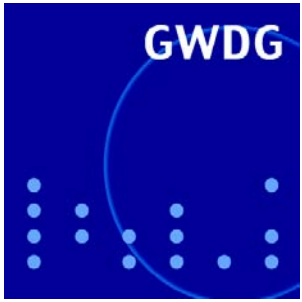

GWDG-Bericht Nr. 61

Sebastian Rieger

Streaming-Media und Multicasting in drahtlosen Netzwerken

Untersuchung von Realisierungs- und Anwendungsmöglichkeiten

Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

Sebastian Rieger

Streaming-Media und Multicasting in drahtlosen Netzwerken

Untersuchung von Realisierungs- und Anwendungmöglichkeiten

Sebastian Rieger

Streaming-Media und Multicasting in drahtlosen Netzwerken

Untersuchung von Realisierungs- und Anwendungmöglichkeiten

GWDG-Bericht Nr. 61
Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen
Am Faßberg
D-37077 Göttingen
Telefon: 0551-201-1510
Telefax: 0551-21119
E-Mail: gwdg@gwdg.de

Satz: Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

Druck: Offset- und Dissertations-Druck Jürgen Kinzel, Göttingen-Weende

ISSN 0176-2516

Inhalt

Vorwort

Wenn ich auf die vergangenen Monate zurückblicke, drängt sich mir die Vermutung auf, dass der Satz „Ich schreibe an meiner Diplomarbeit“, den ich manchen Mitmenschen als Erklärung und besonders in der Endphase auch als Entschuldigung angeboten habe, in dieser Form nicht ganz korrekt war. Eigentlich habe ich nicht einfach an ihr geschrieben - sie ist gewissermaßen gewachsen. Nur so werden die vielen Korrekturen und das Verwerfen von Sätzen, Absätzen und sogar Kapiteln mit einbezogen. Insgesamt sehe ich auch jetzt noch die abgeschlossenen Textpassagen aus zwei Perspektiven. Zum einen bin ich zufrieden, einen stabilen Text vor mir zu sehen, zum anderen möchte ich vieles noch weiter erläutern, andere Ansätze ausprobieren usw. Das ist wohl ein Kompromiss, der für Diplomarbeiten gewöhnlich ist. Wahrscheinlich hat auch dies mit einem „Wachsen“ der Arbeit zu tun. Die Ideen für die Diplomarbeit sind sicherlich auch ein gutes Stück aus meinem Umfeld und vor allem meinem Studium gewachsen. Daher möchte ich mich als erstes bei meinen Dozenten bedanken, die mich über die Semester hinweg mit wichtigen Theorien der Informatik vertraut gemacht haben. Auch meinen Studienkollegen, die mir in zahlreichen Diskussionen Ideen eingepflanzt haben, gilt mein Dank. Unter den Dozenten möchte ich vor allem Prof. Dr. Anatol Badach hervorheben. Er hat mich nicht nur bei dieser Arbeit, sondern auch sonst im Studium an vielen Stellen immer wieder unterstützt und damit meine akademische Laufbahn geprägt. Ohne ihn und seine langjährige Erfahrung hätte diese Arbeit sicherlich ein ganz anderes Gesicht bekommen.

Auch Herrn Prof. Dr. Hartmut Koke sowie allen Mitarbeitern der GWDG gilt mein Dank, die mich während der Zeit meiner Diplomarbeit in Wort und Tat unterstützt haben. Herr Koke regte über die Zeit hinweg immer wieder interessante Projekte in der Praxis an, die hoffentlich dazu beigetragen haben, dass meine Arbeit nicht zu theorielastig ist.

Danken möchte ich auch dem Internet. Viele Informationen, die ich für diese Arbeit dringend benötigte, waren in Büchern nicht zu finden. Nicht zuletzt durch das Internet konnte ich schließlich viele Lücken füllen.

Natürlich möchte ich mich auch bei meiner Familie und meinen Freunden bedanken. Eine Diplomarbeit wachsen zu lassen, ist keine Tätigkeit, bei der nur Wörter, Bilder und Bücher benötigt werden.

