesetzt ist. Damit wissen alle Stationen im Empfangsbereich von A über die Reservierung des Mediums Bescheid. Nur C hat davon noch nichts mitbekommen. B empfängt den RTS-Frame und sendet ihrerseits bei freiem Medium in ihrem Empfangsbereich das CTS aus. Darin ist wiederum der NAV-Wert gesetzt, durch den jetzt auch alle Stationen im Empfangsbereich von B, also auch C, wissen, dass das Medium für eine bestimmte Zeit belegt ist. Die Datenübertragung von A zu B kann nun ungestört erfolgen. Würde A kein CTS von B zurückerhalten, wüsste sie, dass das Medium im Empfangsbereich von B belegt sein muss. Dadurch sendet A nur das kurze RTS aus und belegt oder stört nicht mit ihrem langen Datenframe das Medium, obwohl der Empfänger B gar nicht bereit ist.

802.11-Frameformat

Wie bereits angedeutet, gibt es im WLAN nicht nur Daten-Frames, mit denen die eigentlichen Nutzdaten übertragen werden, die von den höheren Protokoll-Schichten an den MAC-Layer übergeben werden. Für die Funktion eines WLANs werden außerdem Kontroll- und Managementframes benötigt.

Zu den Kontrollframes zählen unter anderem die ACK- und die RTS-/CTS-Frames. Sie dienen zur Sicherstellung der korrekten Datenübertragung und zur Steuerung des Medienzugriffs. Mit Hife der Managementframes wird ein WLAN verwaltet.

Grundsätzlich ist ein 802.11-Frame wie folgt aufgebaut:

Bytes:	***2***	***2***	***6***	***6***	***6***	***2***	***6***	***0-2312***	***4***
	Frame Control	Duration/ ID	Address1	Address2	Address3	Sequence Control	Address4	Frame Body	FCS[7]
	MAC-Header								

Von den verschiedenen Frametypen werden nur bestimmte Felder des Headers genutzt. Deshalb ist der Header je nach Frametyp zwischen 10 und 34 Byte lang. Von allen Frametypen genutzt werden die Felder Frame Control, Duration/ID, Address1 und FCS.

Das Frame-Control-Feld ist in sich wiederum folgendermaßen aufgeteilt:

Bits:	***2***	***2***	***4***	***1***	***1***	***1***	***1***	***1***	***1***	***1***	***1***
	Protocol Version	Type	Subtype	To DS	From DS	More Fragment	Retry	Power Management	More Data	WEP	Order

Einzelheiten zum Frameaufbau können den IEEE-Standards 802.11 entnommen werden.

Als Beispiel eines Kontrollframes ist nachfolgend ein ACK-Frame gezeigt:

Bytes:	***2***	***2***	***6***	***4***
	Frame Control	Duration/ ID	Address1	FCS
	MAC-Header			

7 Frame Check Sequence

Für die Verwaltung eines WLANs werden, wie bereits erwähnt, Managementframes eingesetzt. Von diesem Frametyp gibt es sehr viele verschiedene. Auf die wichtigsten wird im Folgenden eingegangen.

Von einem AP werden zur Bekanntmachung seines WLANs so genannte Beacon-Frames ausgesendet. Diese werden normalerweise in einem Intervall von 102,4ms versandt. Die Beacon-Frames dienen unter anderem der Synchronisierung der Stationen die einem WLAN angehören. Außerdem enthalten sie diverse Daten, die neue Stationen benötigen, um sich dem WLAN anzuschließen. So wird beispielsweise die SSID, der Name des WLANs, über die Beacon-Frames ausgesendet. Des weiteren werden die vom AP unterstützten Datenraten und etwaige weitere Fähigkeiten, wie zum Beispiel WMM, mitgeteilt.

Eine Station, die sich einem bekannten oder unbekannten WLAN anschließen möchte, kann diesen Vorgang durch einen Probe-Request-Frame einleiten. In diesem Frame versendet sie entweder eine bestimmte SSID, um sich einem entsprechenden WLAN anzuschließen, oder alternativ eine Broadcast-SSID um Antwort von unbekannten WLANs zu erhalten. Außerdem sendet die Station ihre unterstützten Datenraten im Probe-Request-Frame mit. So kann beispielsweise ein empfangender AP entscheiden, ob die anfragende Station von ihm aufgenommen werden kann.

Wenn die Aufnahme möglich wäre, sendet der AP ein Probe-Response-Frame aus, in dem ähnlich wie im Beacon-Frame alle für die Station relevanten Daten über das WLAN enthalten sind.

Möchte sich die Station mit dem AP assoziieren, sendet sie im nächsten Schritt ein Association-Request-Frame. Um die Assoziierungsanfrage zu bestätigen, sendet der AP darauf ein Association-Response-Frame.

Anschließend folgt gegebenenfalls noch eine Authentifizierung mittels Authentication-Frames.

Wenn eine Station den Empfangsbereich eines AP verlässt und anschließend wieder in seine Reichweite hineinkommt, ist eine Reassoziierung notwendig, die mit Reassociation-Request- und -Response-Frames erledigt wird.

Dies waren nur die wichtigsten Management-Frames. In 802.11 sind noch einige mehr definiert.

2.2 Quality of Service

Unter dem Begriff Quality of Service (QoS) oder auch Dienstgüte versteht man eine bestimmte zugesicherte Qualität eines Dienstes. Ein solcher Dienst kann zum Beispiel Telefonie sein. Zur Bewertung der Dienstgüte werden im Netzwerkbereich unter anderem folgende Parameter zusammengefasst:

- Paketverlust:
  die Menge oder der prozentuale Anteil an Datenpaketen, die vom Sender abgesendet werden, aber den Empfänger nicht erreichen
- Verzögerung (engl. Delay):
  der Zeitraum zwischen Sendebeginn am Sender und Empfangsbeginn am Empfänger
- Jitter:
  die Schwankung bzw. Varianz der Verzögerung
- Durchsatz:
  die Datenmenge, die pro Zeiteinheit übertragen werden kann[8]

2.2.1 Notwendigkeit von QoS

Verschiedene Dienste haben unterschiedliche QoS-Anforderungen. Beispielsweise ist Paketverlust bei der Übertragung einer E-Mail nicht tolerabel, es würde Text fehlen oder die Mail insgesamt nicht darstellbar sein. Bei Telefonie hingegen ist Paketverlust bis zu einem gewissen Maße für den Menschen gar nicht bemerkbar. Die Anforderungen an die Verzögerung und den Jitter sind bei Telefonie dagegen sehr hoch. Lange Laufzeiten führen dazu, dass sich die Gesprächspartner ins Wort fallen oder der Meinung sind, der andere sei nicht mehr „in der Leitung“. Bei einer E-Mail ist es wiederum unerheblich, ob diese einige Sekunden früher oder später beim Empfänger eintrifft. Die Anforderungen an den Durchsatz sind sowohl bei E-Mail-Übertragung als auch bei Telefonie eher niedrig. Als Gegenbeispiel kann hier IPTV[9]

8 gesamter Absatz: [8]; [9]

9 Fernseh-Übertragung über Internet-Protokoll (IP)

genannt werden, was bei HDTV[10]-Streams ein Vielfaches des Telefonie-Durchsatzes benötigt.[8]

Zur Realisierung von QoS-Mechanismen im Ethernet und in IP-Netzen gibt es mehrere Ansätze. Grundsätzlich muss eine Unterscheidung zwischen wichtigen und unwichtigeren Datenpaketen möglich sein. Die Datenpakete müssen also entsprechend markiert sein, damit sie dann zum Beispiel in Routern bevorzugt behandelt und weitergeleitet werden können.

2.2.2 QoS auf OSI-Schicht 2[11]

Der ursprüngliche IEEE-802.3-Standard (Ethernet) bietet keinerlei Möglichkeiten zur Priorisierung von Datenübertragungen. Dieser Mangel wird jedoch durch die IEEE-Standards 802.1p (veröffentlicht in 802.1d) und 802.1q (VLANs) behoben. Es handelt sich um Erweiterungen des MAC-Layers, die Priorisierung auf OSI-Schicht 2 ermöglichen. QoS auf Schicht 2 wird auch mit Class of Service bezeichnet. In den genannten Standards wird ein weiteres Feld im Header eingefügt, welches unter anderem drei Bits für acht verschiedene Prioritätsklassen enthält. Somit sind Werte zwischen 0 und 7 möglich. 0 ist dabei die niedrigste Priorität, 7 die höchste. So markierte Pakete können bei der Weiterleitung in Switches oder Routern entsprechend bevorzugt behandelt werden. Voraussetzung ist jedoch, dass die Geräte 802.1p-kompatibel sind. Durch das Einfügen des zusätzlichen Header-Feldes wird nämlich das Frame größer als der Ethernet-Standard es vorsieht. Geräte, die nicht 802.1p-kompatibel sind, verwerfen so erweiterte Pakete deshalb möglicherweise.

Ethernet-Header mit VLAN-Tag:

Bytes	***>=7***	***1***	***6***	***6***	***4***			***2***	
	Preamble	SFD	Source Address	Destination Address	VLAN-Tag			Type/ Length	Data ...
					Tagged Frame Type	802.1p Priority	„0“	802.1q VLAN ID	
				Bits	***16***	***3***	***1***	***12***	

10 High Definition Television; hochauflösendes Fernsehen

11 [10]; [11]

2.2.3 QoS auf OSI-Schicht 3[12]

Auf OSI-Schicht 3 ist im IPv4-Header ein so genanntes Type of Service-Feld, kurz TOS, enthalten, das für die Priorisierung von Datenpaketen zur Verfügung steht. Im IPv6-Header gibt es hierfür das Traffic Class-Feld, das wie das TOS-Feld eine Länge von 8 Bit hat. Im inzwischen abgelösten RFC 791-Dokument wurde das TOS-Feld folgendermaßen unterteilt:

Precedence	D	T	R	reserved

- Precedence: 3 Bit für acht verschiedene Prioritätsklassen
  111 – Network Control
  110 – Internetwork Control
  101 – CRITIC/ECP
  100 – Flash Override
  011 – Flash
  010 – Immediate
  001 – Priority
  000 – Routine (Best Effort – Default)
- D: Minimierung der Verzögerung (Delay)
- T: Maximierung des Durchsatzes (Throughput)
- R: Zuverlässigkeit (Reliability)
  D, T, R jeweils 1 Bit; 1 für aktiv, 0 für normal
- reserved: 2 Bit (00) reserviert für spätere Verwendung

Im Dezember 1998 wurde die Definition des TOS-Feldes durch RFC 2474 erneuert. Die vorderen 6 Bit wurden als Differentiated Services-Feld (DS) umdefiniert. Der Wert des DS-Feldes wird als Differentiated Services Code Point (DSCP) bezeichnet. Die beiden letzten Bit blieben bis September 2001 weiter ungenutzt und werden seit RFC 3168 nun als ECN[13] verwendet. Das DS-Feld wurde so aufgebaut, dass eine gewisse Kompatibilität zu bisherigen Implementierungen, die auf RFC 791 aufbauen, sichergestellt ist. Deshalb gibt es im DS-Feld den Class Selector PHB (Per Hop Behavior), der aus den vorderen 3 Bit besteht und kompatibel zu den IP-Precedence-Bit des TOS-Feldes ist. Abzusendende Datenpakete werden schon beim Versand

12 [8]; [12]; [13]; [14]

13 Explicit Congestion Notification – Überlastungsmitteilung für IP-Flusskontrolle

am Endgerät einer entsprechenden Klasse zugeordnet, die mit dem Class Selector im Header markiert wird. Die nächsten 3 Bit werden für verschiedene Behandlungsstufen innerhalb einer Klasse genutzt. Durch den DSCP-Wert wird einem Datenpaket eine bestimmte Per Hop Behavior (PHB) zugeordnet, also ein Verhalten pro Netzwerkabschnitt.

2.2.4 Umsetzungsproblem bei Shared Media

Ein Datenpaket, das auf eine der genannten Weisen markiert wurde, kann auf dem Versandweg bei zum Markierungsverfahren kompatibel arbeitenden Routern entsprechend bevorzugt behandelt werden.

Ein Problem stellt jedoch der Zugriff auf das Versandmedium dar, wenn es sich um ein Shared Medium wie dem WLAN handelt. Ähnlich zum WLAN gab es das Problem des gemeinsam genutzten Mediums im kabelgebundenen LAN bei Bus-Netzwerken. Heutzutage sind jedoch kaum noch 10Base-2- oder Hub-basierte Base-T-Netzwerke im Einsatz. Als dies noch der Fall war, mussten sich alle Stationen, die am Bus angeschlossen waren, die zur Verfügung stehende Übertragungskapazität des Busses teilen. Bei hoher Netzlast kam es regelmäßig zu Kollisionen der Datenpakete. Bei den inzwischen meist eingesetzten Switches gibt es dagegen keinen Bus mehr, an dem alle Endgeräte gleichzeitig angeschlossen sind und immer nur eines senden kann. Es wird im Switch eine Art Direktverbindung zwischen jeweils zwei Stationen geschaltet, die Daten austauschen möchten. Außerdem sind viele gleichzeitige Verbindungen im Switch möglich und Kollisionen kommen kaum vor. Eine Priorisierung ist vor allem dort nötig, wo die Daten auf einen Engpass treffen, also beispielsweise bei Routern.

Da das Übertragungsmedium im WLAN aber ein gemeinsam genutztes Medium ist, gibt es dort das Problem, dass schon beim Versand am Endgerät unvorhersehbare Verzögerungen auftreten können. Das Medium kann durch andere Stationen belegt sein und so müssen wichtige Daten möglicherweise lange auf ihren Versand warten. Um nun im WLAN Daten priorisiert versenden zu können, ist also eine Priorisierung des Medienzugriffes nötig. Eine solche unterschiedliche Behandlung von Nutzdatenversand war im ursprünglichen IEEE 802.11-Standard nicht vorgesehen. Die einzige Unterscheidung beim Medienzugriff im WLAN geschieht zwischen

Steuerinformationen und Nutzdaten anhand des Interframe Spaces (IFS) (siehe Kapitel 2.1.3). Genau an dieser Stelle setzt das WMM-Verfahren an.

2.3 QoS im WLAN mittels Wireless Multimedia (WMM)

2.3.1 Entstehung von WMM

Um im WLAN QoS-Mechanismen bereitstellen zu können, begannen bei der IEEE im Jahr 2000 die Arbeiten an der Standard-Erweiterung 802.11e. Da sich im Laufe der Zeit unter den IEEE 802.11e-Entwicklern zwei Gruppen bildeten und es zu Meinungsverschiedenheiten kam, verzögerte sich die Fertigstellung des Standards über einen langen Zeitraum. Es konnte keine Einigung darüber erzielt werden, welche Funktionen besser für QoS im WLAN geeignet seien und somit in 802.11e standardisiert werden sollten.[15] Da aber bereits erste WLAN-Geräte mit einem Teil der geplanten Funktionen auf den Markt kamen, startete die Wi-Fi Alliance im September 2004 die Zertifizierung von WMM-kompatiblen WLAN-Geräten.

Die Wi-Fi Alliance ist eine gemeinnützige Organisation, in der sich inzwischen mehr als 250 Hersteller zusammengeschlossen haben. Hauptsächliches Ziel ist es, die Kompatibilität von WLAN-Geräten verschiedener Hersteller sicherzustellen. Die Geräte werden dafür getestet und von der Wi-Fi Alliance zertifiziert.

Zusätzlich zu WMM zertifiziert die Wi-Fi Alliance inzwischen auch Geräte, die „WMM Power Save“ unterstützen. Dabei handelt es sich ebenfalls um einen Teil des IEEE 802.11e-Standards, der dafür sorgen soll, dass der Energieverbrauch von WLAN-Geräten gesenkt wird, um zum Beispiel längere Akkulaufzeiten bei mobilen Geräten wie Laptops oder Handhelds zu ermöglichen. Es handelt sich hierbei um eine Verbesserung des Energiesparmodus, der schon im IEEE 802.11-Standard vorhanden war. Die Geräte nutzen hierbei „Schlafphasen“, in denen sie keine Daten senden und empfangen können, dafür aber sehr wenig Energie verbrauchen.[16]

2.3.2 Arbeitsweise von WMM[14]

WMM basiert auf den QoS-Erweiterungen eines frühen Entwurfs (Draft) des IEEE 802.11e-Standards. Fertiggestellt wurde dieser erst im November 2005.

14 [7] ;[6]; [1]

Für WMM wird aus der umfangreichen 802.11e-Erweiterung vorerst nur die Methode des Enhanced Distributed Channel Access (EDCA) genutzt. Hierbei wird der Medienzugriff dezentral auf jeder Station selbst gesteuert und der Zugriff erfolgt in der Contention Period. Die weiteren Möglichkeiten, die in 802.11e definiert sind, sollen möglicherweise später als optionale Funktionen in WMM aufgenommen werden. Hierzu zählt beispielsweise der HCF Controlled Channel Access (Hybrid Coordination Function CCA), der ähnlich zur Point Coordination Function (PCF) auf einer zentralen Steuerung des Medienzugriffs basiert.

WMM arbeitet mit vier Datenverkehrskategorien (Access Category, AC), die aus 802.1d abgeleitet und auch kompatibel zur TOS-Markierung sind. Jede Kategorie entspricht dabei einem bestimmten Prioritätslevel. Die genaue Zuordnung ist in WMM jedoch nicht festgelegt, sondern nur empfohlen. So können in einem Netzwerk auch abweichende Einstellungen vorgenommen werden, wenn dies gewünscht ist. Voraussetzung ist jedoch, dass der Hersteller eines WMM-Produktes diese Möglichkeit überhaupt implementiert. Welche Zuordnung gültig ist, soll der Access Point (AP) den Stationen, die QoS-kompatibel sind, mitteilen. Dies geschieht über die ausgesendeten Beacon- und Probe-Response-Frames.

Folgende Kategorien und Zuordnungen sind nach Wi-Fi Alliance in WMM definiert:

WMM-Kategorie/ Access Category (802.11e-Abk.)	***Beschreibung***	***User Priority (802.1d)***
Voice (VO)	Höchste Priorität: für VoIP gedacht mit geringer Latenz	7, 6
Video (VI)	Zweithöchste Priorität	5, 4
Best Effort (BE)	Standard-Kategorie für Geräte, die WMM nicht unterstützen und für normalen Datenverkehr, der nicht anfällig ist gegen geringe Verzögerungen, jedoch keinen besonders langen Verzögerungen unterliegen soll	0, 3
Background (BK)	Niedrigste Priorität: gedacht für Datenverkehr, der keine speziellen Anforderungen an Latenz und Durchsatz hat (z.B. Datei-Downloads oder Druckaufträge)	2, 1

Tabelle 2: WMM-Kategorien

Datenverkehr mit hoher Priorität erhält mit Hilfe des WMM-Verfahrens eine höhere Zugriffswahrscheinlichkeit auf das Übertragungsmedium und kann dadurch eher übertragen werden als niedrig priorisierter.

Für die Realisierung dieser unterschiedlichen Zugriffswahrscheinlichkeiten werden jeder AC zwei eigene Zeitparameter zugeordnet. Dies ist zum einen der Arbitration Interframe Space (AIFS), welcher in WMM-kompatiblen Stationen den DIFS (siehe Kap. 2.1.3) ersetzt und je nach AC eine unterschiedliche Dauer hat. Diese Dauer wird in Zeitschlitzen (time slots) angegeben durch die AIFS-Number (AIFSN). Die Zuordnung AC zu AIFSN kann Tabelle 3 entnommen werden. Der zweite Zeitparameter ist das Contention Window, welches ebenfalls von der AC abhängig ist und auf dem CW basiert, das schon in 802.11 für alle PHYs definiert ist (siehe Kap. 2.1.3). Durch die Festlegung dieser neuen CW-Werte kann eine kürzere Backoff-Wartezeit realisiert werden, die ebenfalls bewirkt, dass höhere Priorisierung eine höhere Medien-Zugriffswahrscheinlichkeit bewirkt. Verdeutlicht wird dieses Zeitverhalten durch Abbildung 4; die genauen Werte stehen in Tabelle 3.

AC	AIFSN		TXOP-Limit	CWmin	CWmax
	Client	AP			
BK	7	7	0[15]	aCWmin	aCWmax
BE	3	3	0	aCWmin	aCWmax
VI	2	1	a/g: 3008µs; b: 6016µs[16]	(aCWmin+1)/2-1	aCWmin
VO	2	1	a/g: 1504µs; b: 3264µs	(aCWmin+1)/4-1	(aCWmin+1)/2-1

Tabelle 3: EDCA-Werte

Beim Entwurf von WMM beziehungsweise EDCA wurde darauf geachtet, dass eine Koexistenz von WMM- und Nicht-WMM-Geräten in einem WLAN sichergestellt ist. Deshalb muss das Verfahren auf den vorhandenen Standards aufbauen. In [7] heißt es dazu, dass Datenverkehr, der keiner AC zugeordnet ist, standardmäßig der Best Effort-Kategorie zugeordnet wird. Dies gelte für ältere Geräte, die keine WMM-Unterstützung haben, als auch für Geräte bei denen WMM deaktiviert sei.

Dass Geräte ohne WMM aber gerade nicht gleichberechtigt mit WMM-Kategorie BE

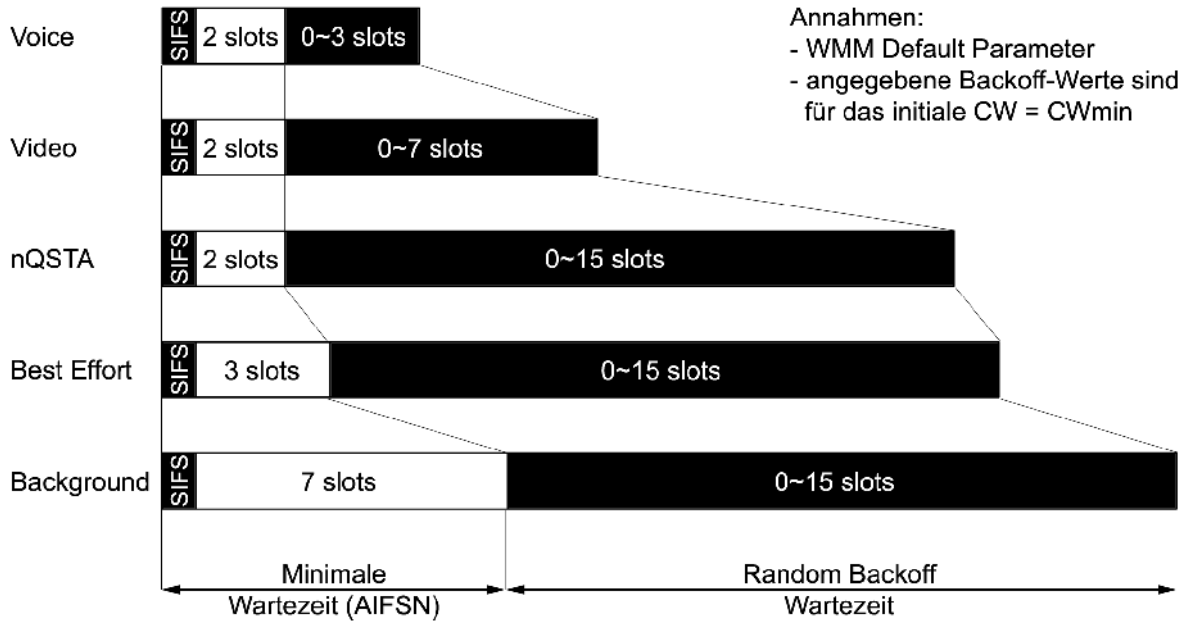


Abbildung 4: WMM AC Timing (z.T. entnommen aus [7])

sind, ist in Abbildung 4 zu sehen. Die Priorität von Geräten ohne WMM-Unterstützung (nQSTA[17]) liegt zwischen den WMM-Kategorien VI und BE. Denn wie

15 Gilt für alle PHYs. 0 bedeutet, dass nur eine MSDU gesendet werden darf.

16 a, b und g stehen für die PHYs der IEEE-Standards 802.11a, b und g.

17 nQSTA – non Quality of Service Station = WLAN-Station ohne WMM-Unterstützung

bereits erläutert, hat bei solchen Stationen DIFS eine Länge von DIFS = SIFS + 2 slots. Hinzu kommt die Backoff-Wartezeit, die im Initialzustand von CW gleich aCWmin ist, also 0 bis 15 slots. Wie man in der Abbildung sieht, resultiert daraus eine um einen Slot kürzere Wartezeit für nQSTAs als für Daten der WMM-Kategorie BE. Somit haben nQSTAs in einem WLAN mit WMM- und Nicht-WMM-Geräten eine etwas höhere Wahrscheinlichkeit den Medienzugriff zu erlangen.

Damit WMM funktionieren kann, müssen die Anwendungen, die auf einer Station genutzt werden, ihre Datenpakete durch Markierung einer der vier WMM-Kategorien zuordnen. Dafür kann zum Beispiel das TOS- beziehungsweise das TrafficClass-Byte genutzt werden, was dann in eine WMM-Kategorie umgesetzt (gemappt) wird. In der Station gibt es, wie in Abbildung 5 zu sehen ist, für jede dieser Kategorien eine Warteschlange, in die die Pakete vor dem Versand eingereiht werden.

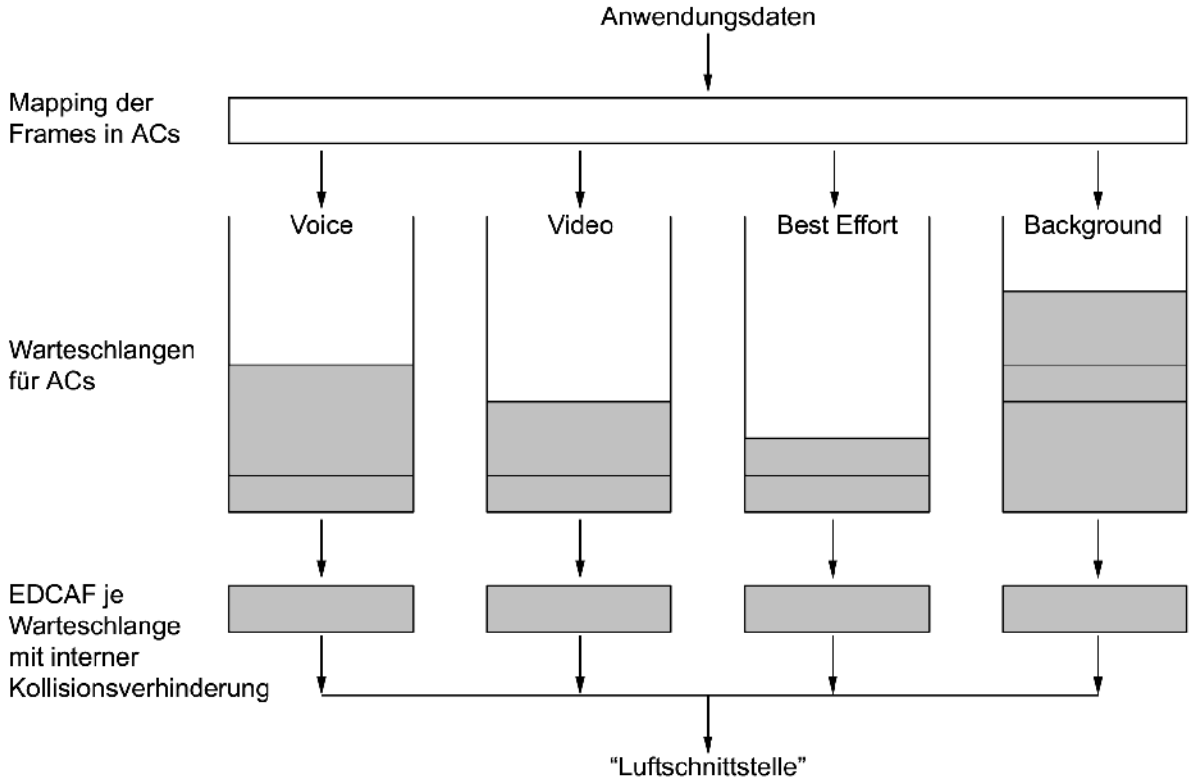


Abbildung 5: AC-Warteschlangen (z.T. entnommen aus [7])

Jede Warteschlange besitzt eine eigene EDCA-Funktion (EDCAF), die feststellt, ob ein Datenpaket aus ihrer Warteschlange gesendet werden darf. Dafür hört jede EDCAF das Medium ab und stellt fest, ob es für die Dauer von AIFS[AC] durchgehend unbelegt war. Ist dies der Fall, muss sie noch die Backoff-Zeit

abwarten, die durch das CW beeinflusst wird. Ist das Medium dann immer noch frei, kann theoretisch gesendet werden. Zunächst wird aber noch innerhalb der Station zwischen den einzelnen Warteschlangen sichergestellt, dass es zu keiner internen Kollision kommt. Ist dies ausgeschlossen, so wird aus der Warteschlange gesendet, dessen EDCAF die Zugriffsberechtigung erhalten hat. Diese Zugriffsberechtigung wird Transmission Opportunity (TXOP) genannt.[18]

Hat eine Station beziehungsweise eine ihrer EDCAFs die TXOP erlangt, darf sie für eine bestimmte maximale Dauer diese Medienzugriffsberechtigung behalten und entsprechend lange Daten senden. Wie lang diese Dauer ist, hängt davon ab, welcher PHY benutzt wird und welche EDCAF die TXOP erhalten hat. In 802.11e wird diese Dauer TXOP-Limit genannt und den Stationen vom AP mitgeteilt. Standardmäßig gilt für die Kategorien BK und BE, dass je TXOP nur eine MAC Service Data Unit (MSDU) übertragen werden darf. Für die ACs VI und VO gilt, dass diese gegebenenfalls mehrere Frames senden dürfen, solange das TXOP-Limit nicht überschritten wird und die Daten aus der selben Warteschlange stammen. Die genauen Werte für das TXOP-Limit können Tabelle 3 entnommen werden. Wie man dort sieht, ist das Limit für die Kategorie VI doppelt so hoch wie für VO, obwohl die Prioritäten umgekehrt sind. Trotzdem ist dies sinnvoll, denn Videodaten (VI) haben in der Regel eine viel höhere Datenrate als Sprachdaten. Für Sprache ist vor allem die geringe Verzögerung beim Versand wichtig. Dies wird durch die kleineren CW-Werte erreicht, die eine wahrscheinlichkeitstheoretisch niedrigere Wartezeit als bei der Kategorie VI bewirken sollen.

Für WMM wurden im MAC-Header Anpassungen nötig. Am Ende des Headers wurde ein zusätzliches 2 Byte langes Feld „QoS Control“ angehängt, in dem unter anderem die WMM-Kategorie und das angefragte TXOP-Limit vermerkt wird. Der Frame Body wurde in 802.11e von 2312 Byte auf 2304 Byte gekürzt[19].

18 Weitere Informationen in [7] Kap. 9.9.1.5

19 Frame Body in 802.11e: 2304 Byte plus jeglicher Overhead durch Verschlüsselung

Ein QoS-Frame sieht damit im Allgemeinen wie folgt aus:

Bytes:	***2***	***2***	***6***	***6***	***6***	***2***	***6***	***2***	***0-2304***	***4***
	Frame Control	Duration/ ID	Address1	Addr.2	Addr.3	Sequence Control	Addr.4	QoS Control	Frame Body	FCS
	MAC-Header									

Insgesamt betrachtet verspricht die Theorie von WMM eine durchaus brauchbare, wenn auch nicht optimale Realisierungsmöglichkeit von QoS im WLAN. Nicht ganz optimal deshalb, weil das Verfahren auf Wahrscheinlichkeit basiert. Eine höhere Priorität bewirkt zwar eine wahrscheinlichkeitstheoretisch schnellere Erteilung des Medienzugriffsrechts, aber es kann durch die Zufallsfunktion des CW bei einzelnen Übertragungen auch mal länger dauern. Außerdem dürfte es zu Problemen kommen, wenn viele Stationen im WLAN mit der gleichen Priorität versuchen zu senden, denn dabei herrscht dann wieder Gleichberechtigung.

Wie gut WMM in der Praxis funktioniert, wird durch diverse Tests in Kapitel 5 untersucht. Welche Vorbereitungen für die Durchführung dieser Bachelor-Thesis nötig waren, wird im folgenden Kapitel erläutert.

3 Vorarbeiten

Für die Durchführung dieser Bachelor-Thesis waren im Voraus intensive Recherchen nötig, um unter anderem einen Überblick über die aktuellen Entwicklungen und Marktverfügbarkeiten zu erlangen.

Die Ergebnisse stellten sich im Wesentlichen wie folgt dar:

Der IEEE-Standard 802.11e wurde erst am 11. November 2005 fertiggestellt. Geräte, die 802.11e in der Endversion unterstützen, waren im Markt nicht verfügbar. Es gab jedoch Geräte die WMM unterstützen und somit auf einem Draft von 802.11e basieren. Die Wi-Fi Alliance, die seit September 2004 Geräte mit WMM-Unterstützung zertifiziert, konnte jedoch keine Auskunft darüber erteilen, welche Draft-Version von 802.11e die Basis für WMM darstellt. An dieser Stelle sei noch angemerkt, dass WMM vor seiner Umbenennung den Namen WME (Wireless Media Extensions) trug.

Auf Basis dieser Erkenntnisse wurde die Entscheidung getroffen, das WMM-Verfahren im Rahmen dieser Bachelor-Thesis genau zu untersuchen.

4 Aufbau einer Testumgebung

In diesem Kapitel geht es um den Aufbau einer Testumgebung, um das WMM-Verfahren im praktischen Einsatz genau untersuchen zu können. Zunächst wird auf die Anforderungen und Festlegungen an und für diese Testumgebung eingegangen.

4.1 Anforderungen und Festlegungen

Im Fachbereich Datennetze war bereits ein WLAN-Router WRT54G[20] von Linksys in der Hardware-Version 2.2 vorhanden, der nach Möglichkeit unter anderem als AP zum Einsatz kommen sollte. Das Gerät ist eine Kombination aus Router, Switch und WLAN-AP. Der exakte interne Aufbau ist nicht bekannt. Es gibt jedoch Erkenntnisse durch Reverse Engineering. Nach Informationen auf den Internetseiten von Seattle Wireless[17] ist der Aufbau wie folgt: Der VLAN-fähige Switch besitzt 6 Ports, von denen 5 nach außen geführt sind und einer über eth0 an den internen Router angeschlossen ist. Die 5 externen Ports sind zum einen der WAN-Port als vlan1 und zum anderen die 4 LAN-Ports als vlan0. Das WLAN ist als eth1 an den Router angebunden. Der Aufbau ist in Abbildung 6 dargestellt. Der Router ist mit einer 200 MHz-CPU ausgestattet und besitzt 16 MiB[21] RAM und 4 MiB Flash-Speicher für das Linux-Betriebssystem. Die LAN/WAN-Ports unterstützen die Bandbreiten 10 und 100 MBit/s nach IEEE 802.3(u). Der WLAN-Port arbeitet nach IEEE 802.11b+g und bietet somit maximal 54 MBit/s an Bandbreite.

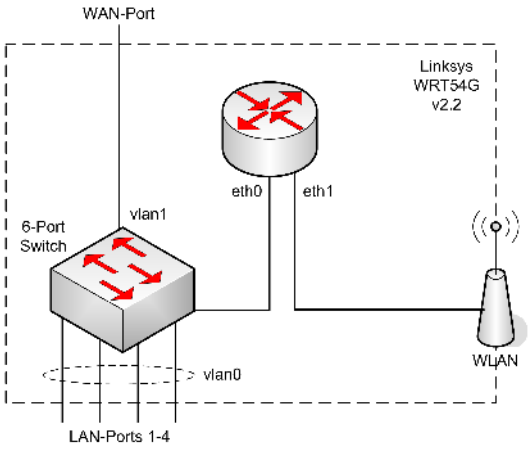


Abbildung 6: interner Aufbau des Linksys WRT54G

Der große Vorteil dieses Gerätes ist das auf Linux basierende Betriebssystem. Seit der Markteinführung des WRT54G ist dieser Router wegen seiner Linux-Basis sehr beliebt und es existieren inzwischen diverse Projekte, die alternative Firmware für

20 http://www.linksys.de → Produkte → Drahtlos → Netzwerk Basisprodukte → Breitbandrouter

21 Mebibyte (2^{20} = 1024^2 = 1.048.576 Byte), siehe Norm IEC 60027-2

das Gerät anbieten. Beispiele hierfür sind OpenWRT[22], DD-WRT[23] und andere. Mit Hilfe dieser Firmwares lassen sich auf dem Gerät Funktionalitäten implementieren, die mit der vom Hersteller mitgelieferten Firmware nicht möglich sind. Eigene Entwicklungen sind mit diesen Alternativ-Firmwares ebenfalls möglich.

Anmerkung: seit einigen Monaten bietet Linksys unter dem Produktnamen WRT54G einen ähnlichen Router an, der nicht mehr auf Linux sondern auf VxWorks basiert und weniger Speicherkapazität hat. Das Linux-basierende Modell heißt seitdem WRT54GL. Bis zum 24. Juni 2006 schien es als fast unmöglich, Alternativ-Firmwares auf dem neuen WRT54G zu betreiben, da Hardwaremodifikationen nötig waren um eine neue Firmware auf das Gerät zu bringen. Seit oben genanntem Datum ist bekannt, dass es nun wohl doch gelungen ist, dies ohne Modifikationen zu erreichen.[18]

Für die Computer, die in der Testumgebung als WLAN-Clients und Messendpunkte zum Einsatz kommen sollten, wurde Linux als Betriebssystem ausgewählt. Linux ist bekanntlich frei verfügbar und Quelltext-offen. Somit spricht einerseits der geldwerte Vorteil für den Einsatz von Linux und andererseits der Vorteil, dass Teile des Betriebssystems angepasst werden können, wenn dies nötig ist. Gleiches gilt für einen großen Teil der Anwendungssoftware, die für Linux verfügbar ist. Auch Treiber für Hardwarekomponenten liegen meist im Quelltext vor, so dass deren Arbeitsweise genauer untersucht werden kann. Bei Windows als Betriebssystem ist beides genau umgekehrt. Windows darf beziehungsweise kann nur mit einer gültigen Lizenz vernünftig betrieben werden, es fallen also Kosten an. Außerdem ist der Windows-Quelltext zum größten Teil nicht offen gelegt. Anpassungen sind also in der Regel nicht machbar. Für Anwendungssoftware und Treiber gilt unter Windows ähnliches.

Eine selbstverständliche Anforderung war, dass die zum Einsatz kommenden WLAN-Geräte WMM-kompatibel sein sollten. Dass dies im Zusammenhang mit der Entscheidung für Linux als Betriebssystem nicht ganz einfach zu erfüllen war, kann dem folgenden Kapitel entnommen werden. Außerdem musste festgestellt werden, ob der oben genannte WLAN-Router WMM unterstützt oder ob dies zum Beispiel durch andere Firmware nachgerüstet werden kann.

22 http://openwrt.org/

23 http://www.dd-wrt.de/

Das WLAN sollte möglichst durch WPA-Verschlüsselung abgesichert werden, um unbefugten Zugriff zu verhindern.

Weiter sollten die Tests und Messungen möglichst klare Aussagen über die Funktionalität von WMM als QoS-Mechanismus ermöglichen. Es sollten also verschiedene Szenarien aufgebaut werden, in denen durch bestimmte Messungen festgestellt werden sollte, ob eine Priorisierung erfolgt. Welche Szenarien gemessen und analysiert wurden, wird in Kapitel 5, S. 38 ff genau erläutert.

4.2 Marktrecherche

Wie bereits erwähnt, war der WLAN-Router WRT54G schon vorhanden. Auf dem Gerät befand sich zu Beginn meiner Arbeiten noch eine OpenWRT-Firmware aus einem älteren Projekt. Da Internet-Recherchen zu diesem Zeitpunkt ergeben hatten, dass OpenWRT keine Unterstützung von WMM bietet, musste nach Alternativen gesucht werden. Für DD-WRT[24] stellte sich ähnliches heraus. Die deutschen Webseiten des Herstellers Linksys offerierten nur die „ursprüngliche Version, deutsche Ausführung“ der Firmware, datiert vom 2. Februar 2005. Seitdem war also bisher noch kein einziger Fehler in der Firmware entdeckt oder schlicht nicht korrigiert worden. Die englischen Linksys-Seiten waren dagegen vielversprechender. Ein Blick in die Versionshistorie der englischen Firmware zeigte, dass dort regelmäßig Fehler behoben und neue Funktionen ergänzt wurden. WMM war aber nicht explizit in der Historie aufgeführt. Eine Nachfrage beim US-Support ergab jedoch, dass „die aktuelle US-Firmware WMM unterstützt“. „Der Betrieb eines für Europa produzierten Gerätes mit einer US-Firmware sei zwar grundsätzlich möglich, aber nicht empfohlen“, so die weiteren Informationen des Support-Mitarbeiters. Da somit nichts gegen einen Test der US-Firmware sprach und WMM-Unterstützung gegeben sein sollte, konnte zunächst nach WLAN-Karten für die Clients gesucht werden.

Für die Client-PCs sollten PCI-WLAN-Karten angeschafft werden. Über die Gerätesuche auf der Homepage der Wi-Fi Alliance ist es möglich, Geräte zu finden, die WMM-zertifiziert sind. Das große Problem ist jedoch, dass die meisten Hersteller nur Treiber für Windows mit ausliefern. Bisher ist mir überhaupt kein Hersteller

24 Inzwischen unterstützt DD-WRT nach eigenen Angaben auch WMM.

bekannt, der seine WLAN-Karten mit Linux-Treibern ausliefert. Deshalb ist man zum Betrieb unter Linux meist auf Alternativ-Treiber angewiesen, die von Linux-Entwicklergruppen oft kostenfrei im Internet bereitgestellt werden. Diese Treiber unterstützen in der Regel aber nur bestimmte WLAN-Chipsätze und können deshalb nur mit ganz bestimmten WLAN-Karten betrieben werden. Aus diesem Grund werden meist Kompatibilitätslisten auf den Projekthomepages bereitgestellt, die zum Teil auf Rückmeldungen von Nutzern basieren. Ganz sicher kann man deshalb nie sein, dass Hardware nach dem Kauf wirklich unter Linux betrieben werden kann. Vor allem besteht das Problem deshalb, weil es vorkommt, dass Hersteller den Chipsatz eines Produktes ändern, aber dieses weiter unter dem selben Produktnamen anbieten.

Um nun WLAN-Karten beschaffen zu können, die unter Linux einsetzbar sind und zusätzlich WMM-Funktionalität bieten, waren erneut zeitaufwändige Recherchen nötig. Die Ergebnisse waren wie folgt: Der MadWifi[25]-Treiber unterstützt laut Projekthomepage vermutlich WMM. „Ich bin nicht sicher, ob es [WMM] komplett implementiert ist, aber die Möglichkeiten scheinen vorhanden zu sein“, so die übersetzte Aussage einer MadWifi-Seite[26]. Allnets PCI-WLAN-Karte ALL0281[27] nutzt einen Atheros-Chipsatz, der vom MadWifi-Treiber unterstützt wird. Da keine anderen Treiberprojekte gefunden werden konnten, die WMM definitiv unterstützen, wurde die Entscheidung getroffen, oben genannte WLAN-Karte von Allnet zu beschaffen und zu testen.

4.3 Installation und Konfiguration

Auf den Computern wurde die Linux-Distribution Debian in der Variante „unstable“ mit einem 2.6er Kernel installiert. Die weiteren Installationshinweise beziehen sich, soweit nicht anders angegeben, auf dieses System.

25 [http://madwifi.org]

26 [http://madwifi.org/wiki/ChipsetFeatures/WMM]

27 Gilt auch für das Modell Allnet ALL0281A, das in Computer Z eingesetzt wurde. [http://www.allnet.de/produkte/27101.html]

4.3.1 WRT54G

Zunächst wurde die offizielle US-Firmware von Linksys in der Version 4.20.7 auf den Router WRT54G geflasht. Ein Nachteil der US-Firmware besteht beim Einsatz in Europa darin, dass 2 WLAN-Kanäle weniger zur Nutzung zur Verfügung stehen. In Deutschland ist der Betrieb von WLANs im 2,4 Ghz-Band auf den Kanälen 1 bis 13 erlaubt. In den USA gibt es jedoch nur eine Zulassung für die Kanäle 1 bis 11. Deshalb ist der Betrieb auf den Kanälen 12 und 13 mit der US-Firmware nicht möglich.

Die Konfiguration des Routers erfolgt bei der Linksys-Firmware ausschließlich über eine Weboberfläche. Zunächst wurde hierüber ein neues Passwort für den Konfigurationszugriff eingerichtet. Die IP-Parameter wurden so festgelegt, dass der Betrieb des Routers am Netzwerk des Fachbereichs möglich war. Nach außen hin bekam der Router die öffentliche IP-Adresse 139.6.19.151. Das interne Netzwerk des Routers erhielt die private Adresse 192.168.1.0/24. Das WLAN wurde so eingerichtet, dass der Zugriff nur mit WPA-Verschlüsselung (PSK TKIP) möglich war. Außerdem wurde der G-only-Mode aktiviert, so dass sich ausschließlich Geräte, die den 802.11g-Standard unterstützen, am AP anmelden konnten. Dies wurde deshalb so eingerichtet, damit alle Messungen bei bestmöglicher WLAN-Performance erfolgen konnten. In einer gemischten WLAN-Umgebung aus b- und g-Clients würde nämlich eine längere Slot-Time und ein größeres aCWmin genutzt (→ Anhang D, S.88), was die Performance verschlechtert. Aktiviert oder deaktiviert wurde je nach Messung die WMM-Funktionalität. Das entsprechende Drop-Down-Feld für diese Einstellung ist in der Weboberfläche unter dem Punkt „Applications & Gaming – QoS – Wireless QoS“ zu finden. Spezielle Konfigurationen sind dort bezüglich WMM nicht möglich.

4.3.2 Allnet-PCI-WLAN-Karten ALL0281(A) mit MadWifi-Treiber[28]

Nach dem Einbau der PCI-WLAN-Karten in die Computer wurde mit Root-Rechten der Befehl „`lspci`“ auf der Shell ausgeführt.

```
# lspci
```

28 [http://madwifi.org/wiki/UserDocs/Distro/Debian/MadWifi]

In der ausgegebenen Liste muss die WLAN-Karte aufgeführt werden. Damit ist sichergestellt, dass die Karte korrekt erkannt wurde.

Als nächstes mussten in der Datei `/etc/apt/sources.list` Quellenangaben ergänzt werden, damit MadWifi über „`apt`“ nachinstalliert werden konnte. Am 13.06.2006 waren dies beispielsweise folgende Einträge:

```
deb http://ftp.au.debian.org/debian/ experimental main
contrib non-free
deb-src http://ftp.au.debian.org/debian/ experimental
main contrib non-free
```

Danach wurden die MadWifi-Pakete per `apt` von den eben ergänzten Quellen heruntergeladen und mit Hilfe des Modul-Assistenten installiert:

```
$ su
# apt-get update
# apt-get install madwifi-ng-source
# apt-get install madwifi-ng-tools
# m-a prepare
# m-a a-i madwifi-ng
# depmod -a
```

Mit Root-Rechten wurde anschließend mit dem Befehl

```
# modprobe ath_pci
```

der MadWifi-Treiber ins laufende System geladen. Damit war die Treiber-Installation bereits abgeschlossen. Für den WPA-verschlüsselten WLAN-Betrieb war noch die Installation des „WPA Supplicant“ nötig. Zunächst war dieser herunterzuladen[29] und zu entpacken. In diesem Projekt kam der WPA Supplicant in Version 0.4.8 zum Einsatz. Im Verzeichnis mit dem entpackten WPA Supplicant musste im nächsten Schritt die Datei „.config“ angepasst werden. Der Inhalt war anschließend folgender:

```
CONFIG_DRIVER_MADWIFI=y
CFLAGS += -I/usr/src/modules/madwifi-ng"
CONFIG_CTRL_IFACE=y
```

29 http://hostap.epitest.fi/wpa_supplicant/

Der Ausdruck `/usr/src/modules/madwifi-ng` gibt hierbei den Pfad zu den MadWifi-Modulen an, die bereits zuvor aus dem Quellcode erstellt worden waren.

Mit `make` wurde dann der Quellcode des WPA Supplicant kompiliert und anschließend die Dateien wpa_cli, wpa_passphrase und wpa_supplicant in das Verzeichnis `/usr/local/sbin` kopiert.

Als nächstes wurde der Preshared Key erzeugt:

```
# wpa_passphrase MEINESSID meinepassphrase

network={
 ssid="MEINESSID"
 #psk="meinepassphrase"
 psk=edda86468aa67c3f7[...]9011d63e699f5381a5b77e0c2a
}
```

Die obige (gekürzte) Ausgabe des `wpa_passphrase`-Befehls wurde in die Datei `/etc/wpa_supplicant.conf` kopiert und mit

```
# chmod 640 /etc/wpa_supplicant.conf
```

wurden die Zugriffsrechte geändert.

Abschließend wurde der WPA Supplicant gestartet:

```
# wpa_supplicant -Bw -Dmadwifi -iath0
                     -c/etc/wpa_supplicant.conf
```

Nach der Konfiguration der IP-Adresse, der Subnetz-Maske und des Standard-Gateways mit `ifconfig` war die WLAN-Karte betriebsbereit.

4.4 Gesamtübersicht der Testumgebung

Der Linksys-Router WRT54G war bei allen Messungen als AP im Einsatz. Außerdem war dieser das Gateway zum Netzwerk des Fachbereichs Datennetze.

Als Clients standen im Endausbau bis zu vier Linux-PCs zur Verfügung, die teilweise sowohl kabelgebunden als auch kabellos an den Router angekoppelt werden

konnten. Je nach Messszenario war manchmal auch der gleichzeitige Betrieb von LAN- und WLAN-Interface an bestimmten PCs notwendig.

Alle Computer befanden sich in einer maximalen Entfernung von zwei Metern zum AP, um eine bestmögliche Signalqualität und somit eine voll nutzbare Bandbreite zu erzielen.

Die folgende Abbildung 7 zeigt die gesamte Testumgebung mit allen verwendeten Geräten, allen ihren IP-Adressen sowie der Verkabelung. Bei den Messungen waren in der Regel nicht alle Interfaces der Clients in Betrieb.