Kurzbeschreibung

Die vorliegende Arbeit befasst sich mit zwei sehr zukunftsträchtigen Bereichen aus der Welt der Informatik. Zum einen mit dem Bereich Streaming-Media, der vorrangig die Übertragung von Audio und Video sowie anderen Informationen in Echtzeit umfasst. Zum anderen mit dem Bereich WLANs respektive WPANs, der den immer größer werdenden Trend zu mobilen „wireless"-Anwendungen unterstützt. In den folgenden Kapiteln werden diese beiden Anwendungspotentiale miteinander verbunden und aneinander angepasst. Dabei werden auch die möglichen Probleme, wie z. B. Sicherheitsrisiken beim Betrieb von WLANs, aufgezeigt und für den konkreten Anwendungsfall Problemlösungen aufgezeigt. Sowohl der Bereich WLANs als auch der Bereich Streaming-Media wird dabei soweit vertieft, dass der Leser alle wichtigen Protokolle und zugrunde liegenden Techniken für den erfolgreichen Einsatz beider Technologien vermittelt bekommt. Über das reine Senden von Streaming-Media-Daten an mehrere Clients (bekannt als Multicasting) im WLAN hinweg zeigt das Kapitel 5 außerdem eine Möglichkeit auf, den gesendeten Content noch effizienter zu verteilen. Dabei wird durch zusätzliche Caching-Verfahren der Empfang der Streaming-Media-Informationen aus Sicht des Endnutzers beschleunigt. Abschließend zeigt der Abschnitt *Anwendungsszenario „LCDNs" im WLAN-Verbund* ein mögliches Szenario auf, wie z. B. zukünftig in großen öffentlichen WLANs (genannt HotSpots) etwa an Flughäfen oder Universitäten eine effiziente Verteilung erreicht werden kann. Dabei werden kleine, lokale CDNs am Übergang zwischen den WLAN-Zellen installiert.

Dem versierten Leser sei die Lektüre ab dem Kapitel 4 empfohlen. In diesem Kapitel werden die einzelnen Techniken, Streaming-Media, Multicasting und drahtlose Netzwerke zusammengeführt und die Kernaussagen dieser Arbeit herausgestellt. Bei der Lektüre dieses Kapitels lässt sich auch abschätzen, wie es um die Realisierungs- und Anwendungsmöglichkeiten bestellt ist. Außerdem sind an allen wichtigen Stellen Verweise auf die einzelnen Techniken (Kapitel 1 bis 3) vermerkt. Bei dem ein oder anderen Verweis sollten auch die schnellsten Leser, die sich bereits gut mit dem Thema auskennen, einmal das Auge schweifen lassen. Einige Aussagen dieser Arbeit sind aufgrund ihrer engen Verzahnung mit der verbundenen Technik in dem jeweiligen Kapitel statt im Kapitel 4 erwähnt.

Einleitung

Als Tim Berners Lee 1990 am CERN durch den Entwurf des Protokolls HTTP den Grundstein zur Verbreitung des WWW und damit des Internet legte, dachte dabei wohl kaum jemand an die Übertragung von Datenmengen jenseits von ein paar Megabyte. Seit dem Jahr 1990 ist jedoch mit der Verbreitung des Internet auch die Erwartung der Endbenutzer gestiegen. Waren es zunächst nahezu ausschließlich Wissenschaftler, die im WWW Informationen nutzen konnten, sind es heute auch Firmen und Privathaushalte, die einen völlig anderen Anspruch an das Internet als Medium haben. Wissenschaftliche Dokumentationen, Diplomarbeiten und sonstige Textinformationen erwecken weniger ihr Interesse, als Dienste wie Videoübertragungen, -konferenzen, Spiele oder sonstige Multimedia-Präsentationen. Auch wenn der Begriff Multimedia bereits abgegriffen erscheint, ist er zu einer Art Erfolgsgeheimnis des WWW geworden. „Das Auge des Surfers isst mit“ – so müssen Webseiten heute nicht nur aufwendige Grafik, sondern auch Videoinformationen enthalten. Außerdem möchte der private Internetnutzer z. B. Radio direkt aus dem Web hören oder nebenbei mit einem Bekannten eine Videokonferenz durchführen. Auch im professionellen Umfeld, in Unternehmen, werden Videokonferenzen über das Web abgehalten oder die Mitarbeiter per Video geschult. Letzteres findet im Bereich E-Learning auch an Hochschulen immer größeres Interesse. So werden beispielsweise an der Georg-August-Universität Göttingen, gefördert durch das BMBF, immer mehr Hörsäle mit Videokonferenzsystemen ausgestattet. Diese Entwicklung wird in der Informatik häufig unter dem Begriff Streaming-Media zusammengefasst. Streaming-Media umfasst dabei Daten,