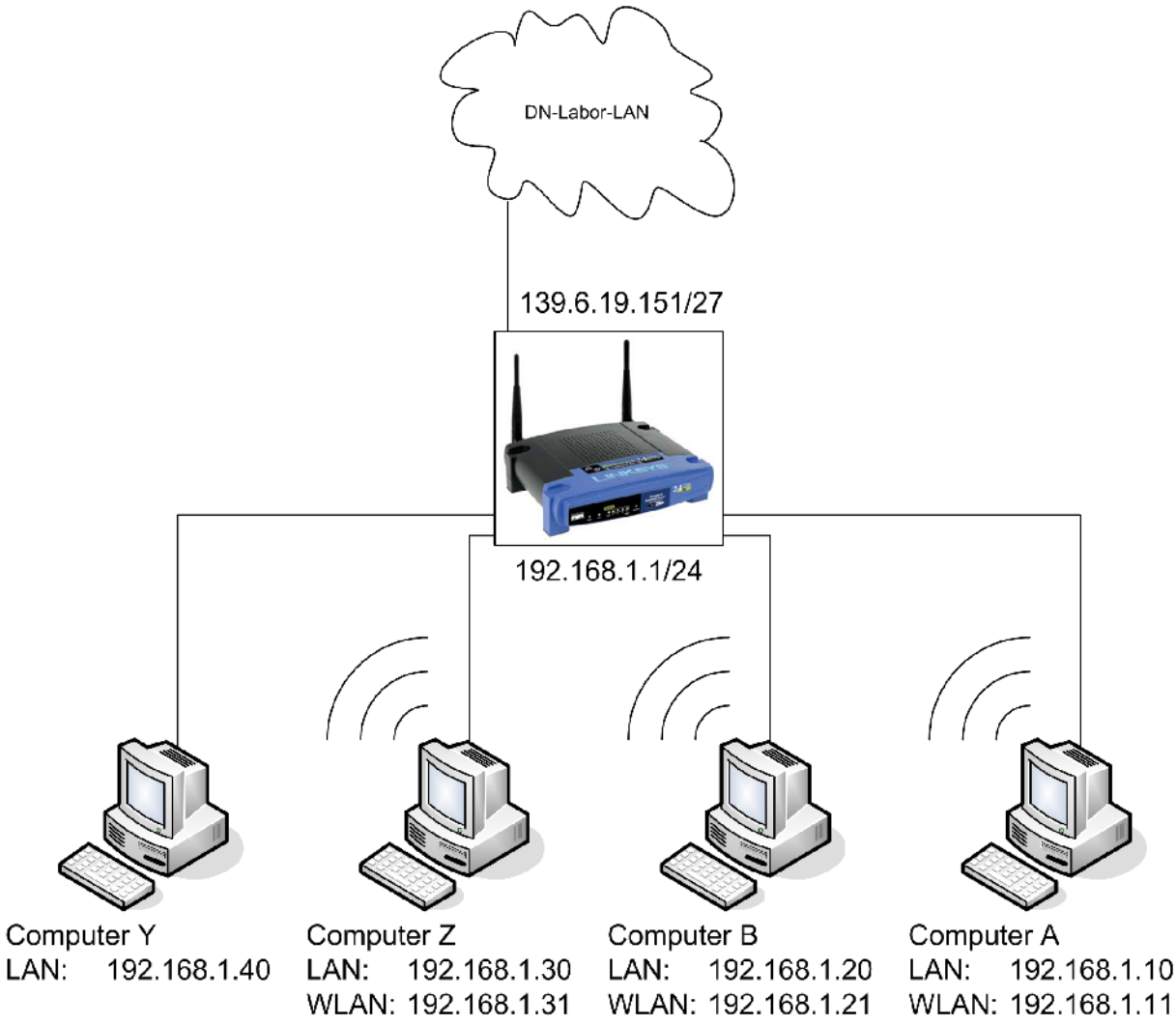


Abbildung 7: Übersicht der Testumgebung

In Anhang D, S.89 sind alle relevanten Daten der PCs wie zum Beispiel CPU, RAM, et cetera in einer Tabelle übersichtlich aufgeführt.

5 Test und Analyse der WMM-Funktionalität

In diesem Kapitel wird zunächst die für Messungen verwendete Software aufgezählt und erläutert. Darauf folgen die Messszenarien, die zugehörigen Messergebnisse, sowie Erläuterungen und Analysen.

5.1 Software für Messungen und Analyse

5.1.1 Ethereal

Der freie Netzwerk-Protokollanalysator Ethereal, inzwischen auch unter dem Namen Wireshark bekannt, kam bei diesem Projekt vor allem zur Analyse des gesamten Datenverkehrs im Empfangsbereich des WLANs zum Einsatz. Die WLAN-Karte eines Clients wurde hierfür in den so genannten Monitor-Mode geschaltet. Damit ist es möglich, jeglichen Datenverkehr der am Empfänger ankommt zu protokollieren. Im Client-Modus (Ad-Hoc- oder Infrastructure-Mode) ist dagegen nur die Protokollierung von WLAN-Paketen möglich, die genau an die Station gerichtet sind oder von dieser versandt werden, auf der der Datenverkehr mitgeschnitten wird. Im Monitor-Mode war es möglich, zu untersuchen, wie sich die verschiedenen Datenströme gegenseitig beeinflussen und wie sich WMM-Priorisierungen auswirken. Außerdem konnte überprüft werden, ob fremde WLANs auf der gleichen Frequenz im selben Empfangsbereich existieren.

Ethereal wurde in den Versionen 0.10.14 und 0.99.0 eingesetzt. Wesentliche funktionelle Unterschiede sind nicht aufgefallen. Die neuere Version wurde verwendet, weil bei der Installation von Ethereal auf Computer Z die Version 0.10.14 über apt-get nicht mehr verfügbar war.

5.1.2 Ping

Das einfache aber effektive Kommandozeilen-Tool „ping“ wurde nicht nur für die Kontrolle der Erreichbarkeit einer Gegenstelle genutzt. Vor allem wurden damit die ersten Tests durchgeführt, mit denen die WLAN-Karten und der Madwifi-Treiber auf WMM-Funktionalität hin überprüft wurden. Es war vor der Anschaffung, wie bereits erwähnt, nicht genau zu klären gewesen, ob WMM definitiv unterstützt wird.

Die Möglichkeit über optionale Parameter TOS-markierte ICMP-Echo-Request-Pakete zu versenden, wurde für oben genannten Test genutzt. Mit dem Befehl

```
# ping 192.168.1.1 -S 224
```

wurden zum Beispiel Datenpakete mit Priorität 7 von einem WLAN-Client an den Router geschickt. Auf einem anderen WLAN-Client wurden diese Datenpakete im Monitor-Mode mit Ethereal protokolliert.

Anmerkung: Unter Linux ist der Versand von Daten aus allen Anwendungen für normale Benutzer ohne Root-Rechte auf den maximalen TOS-Wert 159 beschränkt. Für eine TOS-Markierung größer oder gleich 160 werden zwingend Root-Rechte benötigt.

5.1.3 Iperf[30]

Für die Untersuchung von WMM und der Wirkung einer Priorisierung, wurde ein Tool benötigt, mit dem die Netzwerk-Performance gemessen werden kann. Hierfür sollten Datenstreams zwischen Endpunkten versandt werden können. Außerdem sollten die versandten Daten priorisierbar sein, die Bitrate der Streams einstellbar sein und eine genaue Auswertung ermöglicht werden.

Mit dem Tool Iperf ist genau dies möglich. Es erlaubt den Versand von TCP- und UDP-Streams zwischen Clients und Servern. Am Server werden die vom Client empfangenen Daten ausgewertet und die Messergebnisse ausgegeben. So lässt sich mit dem Tool die TCP- und UDP-Bandbreite, der Jitter der Verzögerung und der Paketverlust einer Verbindung messen. Hauptsächlich eingesetzt wurde das UDP-Verfahren, bei dem Iperf UDP-Pakete versendet, die beliebig gefüllt sind. Es wird kein auf UDP aufsetzendes Protokoll wie RTP verwendet. Trotzdem ähnelt die UDP-Übertragung am ehesten den Übertragungen von Multimedia-Daten. Sowohl bei VoIP als auch bei Video-Streams werden die Nutzdaten meist UDP-basiert übertragen. In der Regel haben diese Multimedia-Streams feste Datenraten. Die Bitrate des von Iperf erzeugten UDP-Streams ist entsprechend konfigurierbar. Ebenfalls einstellbar ist der TOS-Wert, mit dem der Stream markiert und versandt wird. Dies ist für die Untersuchung von WMM wichtig.

30 [19]; [20]; [21]

Im Folgenden werden die in den Messungen verwandten Parameter von Iperf genauer erläutert.

Optionen für Iperf-Server und -Clients:

- `-f, --format [bkmaBKMA]`:
  Einstellmöglichkeit der Einheit, in der die Ausgabe der Bandbreite erfolgt.
  Möglich sind:
  'b' = bits/sec 'B' = Bytes/sec
  'k' = Kbits/sec 'K' = Kbytes/sec
  'm' = Mbits/sec 'M' = Mbytes/sec
  'g' = Gbits/sec 'G' = Gbytes/sec
  'a' = adaptive bits/sec 'A' = adaptive Bytes/sec
- `-i, --interval <#>`:
  Angabe des Intervalls in Sekunden, nach dem jeweils die Messausgaben von Bandbreite, Jitter und Paketverlust erfolgen.
- `-u, --udp`:
  Versand von UDP- statt TCP-Paketen.
- `-B, --bind <host>`:
  Mit dieser Option wird der Versand bzw. der Empfang über das durch `<host>` angegebene Interface durchgeführt.

Optionen für Iperf-Server:

- `-s, --server`:
  Iperf im Server-Modus starten.

Optionen für Iperf-Clients:

- `-c, --client <host>`:
  Iperf als Client starten und Daten an `<host>` senden.
- `-b, --bandwidth <#>[KM]`:
  Angabe der UDP-Bandbreite in bit/s, mit der gesendet werden soll.
- `-t, --time <#>`:
  Dauer, für die gesendet werden soll, in Sekunden.

- `-S, --tos <#>`:

  Angabe des TOS-Wertes für die ausgehenden Pakete.

Während und nach der Durchführung des Streamings erfolgen auf Iperf-Client und -Server diverse Ausgaben mit Daten zur laufenden beziehungsweise abgeschlossenen Messung.

Die Ausgabe an einem Client sah zum Beispiel wie Folgt aus:

```
------------------------------------------------------------
Client connecting to 192.168.1.20, UDP port 5001
Binding to local address 192.168.1.11
Sending 1470 byte datagrams
UDP buffer size: 0.10 MByte (default)
------------------------------------------------------------
[  3] local 192.168.1.11 port 5001 connected with 192.168.1.20
port 5001
[  3]  0.0-10.0 sec  59.6 MBytes  50.0 Mbits/sec
[  3] 10.0-20.0 sec  59.7 MBytes  50.0 Mbits/sec
[  3] 20.0-30.0 sec  59.7 MBytes  50.0 Mbits/sec
[  3]  0.0-30.0 sec   179 MBytes  50.0 Mbits/sec
[  3] Sent 127645 datagrams
[  3] Server Report:
[  3]  0.0-30.0 sec  84.8 MBytes  23.7 Mbits/sec  0.873 ms
67189/127643 (53%)
```

Zu Beginn der Messung werden allgemeine Daten zur Verbindung ausgegeben, die teilweise aus den Startparametern resultieren. Im obigen Beispiel findet die Übertragung vom Client-Interface mit der IP 192.168.1.11 zum entfernten Iperf-Server mit der IP 192.168.1.20 auf Port 5001 statt.

Es werden Datenpakete mit 1470 Byte UDP-Daten versendet.

Während der Messung erfolgt im eingestellten Intervall die Ausgabe der verschickten UDP-Datenmenge. Zum Abschluss werden die Gesamtwerte der abgeschlossenen Messung angezeigt.

Unter „Server Report“ zeigt der Iperf-Client die Messergebnisse des Servers an, die dieser zuvor übermittelt hat. Ausgegeben wird die Dauer der Messung, die insgesamt empfangene UDP-Datenmenge und der mittlere Durchsatz der Messung (jeweils auf eine Nachkommastelle genau), der Jitter (mit drei Nachkommastellen) und die

Anzahl der verlorenen Datenpakete von wieviel maximal empfangbaren. In Klammern wird noch die Verlustrate in Prozent angegeben.

Am Iperf-Server, wo die eigentlichen Messungen durchgeführt werden, sieht die Ausgabe während der Laufzeit ähnlich aus. Sie ist jedoch in Bezug auf die Zeit etwas detailierter. Die gemessenen Werte werden zusätzlich zur Gesamtmessung pro Intervall ausgegeben.

Iperf wurde hauptsächlich in der Version 2.0.2 eingesetzt. Nur auf Computer Y wurde wegen des bereits vorinstallierten Debian-Systems (stable) die Iperf-Version 2.0.1 verwendet.

5.1.4 Shell-Skripte

Netzwerkkonfiguration

Um die Konfiguration der Netzwerkinterfaces in den Clients zu vereinfachen und um nach einer Änderung jeweils einen fest definierten Zustand zu erreichen, wurde ein Shell-Skript erstellt und für jeden PC angepasst. Über ein Auswahlmenü können die LAN- und WLAN-Interfaces deaktiviert oder aktiviert werden und IP-Konfigurationen vorgenommen werden. Außerdem sind Kombinationen über einzelne Menüpunkte schaltbar und der Monitor-Mode der WLAN-Karte ist aktivierbar. Als Beispiel befindet sich das lan.sh-Skript des Computers B im Anhang A, S.73.

Messstarts

Bei bestimmten Messungen war es nötig, auf mehreren PCs möglichst gleichzeitig Iperf mit dem Datenversand beginnen zu lassen. Hierfür wurde in die Skripte mess.sh und sync.sh ein Mechanismus eingebaut, der zuerst die PC-Uhrzeit mit einem Zeitserver abgleicht und dann zum Beginn der nächsten Minute Iperf startet. Für den Zeitabgleich über das NTP-Protokoll wurde ntpdate genutzt, das die aktuelle Uhrzeit vom Zeitserver der FH Köln[31] bezog. Damit konnte ein für die Messungen ausreichend synchronisierter Sendestart erreicht werden. Abweichungen von wenigen Millisekunden waren hierbei vernachlässigbar, da die so gestarteten Messungen über eine verhältnismäßig lange Dauer von 30 Sekunden durchgeführt wurden.

31 time.fh-koeln.de; nur aus dem internen Netzwerk der FH Köln zu erreichen

Mit sync.sh war eine einzelne synchronisierte Iperf-Messung möglich. Die Bandbreite des zu sendenden Streams und der TOS-Wert konnten direkt beim Aufruf als Optionen an das Skript übergeben werden. mess.sh erlaubte den automatisierten Ablauf einer ganzen Reihe von Messungen mit ansteigenden Sendebandbreiten. Durch Aufrufparameter konnte der TOS-Wert an das Skript übergeben werden.

5.2 Messungen

Um möglichst unverfälschte Messergebnisse zu erhalten, war es von besonderem Interesse, dass während der Messungen möglichst keine anderen WLANs den gleichen Kanal wie in der Testumgebung benutzen. Die Nutzung des gleichen Kanals oder der umliegenden 5 beziehungsweise 6 Kanäle würde Interferenzen bedeuten, die zu Störungen und somit falschen Messergebnissen führen würden.

Die Problematik besteht, weil das für 802.11b+g genutzte Frequenzband (2,4 bis 2,4835 GHz) in 13 Kanäle unterteilt ist, deren Center-Frequenzen jeweils nur 5 MHz voneinander entfernt sind. Da die genutzte Bandbreite jedes Kanals jedoch 22 MHz beträgt, bedeutet das, dass sich die Kanäle überlappen. Von der ETSI[32] wird außerdem gefordert, dass der Abstand zwischen den Center-Frequenzen zweier genutzter Kanäle mindestens 30 MHz beträgt. Wird dieser Forderung entsprochen, gibt es nur eine mögliche Kanalgruppierung, die 3 gleichzeitig genutzte Kanäle innerhalb eines Empfangsbereiches zulässt. Dies sind die Kanäle 1, 7 und 13. Die Grenzfrequenzen der Kanäle sowie die Lage zueinander kann Tabelle 4 und Abbildung 8 entnommen werden.[33]

Anmerkung: In den USA ist die WLAN-Nutzung nach 802.11b+g auf die Kanäle 1 bis 11 beschränkt. Laut der dort zuständigen FCC (Federal Communications Commission) brauchen die Center-Frequenzen genutzter Kanäle dort aber nur mindestens 25 MHz auseinander liegen. Somit sind die 3 unabhängig voneinander nutzbaren Kanäle dort 1, 6 und 11.

32 ETSI - European Telecommunications Standards Institute; gemeinnützige unabhängige Organisation, die Standards für Telekommunikationssysteme entwickelt

33 Dieser und der folgende Absatz basieren auf [1] S.94-95.

Kanal	*untere Grenzfrequenz in MHz*	*Center-Frequenz in MHz*	*obere Grenzfrequenz in MHz*
1	2401	2412	2423
2	2406	2417	2428
3	2411	2422	2433
4	2416	2427	2438
5	2421	2432	2443
6	2426	2437	2448
7	2431	2442	2453
8	2436	2447	2458
9	2441	2452	2463
10	2446	2457	2468
11	2451	2462	2473
12	2456	2467	2478
13	2461	2472	2483

Tabelle 4: Kanalbelegung von WLAN im 2,4 GHz-Band in Deutschland [1]

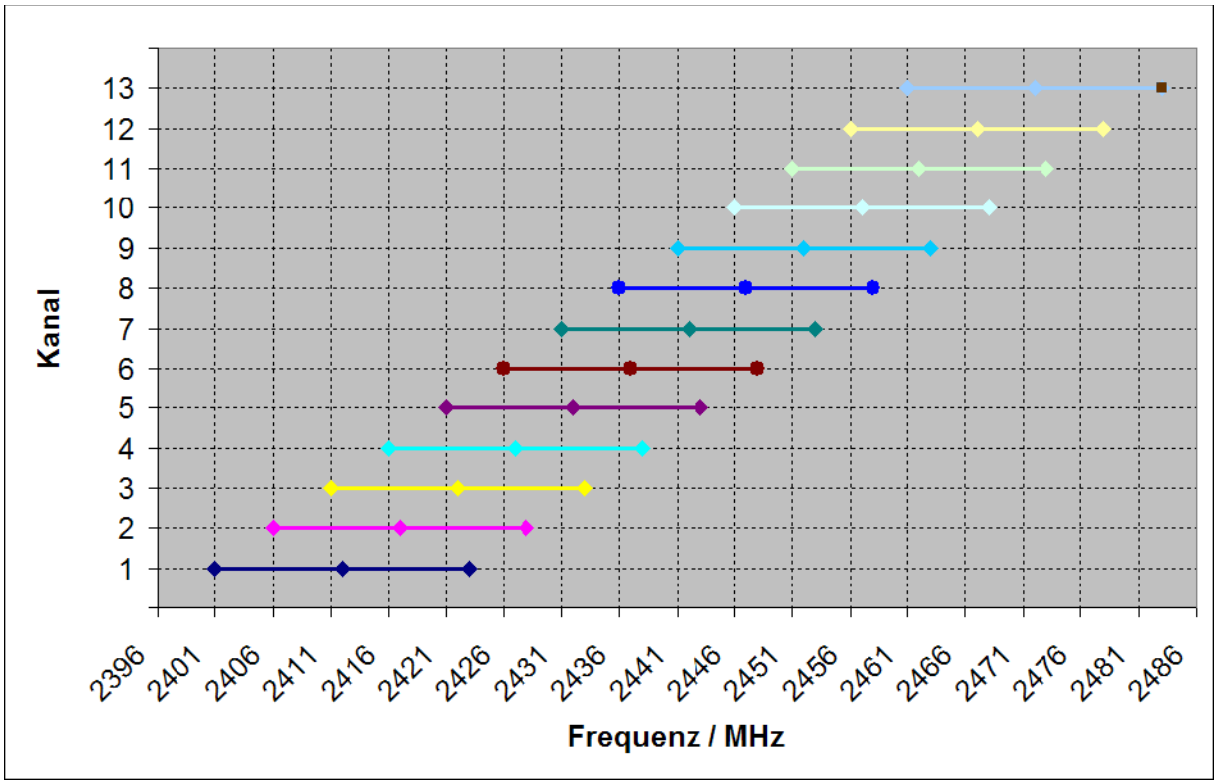


Abbildung 8: Kanalbelegung von WLAN im 2,4 GHz-Band in Deutschland

Um Messbeeinflussungen durch andere WLANs weitestgehend auszuschließen, wurde für den Betrieb der Testumgebung ein Kanal gewählt, der möglichst überlappungsfrei mit genutzten Kanälen war. Wie die Erfahrung zeigte, war die Frequenzbelegung des ISM-Bandes im Empfangsbereich des Datennetze-Labors im Verlaufe des Tages sehr unterschiedlich. Am geringsten war diese am späten Nachmittag und am Abend, weshalb die Messungen meist zu diesen Zeiten durchgeführt wurden. Kanal 1 war außerdem am wenigsten durch fremde WLANs gestört und wurde deshalb bei allen Messungen genutzt. Dass dieser Kanal zur exklusiven Nutzung bereit stand, wurde in unregelmäßigen Abständen an einer Station im Monitor-Mode überprüft. Ganz auszuschließen ist jedoch nicht, dass es während Messungen zu Störbeeinflussungen kam, da ein dauerhafter Monitor-Mode-Betrieb auf Grund der extrem großen aufgezeichneten Datenmengen nicht möglich war.

Während fremde WLANs meist keine Kanalwechsel durchführten und somit Störungen durch diese mit entsprechender Kanalwahl in der Testumgebung weitestgehend auszuschließen waren, gab es jedoch immer wieder WLAN-Geräte, die durch Probe-Requests gewisse Störungen verursachten. Da die Datenmenge solcher Probe-Requests und der zugehörigen Responses des AP der Testumgebung sehr gering sind, konnten diese Störungen bei den Messungen vernachlässigt werden. Wenn offensichtlich falsche Messergebnisse zustande kamen, wurde die jeweilige Messung wiederholt und versucht, den Fehler zu finden. Einige aufgetretene Probleme und Beobachtungen, die während der Messdurchführungen gemacht wurden, sind in Kapitel 5.2.5 aufgeführt. Im Folgenden nun die ersten Messungen und Analysen.

5.2.1 Mapping von TOS-Werten zu WMM-Kategorien

Wie bereits erwähnt, gibt es in WMM 4 Kategorien, denen jeweils 2 Prioritäten aus 802.1d zugeordnet werden (siehe Kapitel 2.3.2, speziell Tabelle 2). Wie sich bei der Analyse an der Luftschnittstelle herausstellte, werden die Pakete im WLAN-Header nicht nur mit 4 verschiedenen Prioritätswerten markiert, sondern mit 8. Es wird genau 1 Byte für die Markierung der QoS-Kategorie im WLAN verwendet. Dem entsprechend sollte die Umsetzung von 802.1d-Tags zu WMM-Prioritätswerten 1 zu 1 erfolgen. Für TOS-Werte sollte das gleiche gelten. Nur die Behandlung der Pakete

sollte entsprechend der Zugehörigkeit eines Prioritätswertes zu einer WMM-Kategorie erfolgen.

Soweit die Theorie, in der Praxis kam es jedoch bei mehreren Messungen zu unerwarteten Ergebnissen. Bei Durchsatzmessungen ohne Konkurrenzsituation (siehe Kapitel 5.2.2) zeigte sich beispielsweise, dass beim Versand von den WLAN-Clients über den AP zum Iperf-Server im LAN kein Unterschied zwischen Versand mit TOS-Wert 40 (Priorität 1) und TOS-Wert 160 (Priorität 5) bestand. Nur bei TOS-Wert 0 war ein deutlich geringerer Durchsatz gemessen worden.

Da die gleichen Messungen in der umgekehrten Richtung aus dem LAN über den AP zu den WLAN-Clients wiederum andere Ergebnisse brachten, die jedoch den Erwartungen entsprachen, wurde dies genauer untersucht.

Für das Verstehen der Versandbehandlung von den WLAN-Clients aus konnte der MadWifi-Quellcode analysiert werden. Trotz des großen Umfangs des Quellcodes gelang es, die relevanten Stellen zu finden, die für das Mapping verantwortlich sind. In der Datei ieee80211_output.c (relevanter Ausschnitt siehe Anhang B) wird in den Zeilen 137 bis 169 das Mapping der TOS-Werte zu WMM-Kategorien durchgeführt. Das Erste was an der Implementierung unlogisch erscheint, ist die Realisierung des Mappings mittels switch-case-Technik. Da auf das 8 Bit lange TOS-Feld zurückgegriffen wird, müssten hier eigentlich 256 Zeilen Code stehen, die nur für das Mapping aller möglichen TOS-Werte zu WMM-Kategorien nötig wären. Implementiert ist aber nur die Umsetzung von 8 irgendwie ausgewählten TOS-Feldern. Zusätzlich erscheint das Mapping dieser 8 TOS-Werte willkürlich. Die genaue Umsetzung ist Tabelle 5, S.47 zu entnehmen.

Um die Mapping-Umsetzung im Linksys-Router bei Datenverkehr aus dem LAN zu WLAN-Clients herauszufinden, wurde ein Skript (tosping.sh, siehe Anhang A) geschrieben, mit dem einzelne ICMP-Echo-Requests mit TOS-Priorisierung verschickt werden konnten. Das Mapping konnte anhand eines Ethereal-Mitschnitts im Monitor-Mode auf einem weiteren PC protokolliert werden. Es ergab sich folgende Mapping-Tabelle.

		WMM-Prioritätswert beim Versand	
Dezimal-Wert des 8-Bit-TOS-Feldes	***TOS-Präzedenz***	***von WRT54G***	***von MadWifi-Clients***
0-7	0	0	0 - ok
8	0	0	1
9-31	0	0	0 - ok
32	1	1	1 - ok
33-39	1	1	0
40	1	1	5
41-47	1	1	0
48	1	1	6
49-63	1	1	0
64-95	2	2	0
96-127	3	3	0
128-135	4	4	0
136	4	4	6
137-159	4	4	0
160	5	5	5 - ok
161-183	5	5	0
184	5	5	6
185-191	5	5	0
192-223	6	6	0
224	7	7	6
225-255	7	7	0

Tabelle 5: TOS-WMM-Mapping

Im MadWifi-Code wurden nur 8 TOS-Werte gemappt, davon aber nur 4 korrekt. TOS-Priorität 7 auf WMM-Prioritätswert 7 zu mappen, ist in der Implementierung des MadWifi-Codes gar nicht möglich.

In der Datei ieee80211_output.c geschieht wie erwähnt das Mapping von TOS-Feld zu WMM-Kategorie, nicht jedoch zu WMM-Prioritätswerten. Das Setzen des WMM-

Prioritätswertes geschieht erst in der Datei ieee80211h.c. Jeder der 4 WMM-Kategorien wird ein dort bestimmter Wert zugeordnet. Somit sind nur 4 verschiedene Wertezuordnungen möglich.

Grundsätzlich wäre dies ausreichend, da nur eine Unterscheidung zwischen den 4 WMM-Kategorien geschieht. Dies ändert jedoch nichts an der fehlerhaften Implementierung des TOS-WMM-Mappings im MadWifi-Code.

Für einen Kurztest wurde der MadWifi-Code in Datei ieee80211h.c mit verschachtelten IF-Abfragen so angepasst, dass zumindest das Mapping von TOS zu WMM-Kategorie für alle 256 möglichen TOS-Werte korrekt geschah.

Wie ein TOS-markiertes Ping-Paket bei Versand mit WMM aussieht, ist im Ethereal-Mitschnitt in Abbildung 9 zu sehen. Aufgeklappt ist dort der 802.11-Header des Frames. Versendet wurde der Ping von Computer Z im WLAN an B im LAN, was anhand der Source- und Destination-Address sichtbar ist. Ein Teil des Headers ist mit „QoS parameters“ bezeichnet. Dort sieht man den Prioritätswert 6, der der Einordnung in WMM-Kategorie VO entspricht. Direkt darunter befinden sich die TKIP-Parameter der WPA-Verschlüsselung. Darauf folgen die eigentlichen (verschlüsselten) Nutzdaten.

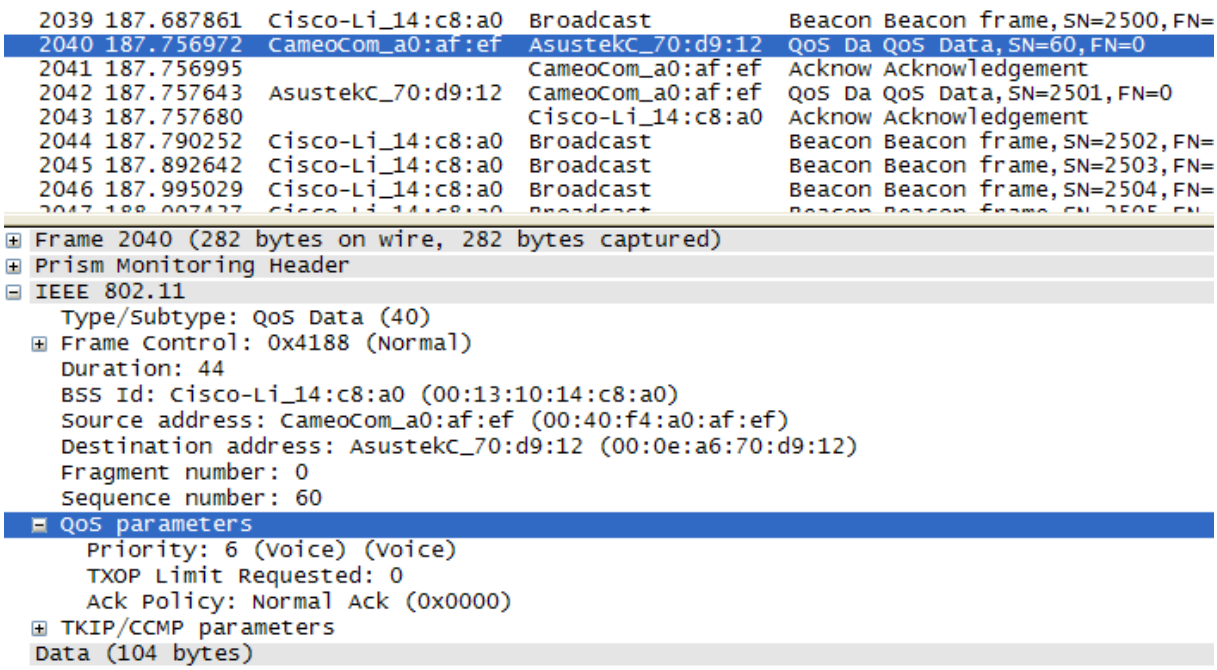


Abbildung 9: Datenpaket mit WMM-Markierung

In diesem Kapitel wurde nur das Mapping betrachtet. Ob die Priorisierung der einzelnen WMM-Kategorien korrekt ist, kann unter anderem Kapitel 5.2.3, S.56ff entnommen werden.

5.2.2 Durchsatzmessungen ohne Konkurrenzsituation

Solange nur ein einziger Client im WLAN am AP betrieben wird, ist ein QoS-Mechanismus wie WMM eigentlich nicht nötig, da es beim Medienzugriff keine Konkurrenz zu anderen Stationen gibt. Einzig im Client selbst kann es Konkurrenzsituationen zwischen verschiedenen Datenströmen geben. Diese Client-interne Priorisierung von WMM wird in Kapitel 5.2.4 behandelt.

In diesem Kapitel war von Interesse, ob und wie sich der Datendurchsatz beim Betrieb mit WMM und ohne WMM unterscheidet. Außerdem sollten Messungen zeigen, ob bei diesen Szenarien Unterschiede zwischen Up- und Downstream zwischen AP und Clients existieren.

Ohne WMM

Im AP wurde WMM deaktiviert. Dies wirkt sich unmittelbar auf alle Clients aus, die diesen AP nutzen, da durch die Beacon-Frames die Fähigkeiten des AP den Clients mitgeteilt werden und diese ihr Verhalten entsprechend anpassen.

Ein Computer befindet sich als Messendpunkt im LAN am WRT54G. Ein weiterer Computer ist der Messendpunkt im WLAN. Gemessen wurden nacheinander die Datendurchsätze in beide Richtungen, also einerseits von der WLAN-Station mit Iperf-Client über den AP zum Iperf-Server im LAN und andererseits vom Iperf-Client im LAN über den AP zum Iperf-Server im WLAN. Mit diesen Konstellationen kann die Sendeperformance der Clients und des AP jeweils für sich festgestellt werden.

Wie schon erwähnt, wurden alle Messungen im „G-only“-Modus durchgeführt. Außerdem war die Verschlüsselung mit WPA-PSK TKIP[34] immer aktiviert. Bei 802.11g können Daten bekanntlich mit maximal 54 MBit/s übertragen werden. Durch den gesamten Overhead kann der in den Messungen verwendete UDP-Datenstrom jedoch unmöglich diesen Durchsatz erzielen. Bei den meisten Messungen wurden 1470 Byte an UDP-Nutzdaten pro Datenpaket vom Iperf-Client versendet, was der Standardeinstellung in Iperf entspricht. Hinzu kamen 8 Byte für den UDP-Header, 20 Byte für den IPv4-Header und 34 Byte für den 802.11-MAC-Header sowie 4 Byte für die Frame Check Sequence am Ende des MAC-Frames. Diese Daten werden im PHY-Layer mit maximal 54 MBit/s übertragen. Auf Schicht 1 werden jedoch vorher außerdem noch Synchronisationsdaten (PLCP[35]-Preamble), der PLCP-Header und OFDM[36]-Synchronisationsdaten übertragen, die teilweise mit 1 MBit/s, teilweise mit 2 MBit/s übertragen werden (→ Anhang D, S.88). Das Ende der Übertragung eines Datenframes wird auf ein volles OFDM-Symbol aufgerundet und abschließend folgt noch eine OFDM Signalerweiterung von 6 Mikrosekunden. Darauf folgt dann der SIFS und das (802.11-)ACK des Empfängers. Erst nach einer weiteren Wartezeit bestehend aus IFS und Backoff-Zeit (siehe Kapitel 2.1.3) erfolgt die Übertragung des nächsten Frames.[37]

Aus diesem gesamten Daten- und Zeitoverhead der Protokolle und des Übertragungsverfahrens resultiert ein wesentlich geringerer UDP-Nutzdatendurchsatz als die 54 MBit/s des 802.11g-Standards.

34 WPA – Wi-Fi Protected Access; PSK – Preshared Key; TKIP – Temporal Key Integrity Protocol witerführende Informationen u.a. auf http://www.wi-fi.org/OpenSection/white_papers.php

35 PLCP – Physical Layer Convergence Protocol

36 OFDM – Orthogonal Frequency Division Multiplexing (das bei 802.11g genutzte Übertragungsverfahren)

37 [1]; [22]

Eine Durchsatzmessung ohne WMM wurde unter anderem von Computer A im LAN zu Computer B im WLAN durchgeführt. Hierfür wurde auf Computer B der Iperf-Server mit Root-Rechten über den Befehl

```
# iperf -f m -i 10 -u -s -B 192.168.1.21
```

gestartet. Von Computer A wurden dann UDP-Streams verschiedener Bandbreiten und mit verschiedenen TOS-Werten verschickt. Ebenfalls mit Root-Rechten mittels

```
# iperf -f m -i 10 -u -B 192.168.1.10 -t 30
-c 192.168.1.21 -b 25000000 -S 224
```

wurde beispielsweise 30 Sekunden lang ein Stream von 25 MBit/s UDP-Daten mit dem TOS-Wert 224 von Computer A (IP: 192.168.1.10) an Computer B (IP: 192.168.1.21) verschickt.

Die Ergebnisse, die sowohl am Iperf-Server als auch am Iperf-Client ausgegeben werden, wurden für alle durchgeführten Messungen in einem Tabellenkalkulations-Programm protokolliert und anschließend ausgewertet.

Wie nicht anders zu erwarten war, hatte die Priorisierung mittels TOS bei den Messungen ohne WMM keinerlei Auswirkungen. Der maximale Durchsatz war bei allen gemessenen TOS-Werten bis auf geringste Abweichungen gleich. Bei Verlusten von weniger als 1% waren vom LAN ins WLAN 28 MBit/s an UDP-Daten übertragbar. In Abbildung 10 ist der Knick in den Graphen an dieser Stelle gut zu erkennen. Bis zum Messpunkt bei 28 MBit/s verlaufen die Graphen absolut linear. Der Jitter lag immer unter 1ms.

Wie bei fast allen Messungen ist in dem jeweiligen Diagramm auf der X-Achse der Versand von UDP-Daten vom Iperf-Client in MBit/s aufgetragen und auf der Y-Achse der Empfang in MBit/s am Iperf-Server. Die Differenz ist in der Regel auf Grund der begrenzten Bandbreite im WLAN beim WLAN-Sender verworfen worden. Geringe Verluste können auch durch Kollisionen bei der Übertragung auftreten, die aber normalerweise durch den Acknowledgement-Mechanismus im WLAN verhindert werden. Nur wenn nach mehrfacher Wiederholung eine Übertragung nicht erfolgreich war, wird der Versand am Sender abgebrochen.

In jedem Messdiagramm ist jeweils eine Legende zu den Graphen angegeben. In der Regel ist dort der TOS-Wert vermerkt, mit dem die Daten versandt wurden. Bei bestimmten Messungen ist zusätzlich die Buchstabenkennung des Computers angegeben, für den der Graph gilt.

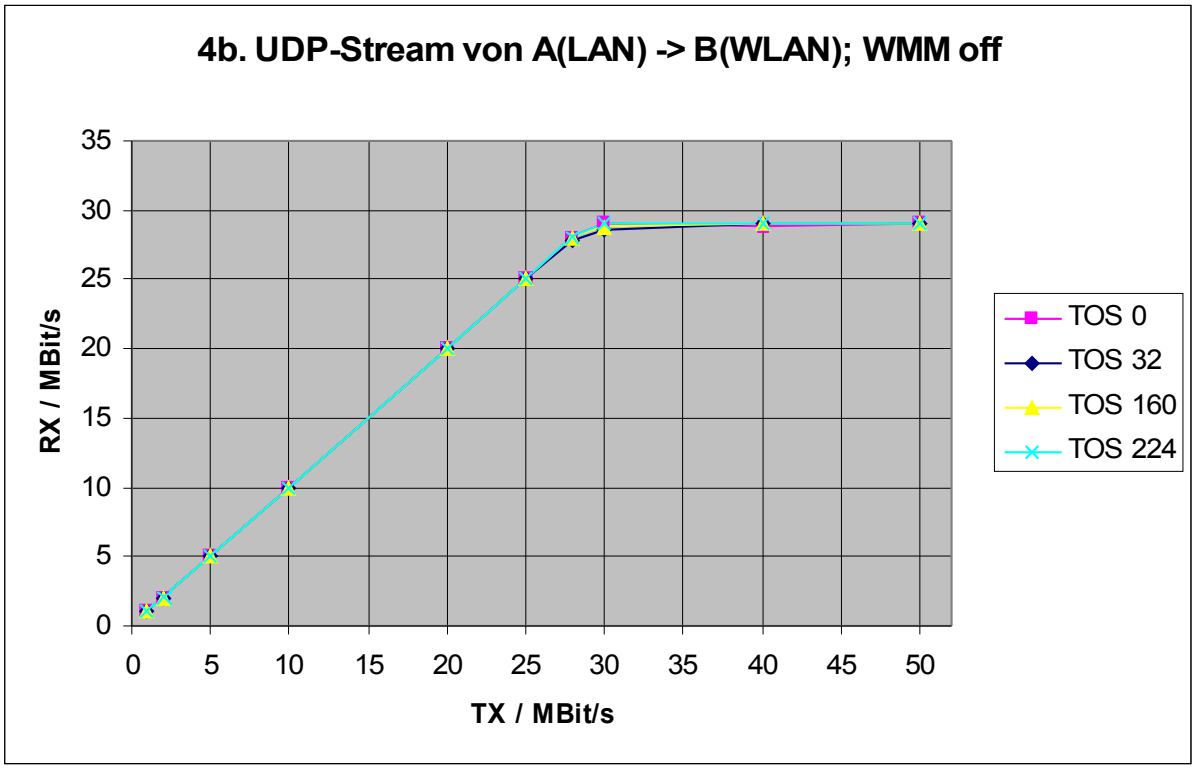


Abbildung 10: Diagramm 4b

In der umgekehrten Richtung vom WLAN ins LAN wurden bei deaktiviertem WMM erheblich höhere Durchsätze von 33 bis 34 MBit/s gemessen. Dies war so nicht zu erwarten gewesen, da eigentlich die erzielbaren Durchsätze in beiden Richtungen ähnlich sein sollten. Eine mögliche Erklärung für dieses Ergebnis könnte die unterschiedliche Hardware in AP und Computer sein. Einerseits sind die Funkmodule in AP und Computern unterschiedlich und andererseits steht auf den Computern eine wesentlich höhere Rechenkapazität als im WRT54G zur Verfügung. Außerdem kann es auch Implementierungsgründe haben, die gegebenenfalls in der Software der Geräte zu suchen sind.

Mit WMM

Beim Betrieb des WLANs mit aktiviertem WMM war zu erwarten, dass andere Durchsätze gemessen werden als ohne WMM. Vor dem Senden nach Erkennung

des freien Mediums wird ohne WMM wie erwähnt DIFS plus Backoff-Zeit lange gewartet. Bei WMM wird DIFS je nach Kategorie durch eine kürzere oder längere Zeit (AIFSN) ersetzt. Außerdem wird das CW, aus dem die Länge der Backoff-Zeit gewählt wird, bei WMM geändert. (siehe Abbildung 4, S.25)

Aus einer kürzeren Wartezeit, in der das Medium ungenutzt ist, resultiert automatisch ein höherer erzielbarer Datendurchsatz. Dies trifft auf die Kategorien VI und VO zu.

In WMM-Kategorie BE dürfte auf Grund der um einen einzigen Slot längeren Wartezeit als im Betrieb ohne WMM ein minimal, wenn überhaupt messbar, geringerer Datendurchsatz resultieren.

Beim Versand in WMM-Kategorie BK hingegen wird eine wesentlich längere Dauer gewartet (AIFS[BK]) als ohne WMM. Hier müsste also ein geringerer Durchsatz gemessen werden.

Die durchgeführten Messungen bestätigten die Erwartungen. Sehr deutlich wird dies in Abbildung 11, die das Äquivalent zu obiger Abbildung 10 ist, diesmal jedoch mit aktiviertem WMM.

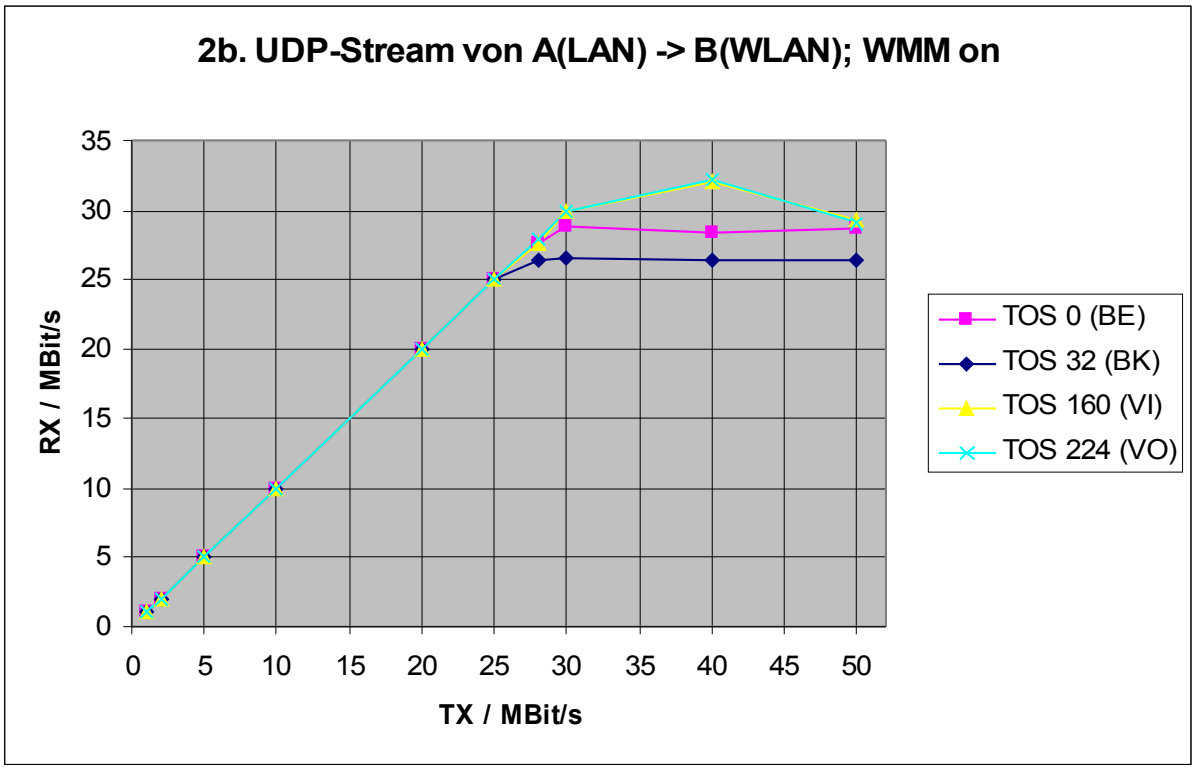


Abbildung 11: Diagramm 2b

Wie erwartet ist der Graph des Streams mit TOS-Wert 0 (WMM-Kategorie BE) sehr ähnlich zu den Graphen (aller TOS-Werte) ohne WMM. Der maximale Durchsatz liegt wiederum bei etwa 28 MBit/s aus dem LAN über den AP zum WLAN-Client.

Der Durchsatz in der BK-Kategorie liegt über 5 Prozent darunter bei etwa 26,5 MBit/s.

Streams der Kategorien VI und VO erzielen einen Maximaldurchsatz von etwa 32,5 MBit/s. Bei Paketverlusten von kleiner gleich 1 Prozent liegt der Durchsatz in VI und VO bei etwa 30 MBit/s und damit klar über BE. Auffallend ist erstens dass es keinen messbaren Unterschied zwischen VI- und VO-Streams gibt. Zweitens ist der Messeinbruch beim Versand von 50 MBit/s etwas verwunderlich. Ein Erklärungsversuch für dieses Phänomen findet sich in Kapitel 5.2.4 (→ WRT54G – Mit WMM), wo Messungen mit ähnlichem Effekt analysiert werden. Die Jitter-Werte sind mit (teilweise deutlich) unter 1ms uninteressant, da besonders positiv.

Soweit die Betrachtung der Richtung LAN → AP → WLAN. In der umgekehrten Richtung sind die Ergebnisse nicht ganz so erwartungsgemäß.

Zur Erinnerung: Ohne WMM war vom WLAN-Client ins LAN ein Durchsatz von 33 bis 34 MBit/s gemessen worden. Wie in Abbildung 12 zu sehen, liegt der Durchsatz in Kategorie BE (TOS 0) mit etwa 26 MBit/s nicht annähernd so hoch. Er liegt nicht mal so hoch wie in der umgekehrten Richtung (28 MBit/s sowohl bei BE mit WMM als auch ohne WMM).

Der BK-Stream liegt noch weiter darunter bei etwa 24 MBit/s.

Während die Kategorien BE und BK somit bei der Richtung WLAN → LAN schlechter abschneiden als in der umgekehrten Richtung, ergibt sich für die Kategorien VI und VO mit jeweils über 34 MBit/s ein gegenteiliges Bild. Der Durchsatz ist dort minimal größer als in der Messung ohne WMM. Die Jitter-Zeiten liegen mit WMM wie gehabt meist bei unter 1ms. Lediglich bei zwei 50 Mbit/s-Messungen wurden Werte von 14ms gemessen.

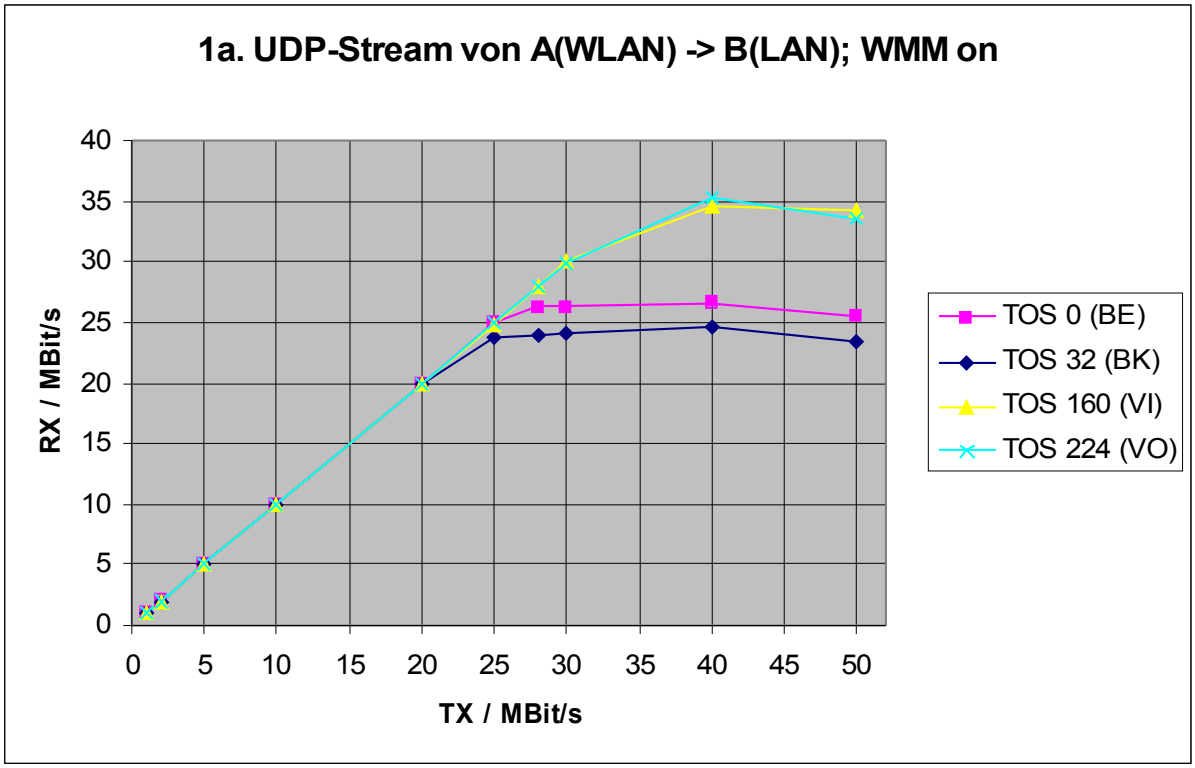


Abbildung 12: Diagramm 1a

Insgesamt betrachtet macht der WLAN-Teil des WRT54G-Routers einen wesentlich ausgeglicheneren Eindruck im Sendeverhalten. Der Maximaldurchsatz aller bis hier erwähnten Messungen liegt zwar bei den WLAN-Karten der Clients mit dem MadWifi-

Treiber höher, jedoch entspricht das Sendeverhalten in der Praxis nicht in dem Maße der Theorie, wie es sein sollte.

Dass der MadWifi-Treiber beziehungsweise die Allnet-WLAN-Karten im Sendebetrieb ohne WMM so hohe Durchsatzraten erzielen ist positiv. Dass aber im WMM-Betrieb die theoretisch mögliche Leistungssteigerung bei den Kategorien VI und VO nicht umgesetzt wird und die Durchsätze in den Kategorien BE und BK im Verhältnis sehr niedrig liegen, lässt auf Fehler in der Treiber-Software schließen. Vor dem Hintergrund der bereits erwähnten fehlerhaften Implementierung von WMM im MadWifi-Code (siehe Kapitel 5.2.1, S.45) ist dies recht wahrscheinlich. An der Hardware kann es schlicht deshalb nicht liegen, da im Betrieb ohne WMM hohe Datenraten möglich sind. Was theoretisch noch denkbar wäre, ist die nicht korrekte Implementierung des Nicht-WMM-Betriebs. Dies würde die besonders hohen Datenraten erklären und auch den Grund dafür liefern, dass bei WMM-VI und -VO keine Leistungssteigerung mehr möglich ist. Dies sind jedoch nur Vermutungen, die nicht nachgewiesen werden konnten.

Die Messungen, die in diesem Kapitel mit nur einem Client im WLAN durchgeführt wurden, werden im folgenden Kapitel nun mit Konkurrenz zwischen zwei Clients betrachtet.