die dem Benutzer direkt während des Empfangs präsentiert werden. Die Information (z. B. Audio oder Video) ist „on demand“ und „in real time“ (in Echtzeit) verfügbar.

Neben den oben genannten Anforderungen an das Internet spielt für die Zukunft eine weitere Entwicklung eine große Rolle. Die Verbindung zum Internet wird immer flexibler. Während man sich in der Anfangsphase ausschließlich per Modem ins Internet einwählen konnte, werden heute neben Breitbandzugängen für Firmen und Privathaushalte auch immer mehr mobile Zugänge realisiert. Nicht nur auf Messen, Flughäfen oder auf dem Universitäts-Campus werden WLANs für den mobilen Internetzugang installiert, auch der Mobilfunkbereich konvergiert zunehmend mit dem Internet. Durch Anwendungen wie GPRS, HSCSD oder UMTS nähert sich das Handy immer mehr einem mobilen Internet-Browser an. Zukünftige Entwicklungen, wie z. B. der Standard IEEE 802.15, beschreiben dabei sogar schon sog. WPANs, die die Wireless-Technologien Bluetooth, 802.11 und Mobilfunkanwendungen wie UMTS zu einem großen Netz verbinden. Dabei unterstützen in diesem Szenario die Endgeräte den Zugang zu all diesen Technologien und können sich je nach Verfügbarkeit an ihre Umgebung anpassen. So kann ein Endbenutzer mit seinem Notebook in einer Stadt, in der ein WPAN verfügbar ist, diese günstige und schnelle Verbindung ins Internet nutzen und beim Verlassen der Stadt automatisch per Mobilfunk auf eine teurere und ggf. langsamere Verbindung vermittelt werden.

Wie auch immer die zukünftige Entwicklung aussehen wird, sie wird „wireless“ sein und damit dem Verlangen des Benutzers nach Flexibilität und räumlicher Unabhängigkeit nachkommen. Somit wird eine Schlüsselfunktion für zukünftige Netze die Integration von Streaming-Technologie in „wireless“ Networks sein. Die vorliegende Arbeit zeigt Kapitel für Kapitel die dafür notwendigen Techniken auf.

1. Streaming-Media

Unter Streaming-Media versteht man den Einsatz von Verfahren, die Audio und Video im Internet zur Verfügung stellen. Dabei können neben Audio und Video theoretisch auch andere Informationsarten übertragen werden.

Unterschiedliche Informationsarten werden in der Regel als Medien bezeichnet. Nicht zuletzt durch das Schlagwort „*Multimedia*“ wurde vor einigen Jahren der Begriff Medium (*media*) als Bezeichnung für unterschiedliche Arten von Streaming-Media wie Audio, Video und Daten geprägt. Dabei wird im Allgemeinen zwischen diskreten und kontinuierlichen Medien unterschieden. Während diskrete Medien lediglich für einen einzelnen Zeitabschnitt Informationen enthalten, liefern kontinuierliche Medien fortlaufend Informationen in der Regel in Echtzeit.