5.2.3 Durchsatzmessungen mit Konkurrenzsituation

Da sich bei einem Shared Medium wie dem Funkfeld die gesamte zur Verfügung stehende Bandbreite auf alle Geräte aufteilt, steht jedem Einzelnen umso weniger Bandbreite zur Verfügung, je mehr WLAN-Geräte die gleichen Frequenzen innerhalb eines Empfangsbereiches nutzen. Theoretisch steht jedem dieser WLAN-Geräte eine Bandbreite von $\frac{Bandbreite_{gesamt}}{Anzahl\ Nutzer}$ zur Verfügung. Da es im WLAN aber bei einer größeren Anzahl von WLAN-Geräten eher zu Kollisionen bei der Übertragung kommt, verringert sich die effektive Bandbreite weiter. Die höhere Kollisionsgefahr besteht, weil es bei vielen Nutzern wahrscheinlicher ist, dass zwei zufällig die gleiche Backoff-Zeit warten und zeitgleich beginnen zu senden.

Gerade bei diesem Konkurrenzszenario wird der Einsatz von WMM interessant. Doch zunächst die Messungen ohne WMM.

Ohne WMM

Für die Analyse der Konkurrenzsituationen wurde im LAN ein Iperf-Server installiert. Im WLAN wurden auf zwei Computern Iperf-Clients gestartet, die zeitgleich mit der Übertragung von UDP-Streams begannen. Der synchronisierte Start beider Iperf-Clients wurde gewählt, um nach einem kurzen Einschwingvorgang eine 30-sekündige Messung im „stabilen Zustand“ zu erreichen. Mögliche Störungen betreffen so außerdem beide Clients gleich. Bei einem versetzten Start wäre es möglich, dass ein Client durch Störungen alleine beeinflusst würde und es so zu verfälschten Messergebnissen käme.

Zu erwarten war beim Betrieb ohne WMM, dass unabhängig vom TOS-Wert des jeweiligen Streams beide Clients immer den gleichen Datendurchsatz erreichen, da sich die Gesamtbandbreite des WLANs auf beide Clients jeweils zur Hälfte aufteilen sollte. Bei einer Überlastung des WLANs müsste somit bei beiden Streams der gleiche Datenverlust auftreten.

Wie in Abbildung 13 zu sehen ist, entspricht das Messergebnis relativ genau der Erwartung. Trotz unterschiedlicher TOS-Werte von 0 beziehungsweise 224 gibt es keine erwähnenswerte Priorisierung eines der beiden Streams. Beide werden mit der gleichen Bandbreite übertragen und die Verluste sind bei beiden Streams ähnlich. Beide Bandbreiten zusammengerechnet ergeben jedoch nicht genau den maximal gemessenen Durchsatz eines einzelnen Client vom WLAN ins LAN ohne WMM. Dies resultiert daraus, dass möglicherweise Kollisionen aufgetreten sind und außerdem bei zwei WLAN-Clients die doppelte Anzahl an ACKs versendet werden mussten. Die gemessenen Jitter-Zeiten lagen meist wieder um 1ms. Nur bei Messungen, wo über 40% Datenverlust auftrat, kam es vereinzelt zu Jitter-Werten von etwa 14ms.

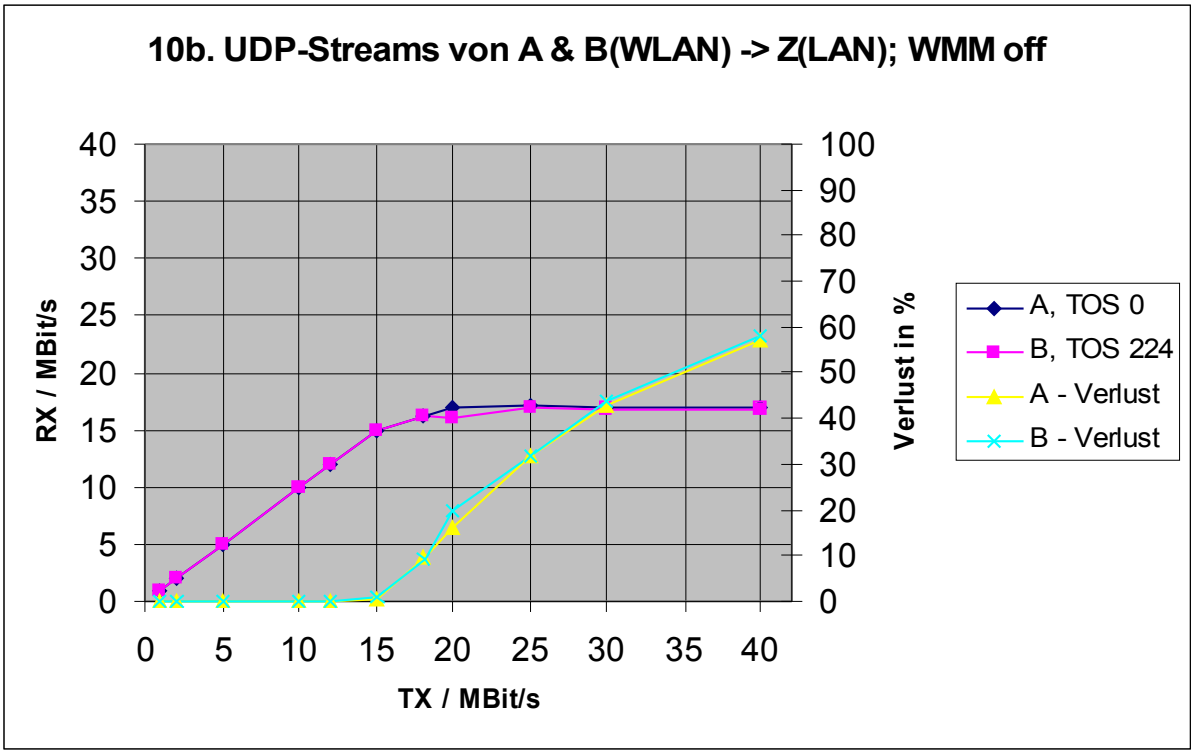


Abbildung 13: Diagramm 10b

Mit WMM

Die Erwartung an die Messungen bei Konkurrenz mit WMM waren eine deutliche Sichtbarkeit der Priorisierungen im Messdiagramm. Streams der Kategorien VI (hier mit TOS-Wert 160) und VO (TOS-Wert 224) sollten auch dann noch verlustfrei übertragen werden, wenn ein BE-Stream (TOS-Wert 0) versucht, die gesamte WLAN-Bandbreite zu belegen. Der BE-Stream sollte in diesem Fall soweit zurückgedrängt werden, dass VI und VO verlustfrei übertragen werden können. Für den BE-Stream bedeutet das zwar Paketverluste, dies ist dabei aber so gewollt.

Für die Messungen wurden möglichst viele Kombinationen priorisierter Streams von den WLAN-Clients A und B (Iperf-Clients) über den AP zum Iperf-Server Z im LAN gesendet. Für jede Kombination wurden wiederum diverse Sendebandbreiten gemessen und die Messergebnisse am Iperf-Server protokolliert und ausgewertet. Alleine für dieses Unterkapitel wurden über 80 Messungen durchgeführt.

Die Messergebnisse entsprechen den Erwartungen und zeigen, dass die Priorisierung mittels TOS und WMM gut funktioniert. Im Folgenden einige Beispiele in Form von Diagrammen und dazugehörigen Erläuterungen:

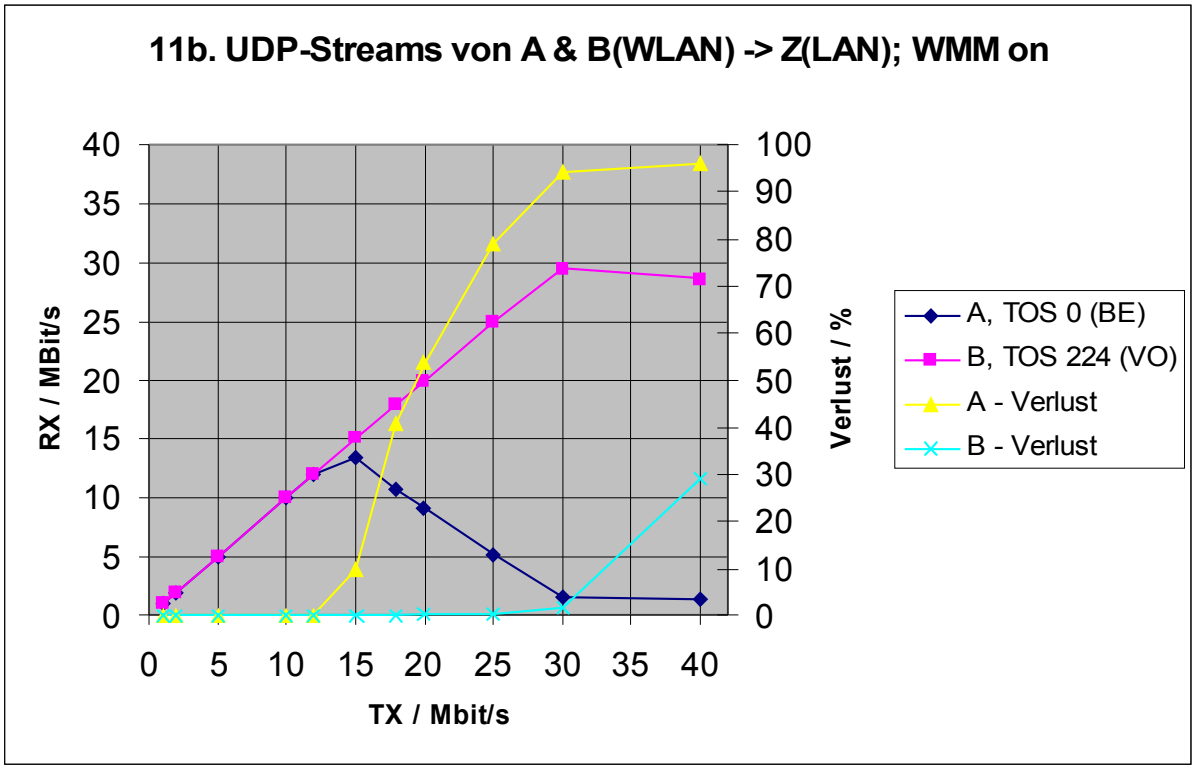


Abbildung 14: Diagramm 11b

Abbildung 14 zeigt die Messergebnisse zweier konkurrierender Streams mit unterschiedlicher Priorisierung. Von Computer A wurde ein Stream mit TOS-Wert 0 abgeschickt, der in die WMM-Kategorie BE eingeordnet wurde. Der Iperf-Client auf Computer B versendete einen Stream mit TOS-Wert 224, der somit die Priorität der WMM-Kategorie VO bekam. Von beiden Clients wurden jeweils UDP-Streams mit der gleichen Bandbreite abgeschickt.

Bis zu 2*12 MBit/s tritt keinerlei Paketverlust auf. Beim Versand von 2*15 MBit/s sind jedoch bei dem BE-Stream die ersten Verluste zu verzeichnen. Dessen Durchsatz-Graph hat an dieser Stelle einen starken Knick. Während die Bandbreite des VO-Streams bis knapp unter die 30 MBit/s-Grenze annähernd verlustfrei steigerbar ist, sinkt die am Iperf-Server empfangene Datenmenge des niedrig priorisierten Streams kontinuierlich bis auf etwa 1,5 MBit/s ab. Interessant ist, dass sich die Empfangsraten bei weiterer Steigerung der versendeten Datenmengen nicht mehr stark verändern.

Zu erwarten war zwar auf Grund des Designs des WLAN-Standards (Backoff-Zeit wird kontinuierlich heruntergezählt), dass der BE-Stream eine gewisse Mindestbandbreite immer nutzen kann. Wie hoch diese aber sein würde, war nicht einzuschätzen. 1,5 MBit/s ist ein Wert, der auf den ersten Blick nicht besonders hoch erscheint, aber für viele Anwendungen doch ausreichend ist. Ein VoIP-Telefonat beispielsweise benötigt einen Bruchteil dessen. Auch zum normalen „Webbrowsen“ ist diese Bandbreite absolut akzeptabel.

Da man vermuten könnte, dass diese Mindestbandbreite des BE-Streams nur deshalb besteht, weil vom Iperf-Client vergleichsweise hohe Datenraten von 30 beziehungsweise 40 MBit/s verschickt wurden, wurde an dieser Stelle eine Sondermessung durchgeführt. Der versandte BE-Stream wurde auf 1 MBit/s beschränkt, der VO-Stream hatte weiterhin 40 MBit/s. Das Ergebnis war ein verlustfrei übertragener BE-Stream. Der VO-Stream veränderte sich nicht. Somit kann festgehalten werden, dass eine Mindestbandbreite bei dieser Konstellation immer zur Verfügung steht. Ohne WMM waren die Ergebnisse der Sondermessung übrigens genauso wie mit WMM. Die Aussage zur Mindestbandbreite kann man jedoch nicht auf Szenarien übertragen, bei denen sich sehr viele Clients im gleichen WLAN befinden. Da sich die Bandbreite bei gleicher Priorisierung auf jeden Client annähernd gleichmäßig aufteilt, dürfte die Mindestbandbreite bei über 30 Clients auch unter 1 MBit/s sinken.

Für die gemessenen Jitter-Werte gilt weiterhin die Aussage: meist um oder unter 1ms. Nur bei Streams mit hohen Paketverlusten steigt der Jitter-Wert auf bis zu 20ms. Dies ist auch verständlich, da hohe Verluste gerade bei den Streams auftreten, wo immer wieder besonders lange auf den Medienzugriff gewartet werden muss. Die Wartezeit schwankt dadurch auch sehr stark, was sich direkt im Jitter-Wert niederschlägt.

Mit WMM wurden weitere Kombinationen von unterschiedlichen Priorisierungen gemessen. Interessant war unter anderem die Funktionsprüfung der BK-Kategorie. Traffic in dieser WMM-Kategorie sollte eine noch geringere Priorität haben als BE. Zur Überprüfung wurde aus dem WLAN von Computer A ein BE-Stream und von Computer B ein BK-Stream an den Iperf-Server Z im LAN versendet.

Das Ergebnis ist in Abbildung Abbildung 15 zu sehen. Der Unterschied zwischen den Streams wird sehr deutlich. Genau wie erwünscht steht dem BE-Stream eine höhere Bandbreite zur Verfügung als dem BK-Stream. Deutlich wird aber auch, dass diese nicht extrem hoch ist. In diesem Zusammenspiel liegen die maximal möglichen Datenraten bei etwa 18+8 MBit/s. Die Summe von 26 MBit/s liegt wesentlich unter der Gesamtbandbreite von 30 MBit/s, die bei Konstellation 11b (Abbildung 14) gemessen wurde. Sieht man sich die Durchsatzmessung ohne Konkurrenz (Abbildung 12) an, so wird klar, dass gar nicht mehr möglich ist. Dort lag der Maximaldurchsatz eines einzelnen BE-Streams ebenfalls nur bei knapp 26 MBit/s. Die Ursache ist ganz klar das WMM-Timing, also die Wartezeiten vor dem Versand von Datenpaketen (siehe Kapitel 2.3.2, S.22ff und Abbildung 4, S.25).

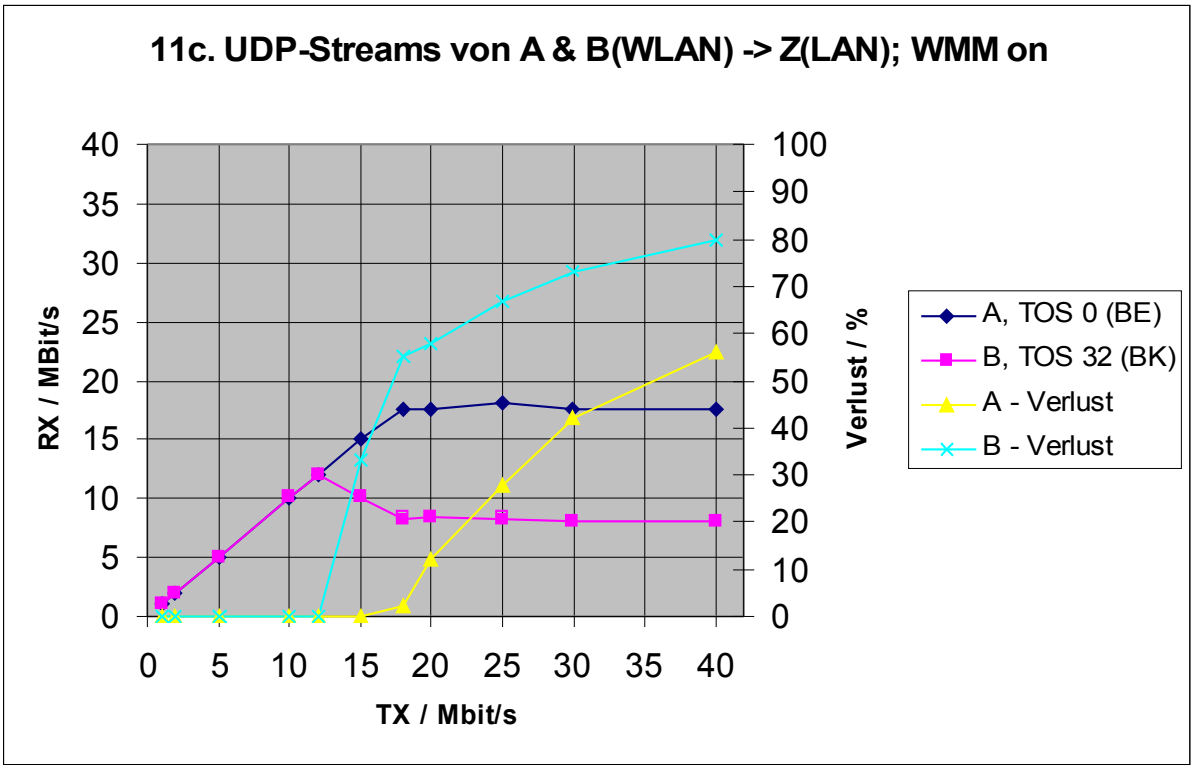


Abbildung 15: Diagramm 11c

Eine weitere Beobachtung bezüglich der BK-Streams (Diagramme 11f und 11h im Anhang C) war, dass diese bei Konkurrenz mit einem VI- oder VO-Stream bei Überlast noch stärker zurückgedrängt werden als BE-Streams. Eine Mindestbandbreite konnte für BK-Streams nicht festgestellt werden. Problematisch ist dies jedoch nicht, da in der BK-Kategorie ohnehin nur Daten übertragen werden sollten, die völlig zeitunkritisch sind. Bei einer Datensicherung beispielsweise, die

über Netzwerk (hier WLAN) durchgeführt wird und mit BK-Klassifizierung läuft, ist es nicht von Belang, ob diese einige Zeit früher oder später fertiggestellt ist.

Bei den bis zu diesem Punkt vorgestellten Messergebnissen ist zu sehen gewesen, dass die Priorisierung von Datenübertragungen mittels WMM im WLAN gut funktioniert. Es gibt jedoch auch Beispiele, wo WMM nicht optimal arbeitet. Beispielsweise konnte bei Konkurrenz zwischen einem VI- und einem VO-Stream kein Unterschied in der Priorisierung gemessen werden. In Abbildung 16 ist diese Konstellation zu sehen. Gut erkennen kann man die schwankenden Graphen oberhalb der 15 MBit/s-TX-Marke. Mal ist der Durchsatz des VI-Streams besser, mal der des VO-Streams. Es sieht aus, als sei der jeweilige Vorteil rein zufällig. Ein Blick auf Abbildung 4, S.25 macht deutlich, dass dies relativ genau so gewollt ist. Die minimale Wartezeit vor dem Versand eines Frames ist bei den WMM-Kategorien VI und VO mit 2 Slots genau gleich lang. Nur das Fenster, aus dem die zufällige Backoff-Wartezeit gewählt wird, ist unterschiedlich. Die Backoff-Zeit sollte im Initialzustand bei VO im Mittel 1,5 Slots lang sein, bei VI 3,5 Slots. Theoretisch wäre eine Priorisierung der VO-Kategorie somit gegeben. Mit den verwendeten Allnet-WLAN-Karten mit dem MadWifi-Treiber war diese, wie man sieht, jedoch nicht messbar. Da auch schon bei den Durchsatzmessungen ohne Konkurrenzsituation keine Unterschiede zwischen den beiden Kategorien VO und VI messbar waren, decken sich hier die Ergebnisse.

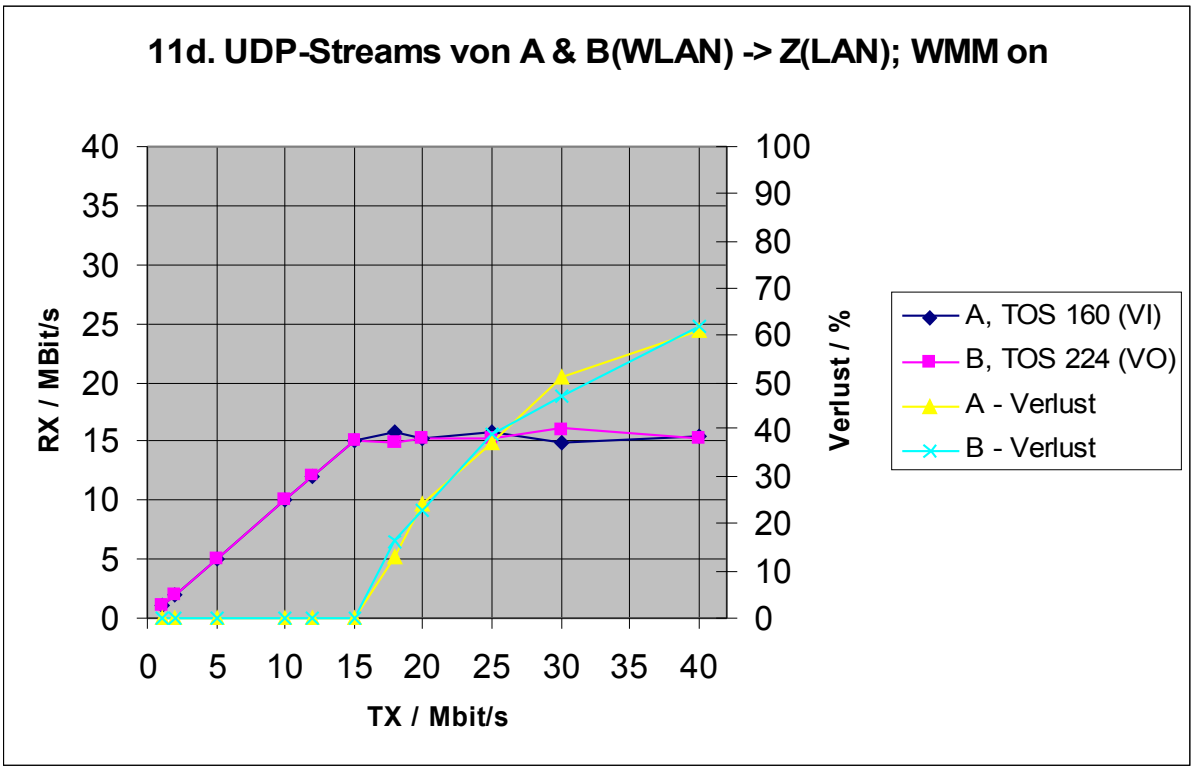


Abbildung 16: Diagramm 11d

Zum Abschluss dieses Unterkapitels noch eine Anmerkung zu gleich priorisierten Streams: Wie nicht anders zu erwarten, sind die Messergebnisse bei zwei Streams der gleichen WMM-Kategorie jeweils bis auf minimalste Abweichungen identisch. In Anhang C ist dies in den Diagrammen 11a und 11e zu sehen. Unterschiede gibt es lediglich bei der genutzten Gesamtbandbreite. Zwei BE-Streams nutzen insgesamt knapp 28 MBit/s, zwei VO-Streams um die 30 MBit/s.

In diesem nun abgeschlossenen Unterkapitel wurde die Priorisierung zweier Datenübertragungen von unterschiedlichen WLAN-Stationen betrachtet. Im folgenden Kapitel wird die interne Priorisierung innerhalb einer Station untersucht.

5.2.4 Client-interne Priorisierung

In den bereits abgehandelten Messungen musste die Treibersoftware der WLAN-Karte nur Daten aus einer einzigen Warteschlange versenden (siehe Abbildung 5, S.26). Eine interne Kollisionsverhinderung zwischen Daten unterschiedlicher Warteschlangen war nicht nötig.

Bei korrekter Implementierung der EDCAF (siehe Kapitel 2.3.2, S.22ff), die sozusagen jede Warteschlange zu einer separaten WLAN-Station macht, sollte die interne Priorisierung ebenso gut funktionieren wie bei den bisherigen Messungen. Einzig die interne Kollisionsverhinderung würde hier einen Unterschied machen.

Um das genaue Verhalten herauszufinden, wurde zunächst die Funktionalität des WRT54G getestet und gemessen. In einem späteren Versuchsaufbau wurden die WLAN-Clients untersucht.

Für die Messung wurden zwei Streams mit unterschiedlicher TOS-Markierung benötigt, die vom AP, also dem WRT54G, an zwei WLAN-Clients versendet werden. Da mit der originalen Linksys-Firmware kein Zugriff auf das Linux-System des Gerätes besteht, wurden diese Streams aus dem LAN zum Router verschickt, der diese dann ins WLAN bridged. Da Iperf nicht die Möglichkeit bietet, zwei Streams von einem Computer aus mit unterschiedlichen TOS-Werten zu versenden, waren zwei Stationen im LAN nötig, die als Iperf-Clients die Streams versandten. Zum Einsatz kamen hierfür die Computer Y und Z. Y versandte Daten an den WLAN-Client A und Z an den WLAN-Client B.

Interne Priorisierung im Linksys WRT54G – Ohne WMM

Zunächst wurden wieder zwei Messreihen ohne WMM aufgenommen. Einmal wurden die Streams mit den TOS-Werten 0 und 0 markiert, einmal mit 0 und 224. Die Messungen zeigten keinen Unterschied. In beiden Fällen wurde die gleiche Datenmenge jedes Streams an die WLAN-Clients übertragen. Der Gesamtdurchsatz beider Streams betrug in jeder Messung etwa 29 MBit/s. Dies ist die gleiche Menge, die auch schon ohne Konkurrenzsituation bei einem einzelnen Stream vom LAN ins WLAN an Maximaldurchsatz gemessen wurde (siehe Kapitel 5.2.2, S.49ff). Ohne WMM fand also keine Priorisierung statt. (siehe Anhang C, Diagramme 20a und 20b)

Interne Priorisierung im Linksys WRT54G – Mit WMM

Mit aktiviertem WMM wurden drei verschiedene Stream-Kombinationen mit den TOS-Werten 0/224, 0/32 und 160/224 gemessen. In Abbildung 17 sind die Messreihen mit TOS 0 und 224 zu sehen.

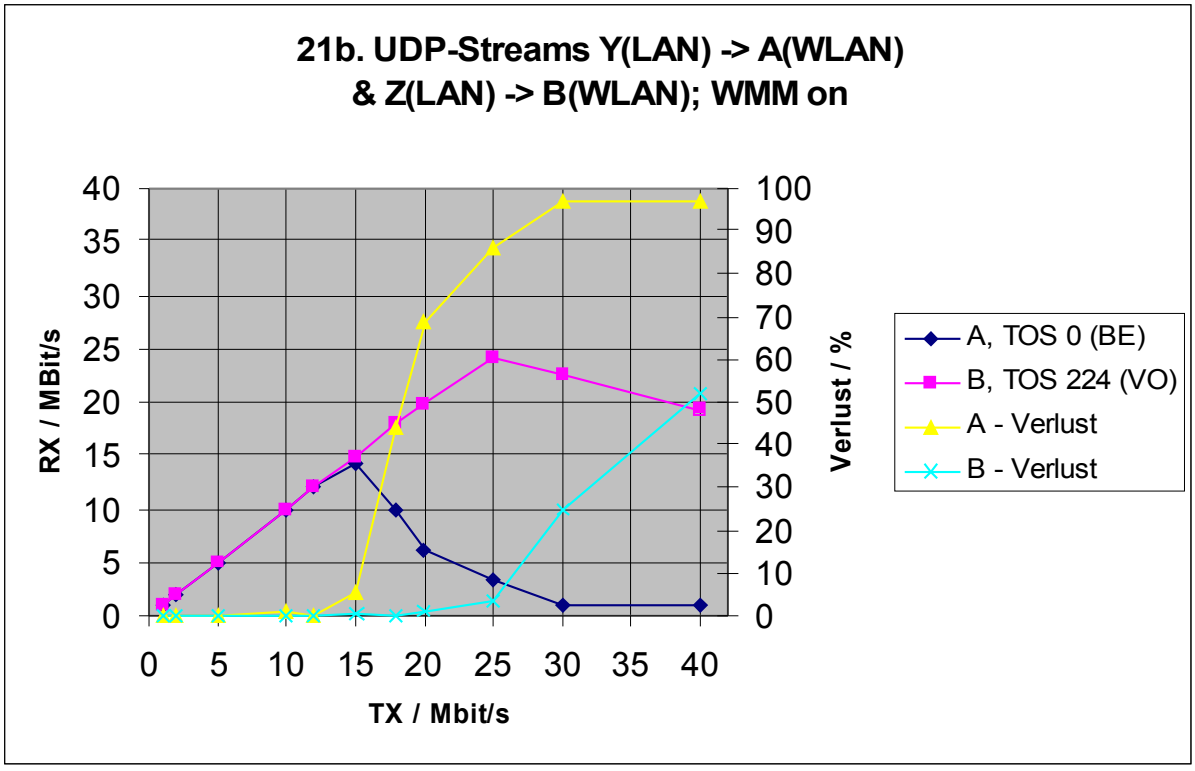


Abbildung 17: Diagramm 21b

Bis zu den Messpunkten bei 25 MBit/s TX sind die Ergebnisse erwartungsgemäß und bestätigen die bisherigen Erkenntnisse zur Funktion von WMM. Bei den weiteren Messpunkten, wo insgesamt 60 MBit/s und mehr aus dem LAN beim Router zum Versand ins WLAN eintreffen, fällt der Messeinbruch des VO-Streams auf. Schon bei der Aufnahme der Messdaten waren Unregelmäßigkeiten aufgefallen. In das obige Diagramm flossen wie bei allen Diagrammen nur die Mittelwerte über die jeweils 30-sekündigen Messungen ein. Iperf wurde jedoch immer so ausgeführt, dass auch die Daten der drei 10-Sekunden-Abschnitte ausgegeben wurden. Während die Messwerte der 10-Sekunden-Abschnitte bei den bisherigen Messungen immer sehr wenig um den Durchschnittswert der Gesamtmessung schwankten, zeigte sich bei der Messung mit 2*30 MBit/s TX Folgendes:

An Client B gab Iperf dies aus:

```
[  4] local 192.168.1.21 port 5001 connected with 192.168.1.30
port 5001
[  4]  0.0-10.0 sec  25.0 MBytes  21.0 Mbits/sec  0.071 ms
7494/25352 (30%)
[  4] 10.0-20.0 sec  28.4 MBytes  23.8 Mbits/sec  0.272 ms
5260/25493 (21%)
```

```
[  4] 20.0-30.0 sec  27.3 MBytes  22.9 Mbits/sec  0.309 ms
6010/25495 (24%)
[  4]  0.0-30.0 sec  80.9 MBytes  22.6 Mbits/sec  0.181 ms
18794/76480 (25%)
```

Die Bandbreite schwankt in den 10-Sekunden-Zyklen zwischen 21 und 23,8 MBit/s. Im Vergleich zu anderen Messungen ist dies eine sehr große Abweichung vom Mittelwert.

An Client A sah die Ausgabe des dortigen Iperf-Servers so aus:

```
[  4] local 192.168.1.11 port 5001 connected with 192.168.1.40
port 5001
[  4]  0.0-10.0 sec  3.55 MBytes  2.98 Mbits/sec  2.389 ms
22822/25356 (90%)
[  4] 10.0-20.0 sec  0.01 MBytes  0.01 Mbits/sec  2.348 ms
25501/25509 (1e+02%)
[  4] 20.0-30.0 sec  0.04 MBytes  0.03 Mbits/sec  0.571 ms
25481/25508 (1e+02%)
[  4]  0.0-30.0 sec  3.74 MBytes  1.04 Mbits/sec  1.106 ms
73864/76532 (97%)
```

In den ersten 10 Sekunden kommen am Iperf-Server noch knapp 3 MBit/s des BE-Streams an, danach so gut wie gar nichts mehr. Der Datenverlust steigt auf über 99,9%.

Die 2*40 MBit/s-Messung sieht an Client A sehr ähnlich aus, an Client B geht der Empfang des VO-Streams auf unter 20 MBit/s zurück.

Die beiden weiteren durchgeführten Messreihen zeigten bei über 2*25 MBit/s TX die gleichen Phänomene (siehe Anhang C, Diagramme 21c und 21d).

Der Grund für diese Problematik ist im WRT54G zu suchen. Alle Clients hatten bei vorherigen Messungen bewiesen, dass sie mit den hohen Datenraten zurecht kommen. Nur der WRT54G hatte bereits erste Anzeichen eines Überlastungsproblems bei bestimmten vom LAN ins WLAN zu sendenden Bandbreiten gezeigt. Wie in Kapitel 5.2.2 (→ Mit WMM) schon erwähnt, brach der Maximaldurchsatz bei einzelnen VO- und VI-Streams vom LAN ins WLAN bei 50 MBit/s ein. Nach verschiedenen wenigen Messungen ohne WMM, aber mit hohen Datenraten, waren auch schon Neustarts (durch Unterbrechung der Stromversorgung) des WRT54G nötig gewesen. In diesen Fällen waren vor dem

Neustart nur noch Übertragungen mit maximal 24 MBit/s vom AP ins WLAN möglich gewesen. Da bereits 29 MBit/s als Maximalwert gemessen worden waren, fiel dies auf. Nach dem Neustart war diese Datenrate auch wieder möglich.

Unterhalb der problematischen Messwerte verhielt sich das Linksys-Gerät in Bezug auf WMM bei interner Priorisierungsnotwendigkeit zweier Streams ebenfalls nicht ganz erwartungsgemäß. Bei den Stream-Kombinationen mit TOS 0/32 und 160/224 wurde jeweils der niedriger priorisierte zu stark zurückgedrängt.

Wahrscheinlich hat der WRT54G ein Problem, große Datenmengen zu verarbeiten und zu routen. Bei 2*30 MBit/s muss er immerhin über 7 MiByte/s und bei 2*40 MBit/s über 9,5 MiByte/s an Daten verarbeiten.

Interne Priorisierung im MadWifi-Client – Mit WMM

Da wie bereits erläutert mit Iperf kein Versand unterschiedlich priorisierter Streams möglich ist und kein anderes adäquates Tool zur Verfügung stand, wurde die interne Priorisierung in den MadWifi-Clients auf andere Weise getestet.

Computer B wurde als WLAN-Client eingesetzt, von dem sowohl ein Iperf-Client einen hoch priorisierten Stream an den LAN-Client Y versandte als auch eine Testdatei per SCP[38] zum LAN-Client A übertragen wurde. Die Testdatei war zuvor mit dem Befehl `dd if=/dev/urandom of=testfile bs=1024 count=153600` erstellt worden und enthielt somit 150 MiB an Zufallsdaten.

38 SCP – Secure Copy; Protokoll und Programm zur verschlüsselten Übertragung von Daten in Netzwerken; weitere Informationen u.a. bei http://www.uni-koeln.de/rrzk/netze/ssh/sshscp.html

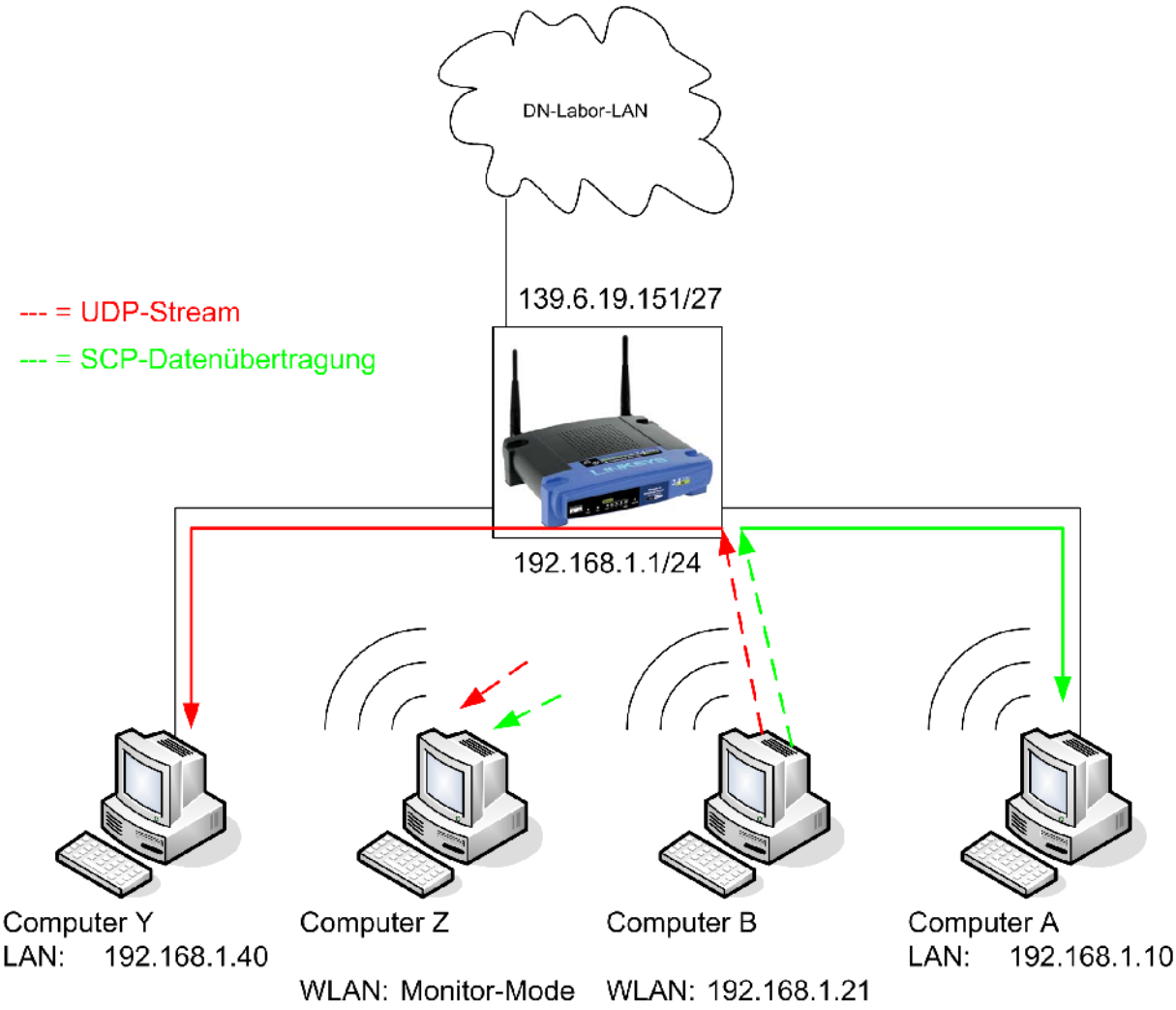


Abbildung 18: Aufbau Client-interne Priorisierung in MadWifi

Um die beiden Streams auf der Luftschnittstelle beobachten zu können, wurde auf Computer Z der Monitor-Mode der WLAN-Karte aktiviert und die Daten mit Ethereal mitgeschnitten. Der gesamte Aufbau ist in obiger Abbildung 18 zu sehen.

Standardmäßig wurde die SCP-Datenübertragung von Computer B in die WMM-Kategorie BK eingeordnet. Der Iperf-Stream wurde mit TOS-Wert 224 als VO-Stream versendet.

Zuerst gestartet wurde die SCP-Datenübertragung. Nach etwa 21 Sekunden wurde der 20-sekündige UDP-Stream des Iperf-Client begonnen.

In Abbildung 19 ist der Ethereal-Mitschnitt des WLAN-Monitor-Computers grafisch aufbereitet dargestellt. Auf der X-Achse ist die Zeit aufgetragen, auf der Y-Achse die am WLAN-Monitor empfangene Datenmenge in Byte. Eingestellt wurde eine

Auflösung in X-Richtung von 2 Pixeln pro 0,1 Sekunden für die Darstellung auf dem Bildschirm.

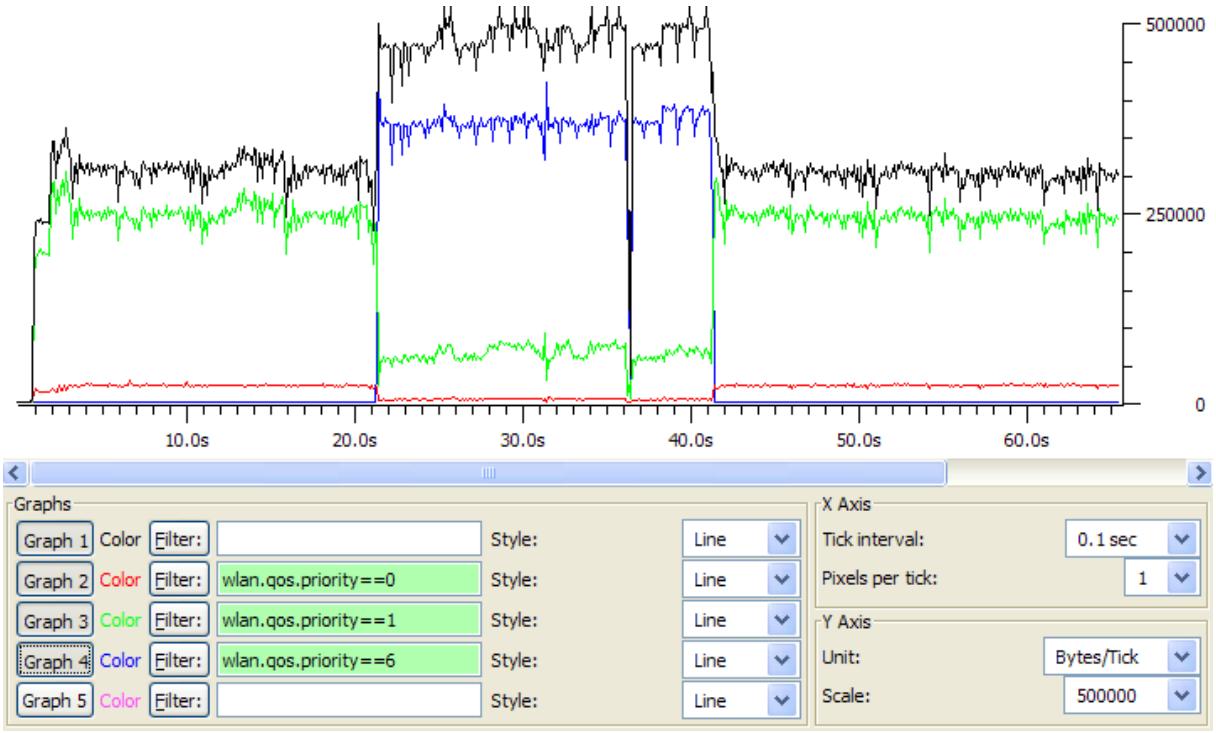


Abbildung 19: Ethereal-Mitschnitt Client-interne Priorisierung in MadWifi

Der schwarze Graph stellt die gesamte empfangene Datenmenge pro Zeiteinheit dar. Darin inbegriffen sind beispielsweise auch die Beacon-Frames, ACKs und Daten anderer WLANs und Geräte, die die gleiche oder überlappende Frequenzen wie die am WLAN-Monitor eingestellte nutzen.

Der rote Graph zeigt die Datenmenge in WMM-Kategorie BE, der grüne in Kategorie BK (SCP-Datenübertragung) und der blaue Graph Daten in Kategorie VO (UDP-Stream).

Zu Beginn hat die SCP-Datenübertragung eine Bandbreite von etwa 20 MBit/s (grün). Die TCP-Acknowledgements dazu werden in Kategorie BE übertragen (rot). Ab dem Beginn des 25 MBit/s-VO-Streams geht die Datenrate der SCP-Verbindung auf etwa 5,5 MBit/s zurück. Der VO-Stream kann dadurch mit Verlusten von gerade mal 0,056% am Iperf-Server empfangen werden.

Unmittelbar nach der Fertigstellung der VO-Übertragung steigt die genutzte Bandbreite der SCP-Datenübertragung wieder auf den alten Wert an.

Der Messeinbruch bei Sekunde 36 ist höchstwahrscheinlich auf ein Performance-Problem des Computers zurückzuführen, mit dem die Aufzeichnung durchgeführt wurde. Immer wieder kam es bei Ethereal-Mitschnitten zu diesem Phänomen. Diverse Kontrollmessungen am wirklichen Empfänger der Daten ergaben immer einen durchgehenden Datenstrom auch wenn Ethereal am WLAN-Monitor Einbrüche zeigte.

Die durchgeführte Messung zeigt eine perfekt funktionierende interne Priorisierung unterschiedlicher Datenströme. Die Messergebnisse decken sich sogar genau mit der in Diagramm 11f (Anhang C) zu sehenden Messung bei 2*25 MBit/s TX. Dort wird der VO-Stream ebenfalls ohne Verluste übertragen. Dem BK-Stream steht dort wie hier die restliche Bandbreite von etwas mehr als 5 MBit/s zur Verfügung.

5.2.5 Weitere Beobachtungen und aufgetretene Probleme

Ethereal

Wie im letzten Unterkapitel schon angedeutet, war das Mitschneiden von Datenübertragungen mit Ethereal im Monitor-Mode mit vielen Komplikationen verbunden. Oft waren Messeinbrüche zu sehen, die in Wirklichkeit nicht existierten. Widerlegt werden konnten die Messeinbrüche beispielsweise durch eine 30-sekündige mitgeschnittene Iperf-Übertragung, die mit Verlusten von weniger als 1 Promille am Iperf-Server ankam, jedoch in Ethereal über insgesamt 10 Sekunden als nicht vorhanden (0,0 MBit/s) angezeigt wurde. Eine andere angewandte Testmöglichkeit war der Vergleich des Ethereal-Mitschnitts mit dem Verlauf der in dem Tool nload[39] am Iperf-Server angezeigt wurde. Wiederum zeigte Ethereal Messeinbrüche wo am Empfänger laut nload keine existierten.

Die wenigsten Messeinbrüche konnten in Ethereal erreicht werden, indem die Datenpakete beim Mitschnitt auf die ersten 300 Byte gekürzt wurden. Dies ist in den Capture-Optionen einstellbar.

Trotzdem bestand auch weiterhin das Problem, dass ein Computer mit 700MHz-CPU und 256MiB RAM nach etwa 5 Messungen und Mitschnitt von jeweils 50 bis 100 MiB Daten den Dienst komplett verweigerte.

39 nload - http://www.roland-riegel.de/nload/

Unter anderem aus diesen Gründen wurden nur wenige Messungen mit Ethereal durchgeführt. Das Tool wurde eher zur Überprüfung der Aktivierung und Deaktivierung von WMM, der Einstrahlung fremder WLANs und zur Protokollanalyse verwendet.

WLAN-Karten mit MadWifi-Treiber

Um die eingesetzten WLAN-Karten in den funktionsfähigen Monitor-Mode zu versetzen, war ein kleiner Trick nötig. Aktiviert man direkt nach dem Start des Computers den Monitor-Mode ohne die WLAN-Karte wenigstens kurz im Client-Mode[40] betrieben zu haben, so scheint es auf den ersten Blick in Ethereal-Aufzeichnungen zu funktionieren. Jedoch werden dann anscheinend nur Kontroll- und Management-Frames mitgeschnitten. Datenframes werden nicht aufgenommen. Vor der Aktivierung des Monitor-Modes ist also zwingend einige Sekunden Client-Mode nötig.

Weitere Probleme traten beim Betrieb der WLAN-Karten auf, wenn einige Male zwischen Monitor-Mode und Client-Mode hin und her geschaltet worden war. Zum Beispiel traten dann Verluste bei Datenübertragungen vom WLAN-Client auf oder die Round-Trip-Time zum AP wuchs bei Pings auf viel zu hohe Werte von bis zu ¼ Sekunde. In diesem Fall half dann nur, den Computer ganz auszuschalten, ihn eine Weile vom Strom zu trennen und dann neu zu starten.

Linksys WRT54G

Wie in Kapitel 5.2.4 (→ Interne Priorisierung im WRT54G) schon ausführlich beschrieben, hat der Router Probleme bei der Übertragung großer Datenmengen vom LAN ins WLAN, teilweise auch in umgekehrter Richtung. Nach versuchten Übertragungen von 2*40 MBit/s aus dem LAN ins WLAN musste das Linksys-Gerät entweder durch Unterbrechung der Stromversorgung neu gestartet werden oder es half manchmal auch einige Minuten zu warten.

40 Client-Mode ist hier gleichzusetzen mit dem Infrastructure-Mode, der Ad-Hoc-Mode wurde nicht betrachtet oder getestet

6 Schlussbetrachtung

Trotz der Probleme, die in den letzten Unterkapiteln angesprochen wurden, hinterlässt WMM insgesamt einen guten Eindruck. Dass die Priorisierung nicht nur theoretisch sondern auch praktisch funktioniert, konnte in vielen Messungen gezeigt werden. Wenn Schwierigkeiten auftauchten, waren diese am ehesten auf Fehler in Hard- oder Software zurückzuführen. Eine Verbesserung des MadWifi-Treibers ist auf jeden Fall möglich, zumindest in Bezug auf das unvollständige und fehlerhafte TOS-Mapping. Das Problem des Linksys-Routers müsste genauer untersucht werden.

Vor allem in kleinen WLAN-Netzwerken mit nicht all zu vielen Clients bietet WMM eine gute Möglichkeit, Multimedia-Streams so zu priorisieren, dass Datenverluste klein bleiben und die Qualität hoch. Bei sehr vielen Streams der gleichen Priorität hingegen ist ein Problem vorprogrammiert. Die begrenzte Bandbreite des WLANs wird auch durch WMM nicht größer. Bei vielen Nutzern steht jedem weniger Bandbreite zur Verfügung und WMM kann nur die wichtigen Daten vor unwichtigeren priorisieren. Übersteigen die zu übertragenden hoch priorisierten Datenmengen die maximale Bandbreite, muss auch WMM kapitulieren.

Bei der zunehmenden Verbreitung von kabellosen Audio- und Videostreaming-Clients und VoIP-WLAN-Telefonen wird ein Verfahren zur Qualitätssicherung im WLAN dringend benötigt. WMM ist der richtige Ansatz dafür.

Das vermutlich viel größere Problem dürfte dagegen sein, dass die wenigsten Datenübertragungen heute mit Prioritäten, zum Beispiel über den TOS-Wert, gekennzeichnet sind. Video-Streams, die ohne TOS-Markierung aus dem Internet an einem AP ankommen und zu einem WLAN-Client übertragen werden sollen, werden keinerlei Priorisierung erhalten. Auch gibt es derzeit kaum Soft-Phones für VoIP, die den Audio-Stream TOS-markieren können. Dass unter Linux für TOS-Werte größer 159 Root-Rechte nötig sind, macht den Versand von Daten in der WMM-Kategorie VO für normale Nutzer unmöglich.

WMM existiert und funktioniert recht gut, seine Leistungsfähigkeit als QoS-Mechanismus kann es aber erst dann zeigen, wenn die verschiedenen Anwendungen es auch nutzen.