Der Begriff *Streaming* bezieht sich auf die Art und Weise, wie die Medien transportiert werden. Dabei werden die Medien als kontinuierlicher Strom von IP-Paketen (*Stream*) über das Internet zum Anwender transportiert. In diesem Zusammenhang spricht man von einem *Medienstrom*. Handelt es sich bei dem Medienstrom um die Übertragung von Audio, bezeichnet man diesen Vorgang als *Audio-Streaming*. Wird durch den Medienstrom Video übertragen, spricht man dementsprechend von *Video-Streaming*.

1.1 Grundlagen von Streaming-Media

Streaming-Media bezeichnet häufig die Wiedergabe von Audio und Video. Die grundlegenden Aspekte von Streaming-Media sind jedoch losgelöst von diesen beiden klassischen Medien. Grundsätzlich lassen sich alle Medien als Stream übertragen. So auch Text (z. B. Newsticker), Bilder (z. B. progres-

sive JPEG-Bilder) oder jede andere Form von Daten, die als Paketstrom mit Zeitstempeln versendet werden kann, dessen einzelne Pakete dem Benutzer sofort präsentiert werden sollen. Da das Internet paketvermittelt arbeitet, bezeichnet der Begriff IP-Paket dabei eine Informationseinheit für einen (diskreten) Zeitabschnitt. Die kontinuierliche Übertragung solcher IP-Pakete bezeichnet man als *Stream*.

Bei Streaming-Media handelt es sich um eine gleichzeitige Übertragung von verschiedenen Medien (Audio, Video, Daten), die einen kontinuierlichen Medienstrom in Echtzeit bilden, wobei die einzelnen übertragenen Medien dem Anwender sofort präsentiert werden. Die Übertragung von Multimedia-Inhalten fällt genau dann unter den Begriff Streaming-Media, wenn mindestens ein kontinuierliches Medium (wie Audio oder Video) darin enthalten ist.

Streaming-Media zeichnet sich durch die folgenden Eigenschaften aus:

- der übertragene Medienstrom ist kontinuierlich,
- der Medienstrom wird unmittelbar (während des Empfangs) wiedergegeben,
- der Medienstrom wird nicht zunächst komplett heruntergeladen, allerdings ggf. lokal gepuffert,
- der Medienstrom ist kontrollierbar, z. B. um vor- bzw. zurückzuspulen.

Durch diverse *Streaming-Protokolle* hat der Betrachter eines Medienstroms die Möglichkeit, Einfluss auf die Wiedergabe der Medien zu nehmen. So lässt sich der Medienstrom in der Regel anhalten sowie vor- und zurückspulen. Auch komplexere Anwendungen, wie das Aufnehmen oder das Kombinieren von verschiedenen Medienströmen, wird von den Protokollen definiert.

Streaming-Media kann entweder „live“ (unmittelbar während der Aufnahme) oder als Konserve (zu einem späteren Zeitpunkt als Aufzeichnung) an den Betrachter geliefert werden. Im ersten Fall spricht man von einem *Live-Stream*, während der zweite Fall als *Konserven-Stream* bezeichnet wird. Abbildung 1-1 zeigt ein Beispiel für einen Live-Stream bei dem die Eingabedaten direkt kodiert und übertragen werden. Ein Konserven-Stream gibt eine zuvor abgespeicherte Datei wieder. Live-Streams werden z. B. von Fernsehsendern eingesetzt, um das laufende Fernsehprogramm o. ä. wiederzugeben. Konserven-Streams können auch Reportagen usw. wiedergeben. Der Aufwand für ein Live-Stream ist etwas höher, da hier die Eingabedaten

sehr schnell kodiert werden müssen, während bei einem Konserven-Stream die Daten bereits kodiert aus der Datei übertragen werden können.

Die häufigsten Streaming-Media-Anwendungen im Web sind:

- Web-Radio

  Viele Radiosender senden ihr Hörfunkprogramm mittlerweile auch über das Internet, was als Web-Radio bezeichnet wird. Dabei wird das laufende Radio-Programm direkt als Live-Stream ins Internet gesendet. Durch das Web lassen sich Hörer in der ganzen Welt erreichen. Häufig bieten Web-Radios außerdem auf einer zugehörigen Web-Seite zusätzliche Informationen zum Hörfunkprogramm, wie Nachrichten, gerade gespielter Titel und Interpret usw. an.

- Web-TV

  Analog zu den Radiosendern bieten auch Fernsehsender vermehrt einzelne Berichte im Internet an. Dabei werden diese Berichte vorrangig als Konserven-Stream gesendet. Auch Live-Streams des laufenden Fernsehprogramms findet man vereinzelt. Wie die Radiosender, bieten auch die Fernsehsender auf den zugehörigen Web-Seiten eine Fülle von Zusatzinformationen zum Programm.

- Videokonferenzen, E-Learning uvm. sind in Abschnitt 1.9 beschrieben.

1.1.1 Funktionskomponenten bei Streaming-Media

Die Abbildung 1-1 zeigt, welche Funktionskomponenten bei Streaming-Media zum Einsatz kommen. Dabei wurde als Beispiel eine Audio-Übertragung im Internet dargestellt (z. B. Web-Radio).

Der *Kodierer* digitalisiert das Medium (hier Sprache), das z. B. über ein Mikrofon eingegeben wird und bringt die Audio-Information in das gewünschte *Format*. Dabei wird die Information zusätzlich komprimiert und auf das Übertragungsnetz angepasst (in IP-Pakete aufgeteilt). Die Streaming-Media-Anwendung auf der Seite des Senders, bezeichnet man auch als *Media-Encoder.*

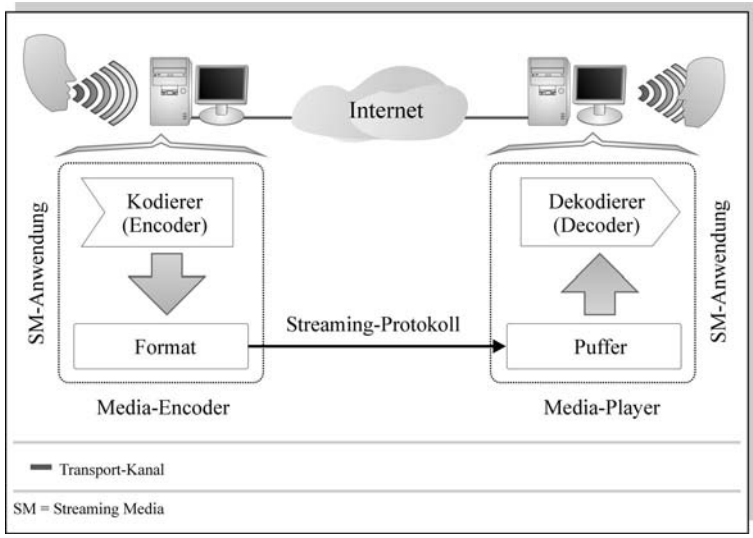

Abb. 1-1: Beispiel einer Audio-Übertragung über das Internet

Bei der Übertragung im Netz kommt ein *Streaming-Protokoll* zum Einsatz, das neben der Adressierung des Empfängers im Netz z. B. auch für eine schnelle und sichere Übertragung sorgt. Protokoll und Format können dabei auch einen Verbund aus mehreren Protokollen und Formaten darstellen. Auf der Seite des Empfängers werden die empfangenen Pakete zunächst in einem *Puffer* zwischengespeichert, den man auch als *Jitter-Puffer* bezeichnet. Dadurch können Verzögerungen bei der Übertragung (*Jitter*) ausgeglichen werden. Puffer sorgen damit für eine möglichst flüssige Wiedergabe des Medienstroms. Im Einzelfall kann auch der Kodierer einen Puffer einsetzen, um eine gleichmäßige Übertragung zu erreichen.