Anhang A: Shell-Skripte

Beispielhaft hier nur die Skripte für Computer B.

lan.sh:

```
#!/bin/bash

while [ "$antwort" != "q" ]; do
     echo " Menue:"
     echo " 1 - LAN private"
     echo " 2 - LAN public"
     echo " 3 - WLAN client"
     echo " 4 - WLAN monitor"
     echo " 5 - LAN private + WLAN monitor"
     echo " 6 - LAN public + WLAN monitor"
     echo " q - quit"
     read answer

     case $answer in
     1)   #LAN private
          ifconfig eth0 down
          ifconfig ath0 down
          ifconfig eth0 192.168.1.20 netmask
255.255.255.0
          route add default gw 192.168.1.1
          ifconfig eth0 up
          echo done
     ;;

     2)   #LAN public
          ifconfig eth0 down
          ifconfig ath0 down
          ifconfig eth0 139.6.19.151 netmask
255.255.255.224
          route add default gw 139.6.19.129
          ifconfig eth0 up
          echo done
     ;;

     3)   #WLAN client
```

```
            ifconfig eth0 down
            ifconfig ath0 down

            wlanconfig ath0 destroy

            modprobe ath_pci
            wlanconfig ath0 create wlandev wifi0 wlanmode
sta
            iwpriv ath0 mode 3

            #iwconfig ath0 essid fh-work
            #iwconfig ath0 mode Managed
            #iwconfig ath0 key 22222777779999944444000001

            ifconfig ath0 192.168.1.21 netmask
255.255.255.0
            route add default gw 192.168.1.1

            ifconfig ath0 up

            wpa_supplicant -Bw -Dmadwifi -iath0 -c
/etc/wpa_supplicant.conf

            iwpriv ath0 wmm 1
            echo done
      ;;

      4)    #WLAN monitor
            ifconfig ath0 down
            pkill wpa_supplicant
            wlanconfig ath0 destroy
            modprobe ath_pci
            wlanconfig ath0 create wlandev wifi0 wlanmode
monitor
            ifconfig ath0 up 192.168.150.150
            echo done
      ;;

      5)    #LAN private + WLAN monitor
            ifconfig eth0 down

            ifconfig ath0 down
```

```
          pkill wpa_supplicant
          wlanconfig ath0 destroy
          modprobe ath_pci
          wlanconfig ath0 create wlandev wifi0 wlanmode
monitor
          ifconfig ath0 up 192.168.150.150

          ifconfig eth0 192.168.1.20 netmask
255.255.255.0
          route add default gw 192.168.1.1
          ifconfig eth0 up
          echo done
     ;;

     6)   #LAN public + WLAN monitor
          ifconfig eth0 down
          ifconfig ath0 down
          ifconfig eth0 139.6.19.151 netmask
255.255.255.224
          route add default gw 139.6.19.129
          ifconfig eth0 up

          pkill wpa_supplicant
          wlanconfig ath0 destroy
          modprobe ath_pci
          wlanconfig ath0 create wlandev wifi0 wlanmode
monitor
          ifconfig ath0 up 192.168.150.150
          echo done
     ;;

     q)   exit
     ;;

     *)   echo wrong selection!
     ;;

     esac
done
```

sync.sh:

```
#!/bin/bash

ntpdate time.fh-koeln.de
seconds=`date +%S`
if [ "$seconds" -gt 50 ]
then
echo Ã¼ber 50
sleep 15
seconds=`date +%S`
fi
while [ "$seconds" != 00 ]
do
seconds=`date +%S`
#echo $seconds
done
sleep .017
date +%N

echo iperf -f m -i 10 -u -B 192.168.1.21 -t 30 -c
192.168.1.30 -b $1000000 -S $2
iperf -f m -i 10 -u -B 192.168.1.21 -t 30 -c
192.168.1.30 -b $1000000 -S $2
```

mess.sh:

```
#!/bin/bash

for x in 1 2 5 10 12 15 18 20 25 30 40;
do

ntpdate time.fh-koeln.de
seconds=`date +%S`

if [ "$seconds" -gt 50 ]
  then
  echo Ã¼ber 50
  sleep 15
  seconds=`date +%S`
fi
```

```
while [ "$seconds" != 00 ]
  do
  seconds=`date +%S`
  #echo $seconds
done

sleep .017
date +%N

echo iperf -f m -i 10 -u -B 192.168.1.21 -t 30 -c
192.168.1.30 -b "$x"000000 -S $1
iperf -f m -i 10 -u -B 192.168.1.21 -t 30 -c
192.168.1.30 -b "$x"000000 -S $1

done
```

tosping.sh:

(Skript von Computer Z)

```
#!/bin/bash

i=0
while [ "$i" -le 255 ]
do
     echo
     echo '======'
     echo '> '$i':'
     echo '======'
     ping -c 1 192.168.1.20 -Q $i
     i=$[$i +1]
     read answer
     if [ "$answer" = "w" ]
     then i=$[$i -1]
     fi
     if [ "$answer" = "n" ]
     then
          echo anderer tos-wert:
          read answer
          echo '======'
          echo '> '$answer':'
          echo '======'
          ping -c 1 192.168.1.20 -Q $answer
```

```
            read answer
        fi
    done
    echo
    echo ende
```

Anhang B: MadWifi-Codeausschnitt

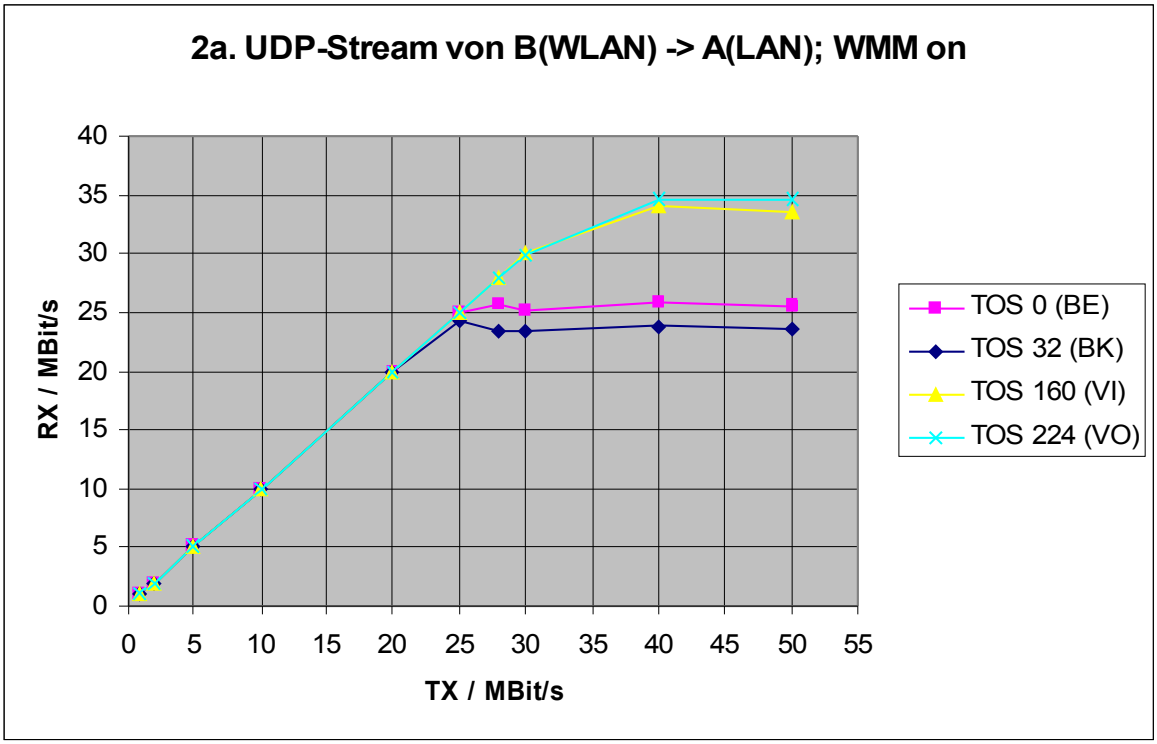
Ausschnitt aus Datei ieee80211_output.c:

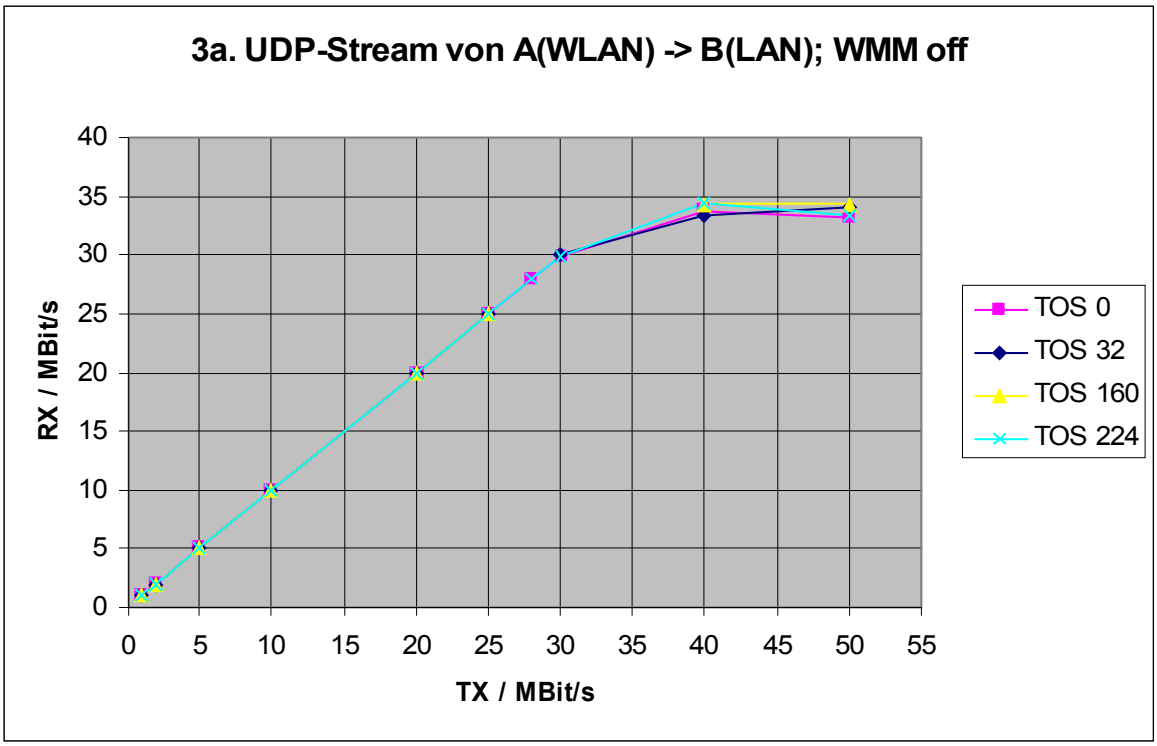
```
75)  /*
76)   * Determine the priority based on VLAN and/or IP TOS.
     Priority is
77)   * written into the skb->priority field. On success,
     returns 0. Failure
78)   * due to bad or mis-matched vlan tag is indicated by
     non-zero return.
79)   */
80)  static int
81)  ieee80211_classify(struct ieee80211_node *ni, struct
     sk_buff *skb)
82)  {
83)  struct ieee80211vap *vap = ni->ni_vap;
84)  struct ether_header *eh = (struct ether_header *) skb-
     >data;
85)  int v_wme_ac = 0, d_wme_ac = 0;
86)
87)  /* default priority */
88)  skb->priority = WME_AC_BE;
89)
90)  if (!(ni->ni_flags & IEEE80211_NODE_QOS))
91)       return 0;
92)
93)  /*
94)   * If node has a vlan tag then all traffic
95)   * to it must have a matching vlan id.
96)   */
97)  if (ni->ni_vlan != 0 && vlan_tx_tag_present(skb)) {
98)       u_int32_t tag=0;
99)       int v_pri;
100)
101)      if (vap->iv_vlgrp == NULL) {
102)           IEEE80211_NODE_STAT(ni, tx_novlantag);
103)           ni->ni_stats.ns_tx_novlantag++;
104)           return 1;
105)      }
106)      if (((tag = vlan_tx_tag_get(skb)) &
     VLAN_VID_MASK) !=
```

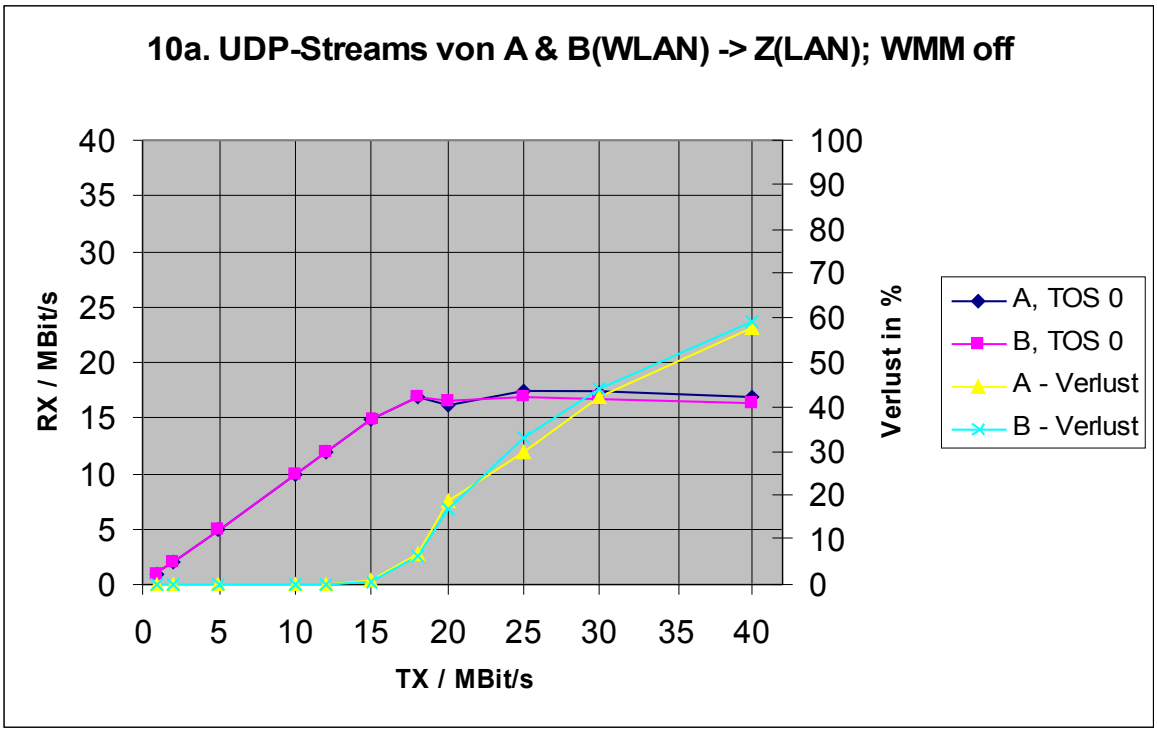
```
107)          (ni->ni_vlan & VLAN_VID_MASK)) {
108)           IEEE80211_NODE_STAT(ni, tx_vlanmismatch);
109)           ni->ni_stats.ns_tx_vlanmismatch++;
110)           return 1;
111)      }
112)      if (ni->ni_flags & IEEE80211_NODE_QOS) {
113)           v_pri = (tag >> VLAN_PRI_SHIFT) &
     VLAN_PRI_MASK;
114)           switch (v_pri) {
115)           case 1:
116)           case 2:        /* Background (BK) */
117)                v_wme_ac = WME_AC_BK;
118)                break;
119)           case 0:
120)           case 3:        /* Best Effort (BE) */
121)                v_wme_ac = WME_AC_BE;
122)                break;
123)           case 4:
124)           case 5:        /* Video (VI) */
125)                v_wme_ac = WME_AC_VI;
126)                break;
127)           case 6:
128)           case 7:        /* Voice (VO) */
129)                v_wme_ac = WME_AC_VO;
130)                break;
131)           }
132)      }
133) }
134)
135) if (eh->ether_type == __constant_htons(ETHERTYPE_IP)) {
136)      const struct iphdr *ip = (struct iphdr *)
137)           (skb->data + sizeof (struct ether_header));
138)      /*
139)       * IP frame, map the TOS field.
140)       *
141)       * XXX: fill out these mappings???
142)       */
143)      switch(ip->tos) {
144)      case 0x08:                    /* Background */
145)      case 0x20:
146)           d_wme_ac = WME_AC_BK;
147)           break;
```

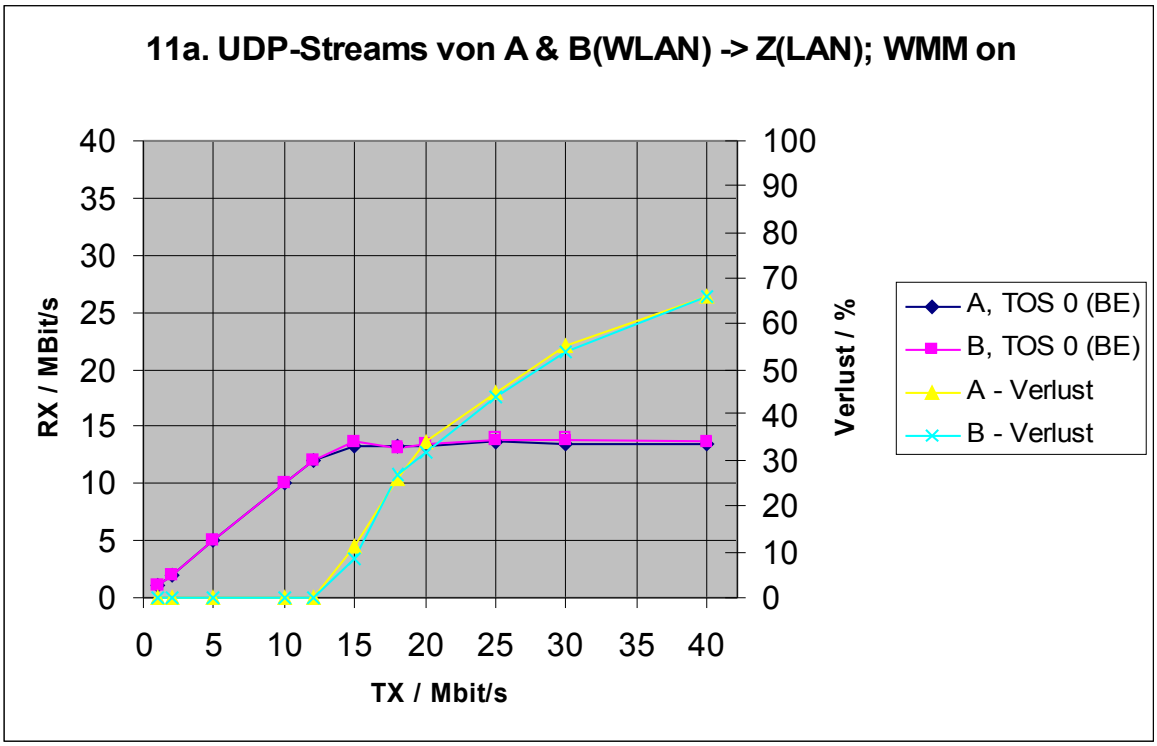
```
148)      case 0x28:                    /* Video */
149)      case 0xa0:
150)           d_wme_ac = WME_AC_VI;
151)           break;
152)      case 0x30:                    /* Voice */
153)      case 0xe0:
154)      case 0x88:                    /* XXX UPSD */
155)      case 0xb8:
156)           d_wme_ac = WME_AC_VO;
157)           break;
158)      default:                 /* All others */
159)           d_wme_ac = WME_AC_BE;
160)           break;
161)      }
162) } else {
163)      d_wme_ac = WME_AC_BE;
164) }
165) skb->priority = d_wme_ac;
166) if (v_wme_ac > d_wme_ac)
167)      skb->priority = v_wme_ac;
```