Der *Dekodierer* sorgt schließlich für die Dekomprimierung des Audio-Signals und dessen Wiedergabe z. B. über einen Lautsprecher. Die Wiedergabe des übertragenen Mediums erfolgt kontinuierlich und in Echtzeit. In Zusammenhang mit der Streaming-Media-Anwendung auf der Seite des Empfängers spricht man häufig von einem *Media-Player*.

Die Anforderungen des Mediums stellen sich an die gesamte Kette von Komponenten. So muss der Kodierer beim Sender die Eingabedaten in ausreichend schneller Zeit in das gewünschte Format und die gewünschte Kompression bringen. Auf der anderen Seite muss der Dekodierer des

Empfängers die Daten ausreichend schnell wieder dekomprimieren und anzeigen. Auch das verwendete Format selbst sollte hinreichend effizient sein und ebenso wie das verwendete Streaming-Protokoll nicht unnötigen Overhead übertragen.

Die schwächste Komponente (z. B. bezogen auf das größte Delay) in dieser Kette bestimmt die letztendlich verfügbare Qualität des Mediums.

1.1.2 Anforderungen an das Internet

Die Übertragung von Audio und Video stellt hohe Anforderungen an das Internet. Die übertragenen Datenmengen beim Besuch einer Web-Seite werden durch die zusätzlichen Bilder, Animationen, Audio-Inhalte oder Videos immer größer. Der daraus resultierende Engpass bei der Übertragung der Medienströme kann nicht allein durch schnelle Netzanschlüsse für die Endbenutzer reduziert werden. Auch die Protokolle und Formate, die bei der Übertragung von verschiedenen Echtzeitmedien zum Einsatz kommen, müssen effizient ausgewählt und eingesetzt werden. Noch wichtiger ist die Anpassung der Struktur des Internet als Backbone an die erhöhten Anforderungen. Ein Broadcast-Verfahren, wie es z. B. beim Fernsehen oder Radio zum Einsatz kommt, lässt sich im Internet nicht realisieren.

Wie in Abbildung 1-2 zu erkennen ist, unterscheidet sich die Übertragung von Audio und Video im Internet stark vom klassischen Broadcast-Verfahren (beim Radio oder Fernsehen). Im Web werden die Medienströme an jeden Zuschauer einzeln gesendet (*Unicast*).

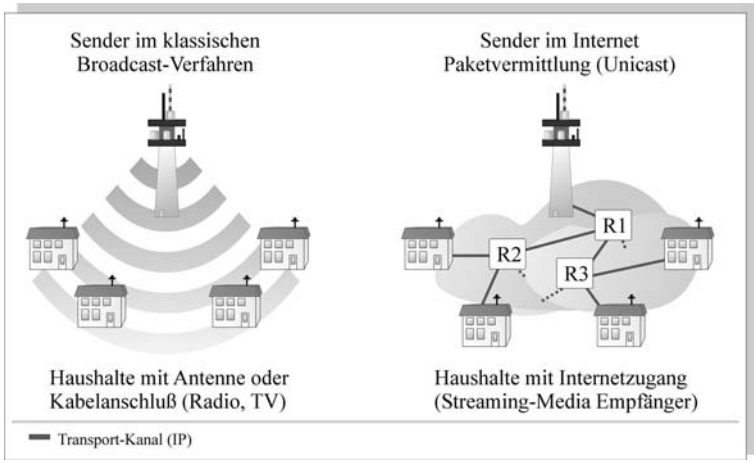

Abb. 1-2: Unterschied: klassische Übertragung von Audio und Video und Streaming-Media im Web

Für den Zuschauer ergibt sich somit für den Server eine maximal benötigte Bandbreite von:

*max. benötigte Bandbreite = n * Bandbreite des Streams*

Auch an den Routern zwischen Sender und Zuschauer wird die n-fache Bandbreite benötigt. Während R2 und R3 „nur“ die 2-fache Bandbreite benötigen, muss an R1 die 4-fache Bandbreite des übertragenen Streams zur Verfügung stehen. Daher müssen neue Infrastrukturen im Netz (sog. Multicasting-Netze) etabliert werden, um die effiziente Verteilung von Audio und Video an mehrere Teilnehmer zu unterstützen. Multicasting-Netze sind im Kapitel 2 ausführlich beschrieben.

Die Anforderungen von Streaming-Media im Internet werden vom Empfänger in erster Linie an die Übertragung gestellt. Beispielsweise möchte ein Zuschauer eines Video-Streams nicht lange auf den Beginn der Übertragung warten und eine möglichst flüssige Wiedergabe erhalten. Diese Anforderungen beziehen sich vor allem auf *Quality-of-Service(QoS)*-Eigenschaften der Verbindung. QoS definiert Übertragungseigenschaften im Netzwerk und garantiert in der Regel akzeptable Mindestwerte. Beim QoS werden folgende Eigenschaften unterschieden:

- Bandbreite / Bitrate
- Verzögerung (*delay*)
- Verzögerungsabweichung (*jitter*)
- Fehlerrate

Diese Anforderungen stehen in direktem Zusammenhang mit der übertragenen Information. So wird z. B. für sehr große Informationsmengen (z. B. ein hochauflösendes Video) eine große Bitrate benötigt und für hohe Bildwiederholraten z. B. ein geringes *delay* und ein geringer *jitter*. Eine genauere Beschreibung des QoS findet sich in [BADA_01].

1.1.3 Streaming-Protokolle

Um den Transport von Streaming-Media zum Anwender zu ermöglichen, werden spezielle Protokolle benötigt, die sog. Streaming-Protokolle. Bei den Streaming-Protokollen wird zwischen *Kontroll-* und *Transportprotokollen* unterschieden. Kontrollprotokolle ermöglichen es dem Anwender, den Medienstrom zu kontrollieren (z. B. vor und zurück zu spulen). Transportprotokolle versuchen eine möglichst flüssige Wiedergabe zu erzielen. Um

dies zu erreichen, siedeln sich die Streaming-Transportprotokolle innerhalb des TCP/IP möglichst nahe der Transportschicht (Schicht 3) an. Dabei nutzen sie dessen Protokoll UDP (User Datagram Protokoll) für die Übertragung von Streaming-Inhalten. UDP bietet im Vergleich zum TCP (Transmission Control Protocol) sehr viel geringere Verzögerungszeiten. Die Tatsache, dass diese geringen Verzögerungszeiten mit einer höheren Fehlerrate erkauft werden, ist für Streaming-Media-Anwendungen unerheblich, da die erneute Übertragung z. B. von fehlerhaften Bildern zu einem späteren Zeitpunkt für den Betrachter sehr viel störender ist, als die Kaschierung von eventuell fehlenden Bildern. Außerdem bietet UDP eine von Anfang an hohe Durchsatzrate, im Gegensatz zum TCP („slow-start"-Problem [BADA_01]).

Streaming-Protokolle definieren ein Verfahren, nach dem auch Media-Player von unterschiedlichen Herstellern den Medienstrom eines Servers wiedergeben können. Leider existiert, wie so oft, auch hier kein einheitlicher Standard. Somit wird man, um Streaming-Media im Internet nutzen zu können, wohl noch einige Zeit unterschiedliche Media-Player installieren müssen.

Die am häufigsten eingesetzten Streaming-Protokolle im Web sind:

- RTSP (Real Time Streaming Protocol), ein offenes Protokoll, welches in Abschnitt 1.6 genauer beschrieben wird,
- MMS (Microsoft Media Server), das in Abschnitt 1.7 behandelt wird,
- sowie im Einzelfall HTTP (siehe Abschnitt 1.5.).

1.2 Medien und ihre Eigenschaften

Die einzelnen Medien und ihre Eigenschaften stellen sehr unterschiedliche Anforderungen an das Streaming. Medien lassen sich, bezogen auf die Zeit als Dimension, in zwei Klassen aufteilen:

- diskrete Medien (Text, Standbild usw.)
- kontinuierliche Medien (Audio, Video usw.)