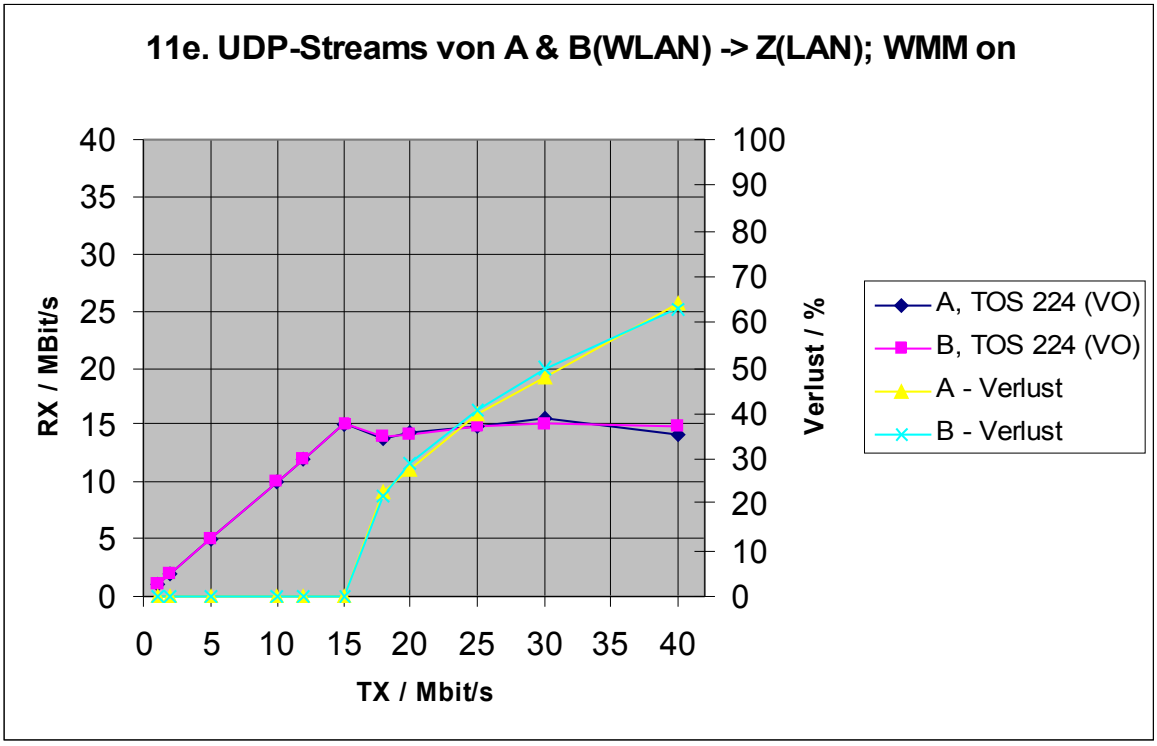
Anhang C: Messdiagramme

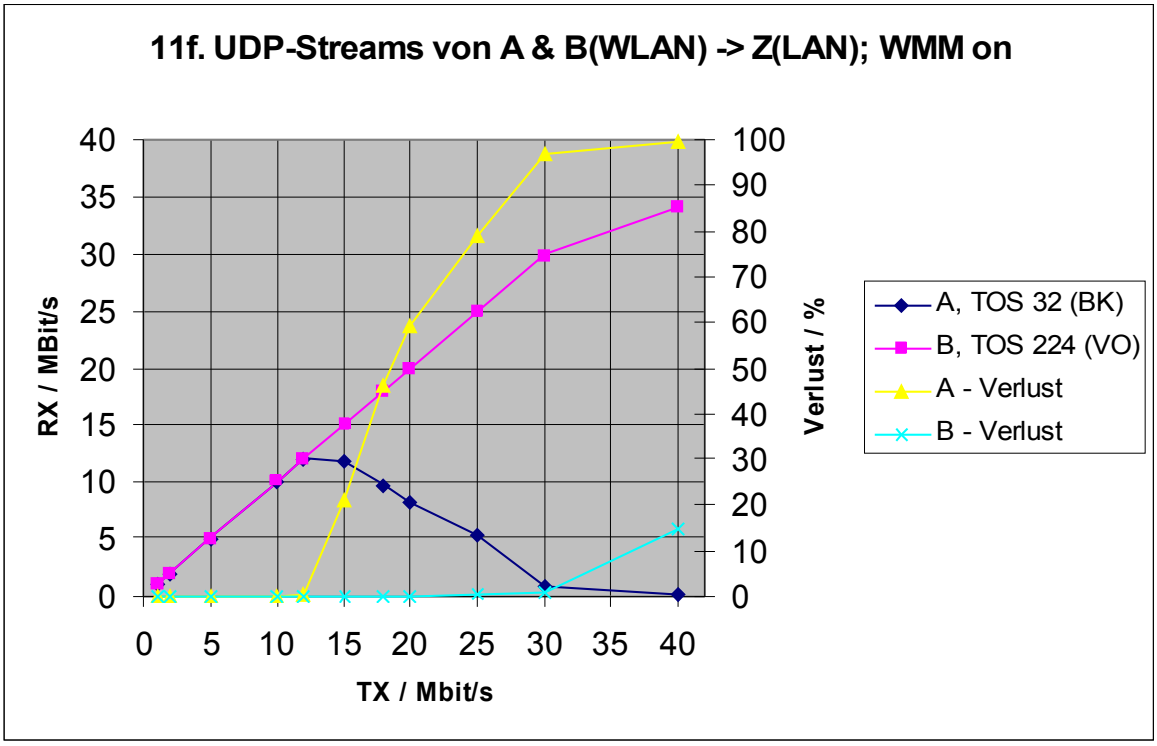


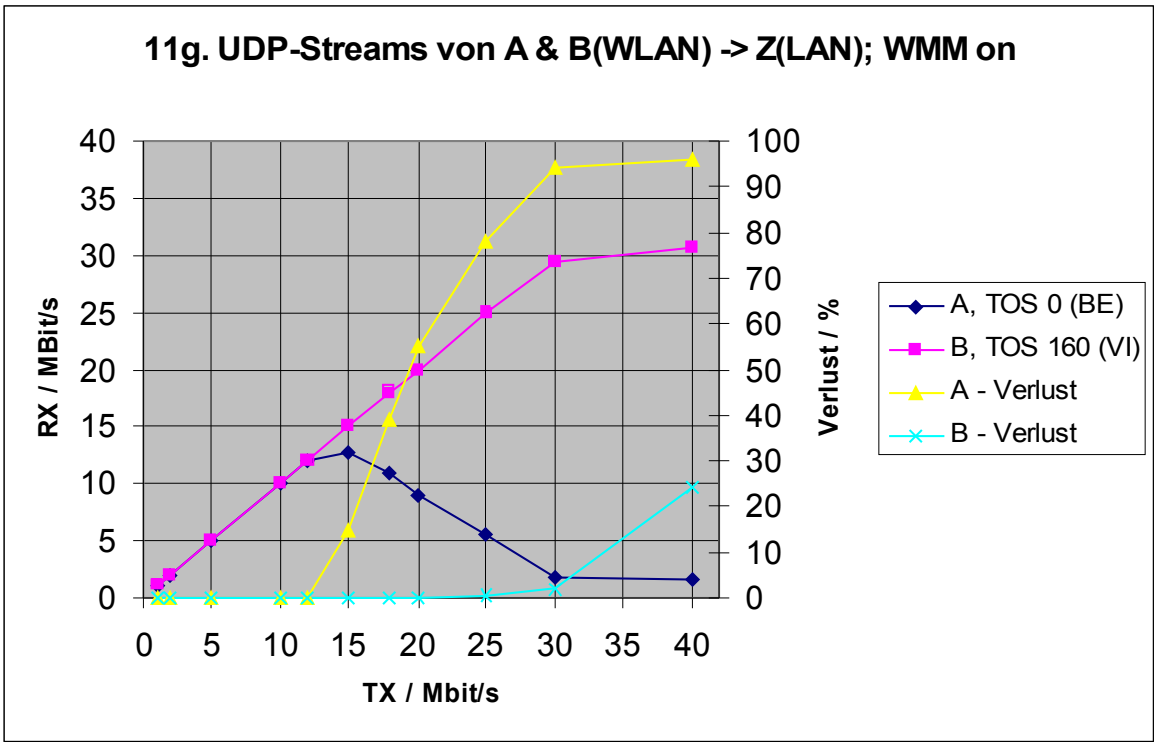


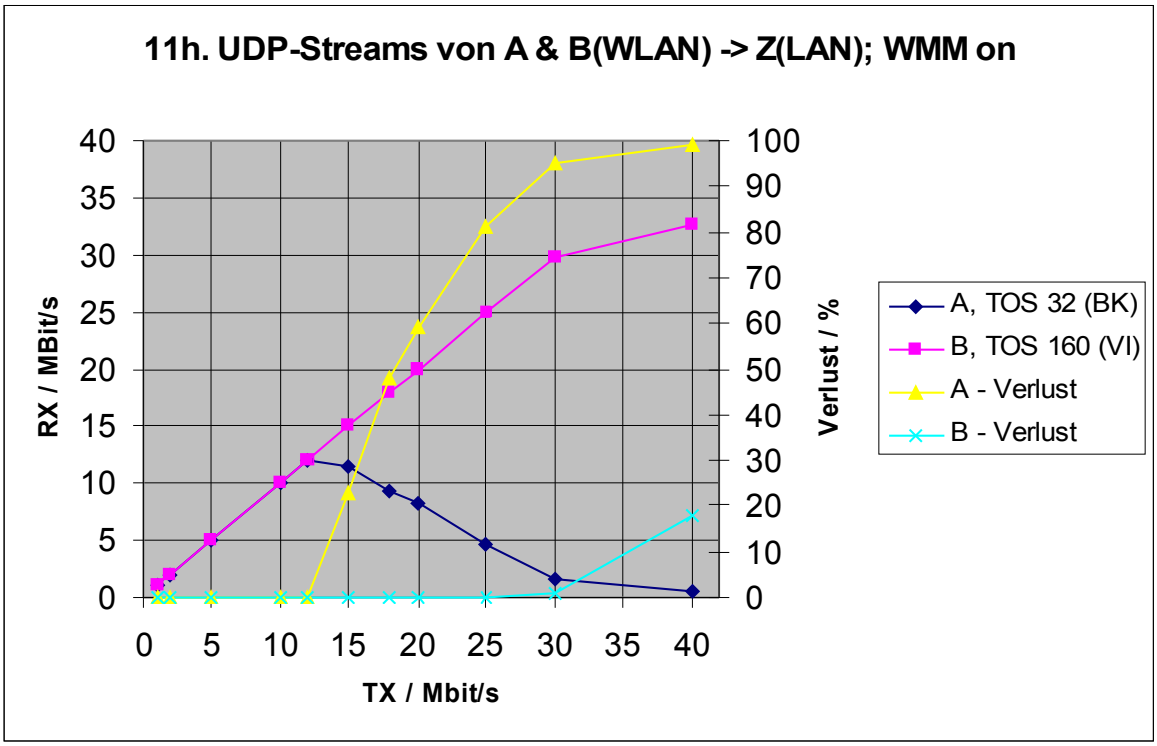

10a. UDP-Streams von A & B(WLAN) -> Z(LAN); WMM off
RX / MBit/s
Verlust in %
TX / MBit/s
A, TOS 0
B, TOS 0
A - Verlust
B - Verlust

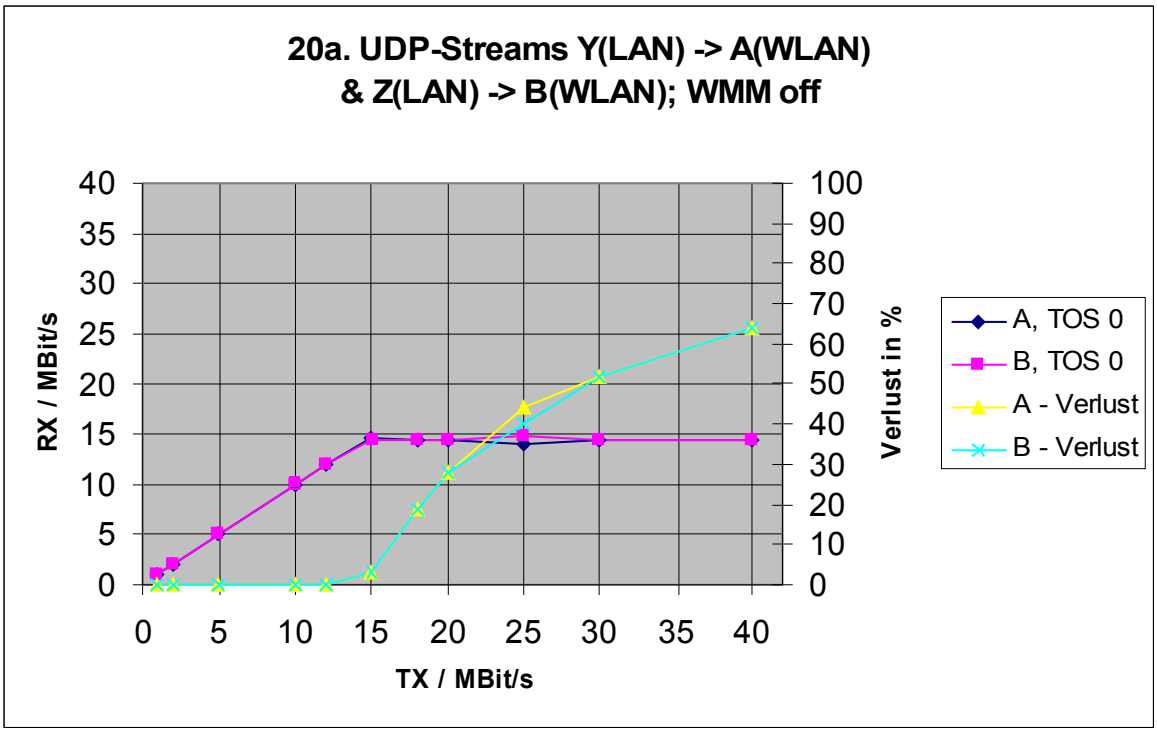

11a. UDP-Streams von A & B(WLAN) -> Z(LAN); WMM on
RX / MBit/s
Verlust / %
TX / Mbit/s
A, TOS 0 (BE)
B, TOS 0 (BE)
A - Verlust
B - Verlust

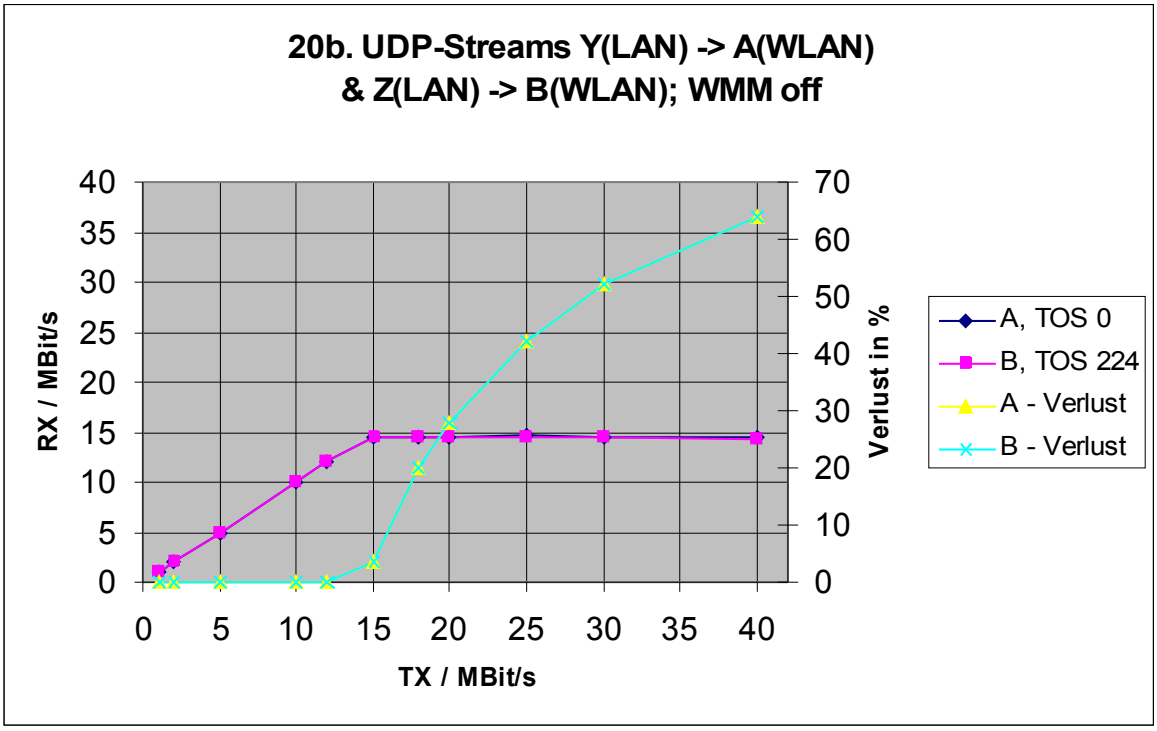

11e. UDP-Streams von A & B(WLAN) -> Z(LAN); WMM on
RX / MBit/s
Verlust / %
TX / Mbit/s
A, TOS 224 (VO)
B, TOS 224 (VO)
A - Verlust
B - Verlust

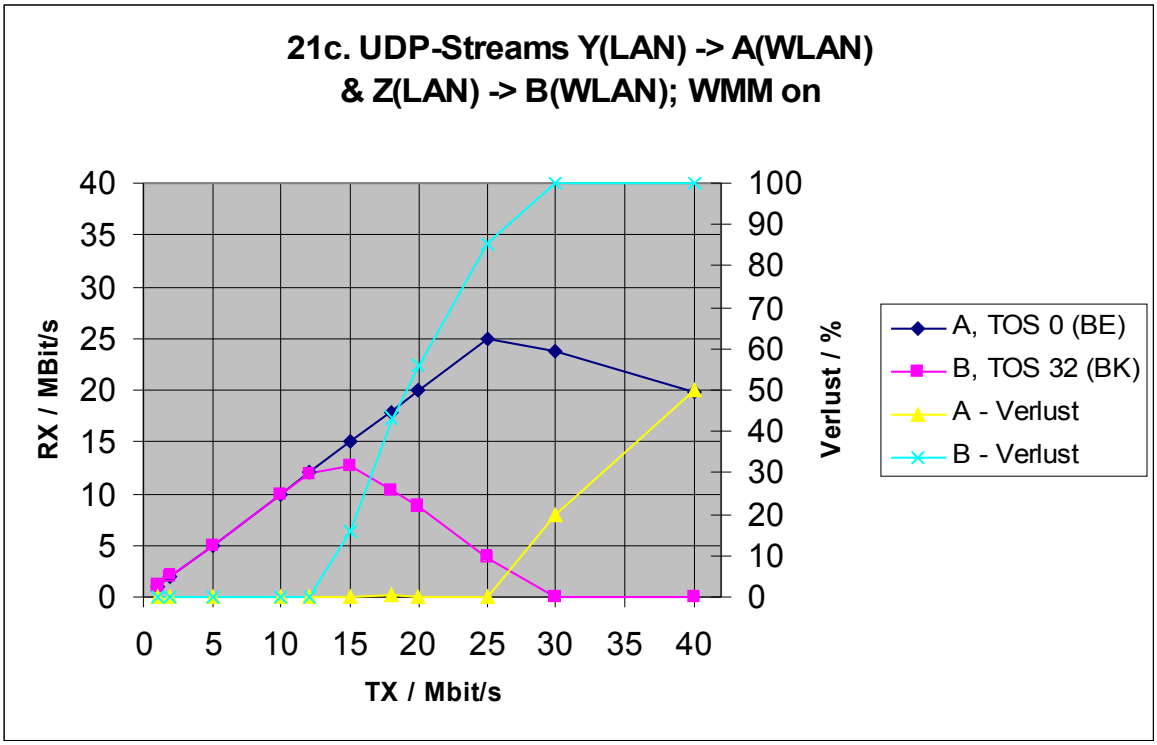

11f. UDP-Streams von A & B(WLAN) -> Z(LAN); WMM on
RX / MBit/s
Verlust / %
TX / Mbit/s
A, TOS 32 (BK)
B, TOS 224 (VO)
A - Verlust
B - Verlust

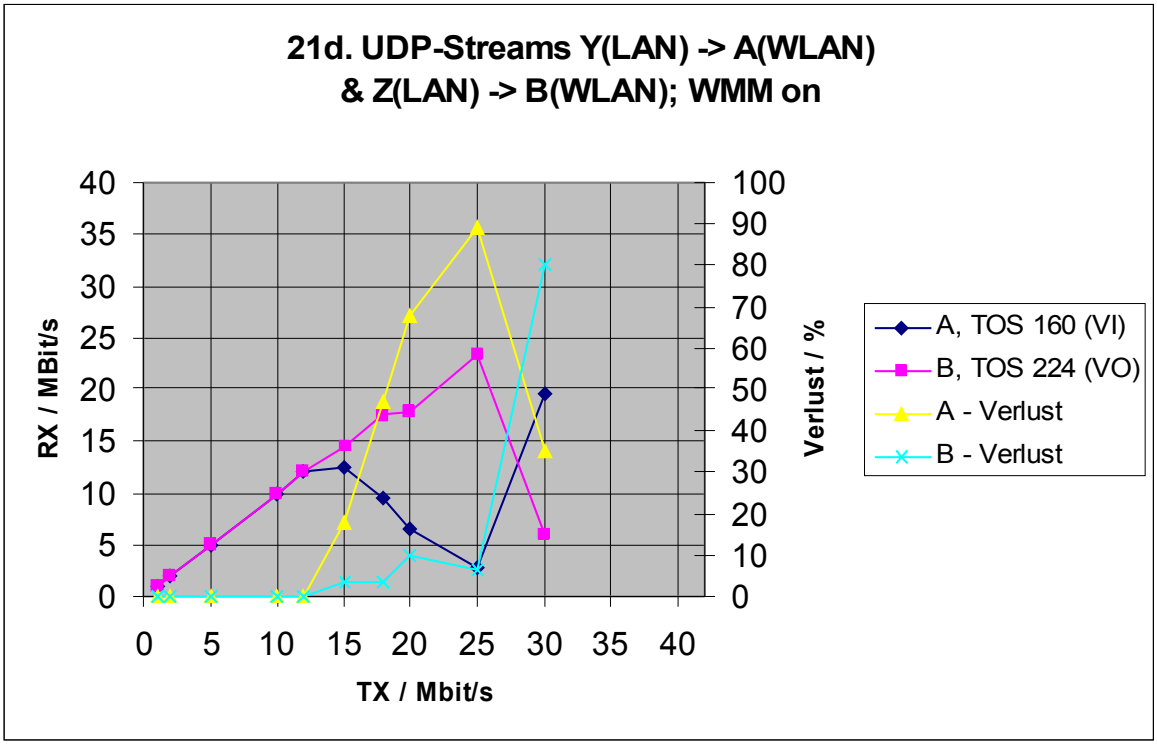

11g. UDP-Streams von A & B(WLAN) -> Z(LAN); WMM on
RX / MBit/s
Verlust / %
TX / Mbit/s
A, TOS 0 (BE)
B, TOS 160 (VI)
A - Verlust
B - Verlust


11h. UDP-Streams von A & B(WLAN) -> Z(LAN); WMM on
RX / MBit/s
Verlust / %
TX / Mbit/s
A, TOS 32 (BK)
B, TOS 160 (VI)
A - Verlust
B - Verlust


20a. UDP-Streams Y(LAN) -> A(WLAN)
& Z(LAN) -> B(WLAN); WMM off
RX / MBit/s
Verlust in %
TX / MBit/s
A, TOS 0
B, TOS 0
A - Verlust
B - Verlust


20b. UDP-Streams Y(LAN) -> A(WLAN)
& Z(LAN) -> B(WLAN); WMM off
RX / MBit/s
Verlust in %
TX / MBit/s
A, TOS 0
B, TOS 224
A - Verlust
B - Verlust


21c. UDP-Streams Y(LAN) -> A(WLAN)
& Z(LAN) -> B(WLAN); WMM on
RX / MBit/s
Verlust / %
TX / Mbit/s
A, TOS 0 (BE)
B, TOS 32 (BK)
A - Verlust
B - Verlust


21d. UDP-Streams Y(LAN) -> A(WLAN)
& Z(LAN) -> B(WLAN); WMM on
RX / MBit/s
Verlust / %
TX / Mbit/s
A, TOS 160 (VI)
B, TOS 224 (VO)
A - Verlust
B - Verlust

Anhang D

Timing-Werte aller 802.11-PHYs

PHY	*aCWmin*	*aCWmax*	*Slot time*	*SIFS time*	*PLCP Header*	*PLCP Preamble*
FHSS (1+2MBit/s)	15	1023	50 µs	28 µs	32 µs	96 µs
DSSS (1+2MBit/s)	31	1023	20 µs	10 µs	48 µs	144 µs
IR (1+2MBit/s)	63	1023	8 µs	10 µs	41 µs** 25 µs***	16 µs** 20 µs***
OFDM 5GHz	15	1023	9 µs	16 µs	4 µs	20 µs
DSSS High Rate (bis 11MBit/s)	31	1023	20 µs	10 µs	48 Bit	144 µs
OFDM 2,4GHz	31* 15	1023	20 µs* 9 µs	10 µs	4 µs	20 µs

* - bei b+g Mixed Mode

** - bei 1MBit/s

*** - bei 2MBit/s

[1]; [3]; [4]; [5]

PC-Daten

	Computer-Bezeichnungen (Buchstaben-Kürzel)			
	Y	***Z***	***B***	***A***
Name		murmansk3	murmansk	murmansk2
sonstiges	Scenic Pro M5	Scenic XL	DDS	Scenic 800
Linux-Kernel	2.6.8-3-386	2.6.15-1-486	2.6.15-1-486	2.6.15-1-486
Maschine		i686	i686	i686
KDE	-	3.5.2	3.5.1	3.5.1
CPU	Pentium MMX 233MHz	P3 700MHz	Intel P4 2,8GHz	P3 600MHz
CPU-Cache		256KB	512KB	256KB
RAM	64MB	256MB (250,91MB)	512MB (504,46MB)	256MB (250,91MB)
private IPs				
- LAN1	eth0: 192.168.1.40	eth0: 192.168.1.30	eth0: 192.168.1.20	eth0: 192.168.1.10
- WLAN		ath0: 192.168.1.31	ath0: 192.168.1.21	ath0: 192.168.1.11
MAC-Adressen				
eth0 (LAN)	00:A0:C9:91:FF:CE	00:30:05:03:8C:2D	00:0E:A6:70:D9:12	00:30:05:01:FE:40
eth1 (LAN)				00:04:76:9E:34:C6
ath0 (WLAN)		00:40:F4:A0:AF:EF	00:0F:A3:73:48:CD	00:0F:A3:73:48:CF
Software				

	Computer-Bezeichnungen (Buchstaben-Kürzel)			
	Y	***Z***	***B***	***A***
Iperf-Version	2.0.1 (08 Nov 2004)	2.0.2 (03 May 2005)	2.0.2 (03 May 2005)	2.0.2 (03 May 2005)
MadWifi- Version		svnr1500 20060412	svnr1472 20060310	svnr1472 20060310

Abkürzungsverzeichnis

AC	Access Category
ACK	Acknowledgement
AIFS	Arbitration Interframe Space
AIFSN	Arbitration Interframe Space Number
AP	Access Point
BE	Best Effort
BK	Background
CCA	Hybrid Coordination Function
CF	Contention Window
CPU	Central Processing Unit
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access Collision Detection
CTS	Clear to Send-Verfahren
CW	Contention Window
DCF	Distributed Coordination Function
DIFS	Distributed (Coordination Function) Interframe Space
DS	Differentiated Services-Feld
DSCP	Differentiated Services Code Point
ECN	Explicit Congestion Notification
ECP	Extended Capability Port
EDCA	Enhanced Distributed Channel Access
EDCAF	Enhanced Distributed Channel Access Funktion
EIFS	Extended Interframe Space
ETSI	European Telecommunications Standards Institute
HCF	Controlled Channel Access
HDTV	High Definition Television
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IFS	Interframe Space
IP	Internet-Protokoll

IPTV	Internet-Protokoll Television
ISM-Band	Industrial, Scientific and Medical Band
LAN	Local area network
LLC(-Layer)	Logical Link Control
MAC-Layer	Medium Access Control-Layer
MHz	Megahertz
MiB	Mebibyte (2^{20} = 1024^2 = 1.048.576 Byte), siehe Norm IEC 60027-2
MSDU	Mac Service Data Unit
NAV	Network Allocation Vector
nQSTA	non QoS Station (im WLAN)
NTP	Network Time Protocol
OFDM	Orthogonal Frequency Division Multiplexing
OSI	Open Systems Interconnection
PCF	Point Coordination Function
PCI	Protocol Control Information
PHB	Per Hop Behavior
PHY(-Layer)	Physical Layer
PIFS	Point (Coordination Function) Interframe Space
PLCP	Physical Layer Convergence Protocol
PSK	Preshared Key
QoS	Quality of Service
RAM	Random Access Memory
RFC	Request for Comments
RTP	Real-Time Transport Protocol
RTS	Request to Send
RX	Empfangen
SCP	Secure Copy
SIFS	Short Interframe Space
SLRC	Station Long Retry Count
SSRC	Station Short Retry Count
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
TOS	Type of Service-Feld

TX	Senden
TXOP	Transmission Opportunity
UDP	User Datagram Protocol
VI	Video
VLAN	Virtual Local Area Network
VO	Voice
VoIP	Voice over IP
WAN	Wide Area Network
Wi-Fi Alliance	Wireless Fidelity Alliance
WLAN	Wireless Local Area Network
WME	Wireless Media Extensions
WMM	Wireless Multimedia
WPA	Wireless Fidelity Protected Access

Quellenverzeichnis

[1] Rech, Jörg:
Wireless LANs – 802.11-WLAN-Technologie und praktische Umsetzung im Detail.
Hannover: Heise Verlag, 2004

[2] IEEE:
802.11 Timelines
http://grouper.ieee.org/groups/802/11/802.11_Timelines.htm

[3] IEEE:
IEEE Wireless LAN Edition
A compilation based on IEEE Std 802.11TM-1999 (R2003) and its amendments.

[4] IEEE:
IEEE Std 802.11a-1999(R2003)

[5] IEEE:
IEEE Std 802.11g™-2003

[6] IEEE:
P802.11e/D13.0, January 2005

[7] Wi-Fi Alliance:
Wi-Fi CERTIFIED™ for WMM™
Support for Multimedia Applications with Quality of Service in Wi-Fi® Networks – WMM-Whitepaper
2004

[8] Grebe, Andreas:
Unterlagen zur Vorlesung Datennetze I und II
2004/2005

[9] Wikipedia Foundation:
http://de.wikipedia.org

[10] Haden, Rhys:
http://www.rhyshaden.com/ethernet.htm

[11] IEEE:
IEEE Std 802.1D™- 2004

[12] RFC 791:
http://www.ietf.org/rfc/rfc791.txt

[13] RFC 2474:
http://www.ietf.org/rfc/rfc2474.txt

[14] RFC 3168:
http://www.ietf.org/rfc/rfc3168.txt

[15] Merrit, Rick:
Bei QoS trennen sich die Wege der WLAN-Hersteller
http://www.eetimes.de/at/news/showArticle.jhtml?articleID=19503430
2003

[16] http://www.wifialliance.com/white_papers/whitepaper-120505-wmmpowersave/

[17] Seattle Wireless
LinksysWrt54g.
http://www.seattlewireless.net/index.cgi/LinksysWrt54g

[18] Linksysinfo.org
http://www.linksysinfo.org

[19] http://dast.nlanr.net/Projects/Iperf/

[20] Iperf-Dokumentation zu Version 1.7.0:
http://dast.nlanr.net/Projects/Iperf/iperfdocs_1.7.0.php

[21] Iperf-Hilfe:
Shell-Ausgabe von „iperf -h“

[22] Piller, Udo:
Unterlagen zur WLAN-Vorlesung
2004/2005

Foto des Linksys WRT54G in Abbildung 7 und 18
→ Linksys (http://www.linksys.de)