Diskrete Medien sind für den Bereich Streaming-Media weniger interessant, da sie nur für einen einzelnen Zeitabschnitt Informationen enthalten. Kontinuierliche Medien dagegen, bieten Informationen für beliebig viele Zeitabschnitte. Daher bieten sie sich hervorragend für Streaming an, indem jeweils ein Zeitabschnitt „quasi diskret" als IP-Paket im Stream übertragen wird. Daraus resultiert die Dominanz von Audio und Video im Bereich Streaming-Media.

1.2.1 Eigenschaften von Audio

Als Audio bezeichnet man jede Art von Information, die der Mensch über das Ohr wahrnehmen kann. Es ist somit zwischen mehreren Arten von Audio zu unterscheiden. Die wichtigsten Arten von Audio sind:

- Sprache
- Musik
- Töne, Klänge, Geräusche

Abbildung 1-3 zeigt die Audio-Arten in Zusammenhang mit dem von ihnen abgedeckten Frequenzbereich.

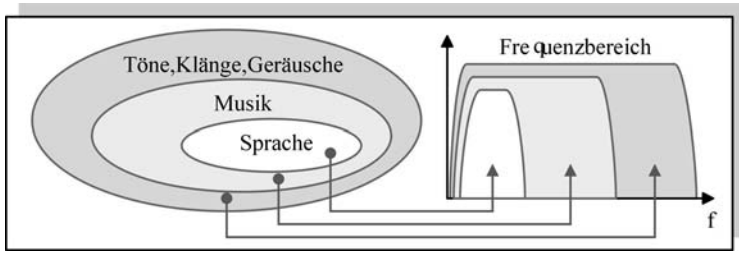

Abb. 1-3: Audio-Arten und ihre Frequenzen

Während das menschliche Ohr Frequenzen bis ca. 20 kHz wahrnehmen kann, nutzt beispielsweise die Sprache in der Regel nur den Frequenzbereich zwischen 300 und 3800 Hz. Dabei ist jedoch nicht nur die Frequenz ein Unterscheidungsmerkmal, sondern auch der Gehalt der Information. Sprache weist einen weitaus geringeren Informationsgehalt auf als Musik, da im Wesentlichen immer die gleichen Laute kombiniert werden. Durch diese unterschiedlichen Merkmale der Klassen lassen sich diese sehr individuell komprimieren.

Bevor die Informationen jedoch komprimiert werden können, sollten die Anforderungen, die bei der Kompression eingehalten werden müssen, definiert werden. Ein Zuhörer stellt zwei Anforderungen an das gehörte Audio:

- Anforderungen an die Güte (Qualität) des Audio
- Anforderungen an die Übertragung des Audio (QoS)

Die Anforderungen an die Güte des Audio sind abhängig vom Anwendungsfall. Beim Radiohören während der Arbeit erwartet der Zuhörer beispielsweise weniger Qualität als beim Musikhören auf dem Wohnzimmersofa. Diese Anforderungen können bereits vor der Kompression die Datenmenge

stark reduzieren. Neben der Fokussierung auf bestimmte Frequenzen lässt sich dabei auch die Auflösung des Audio-Signals anpassen (Abtastfrequenz, Digitalisierungsstufen) oder die Anzahl der Kanäle reduzieren (z. B. Mono statt Stereo).

Die Anforderungen an die Übertragung von Audio lassen sich in erster Linie auf die in Abschnitt 1.1.1 aufgezeigten QoS-Parameter zurückführen. Insbesondere sind für Audio folgende QoS-Parameter wichtig:

- Bitrate (ausreichend für die Anforderungen an die Güte)
- Delay (besonders bei bidirektionaler Übertragung z. B. Internet-Telefonie)
- Jitter (Vermeidung von „Knacksen“ und „Aussetzern“ bei der Wiedergabe)

Die wichtigsten Anwendungen für Audio-Streaming im Web sind:

- Internet-Radio (Live-Streaming, Berichte als „Tonkonserven“, …)
- Übertragung von Musikstücken (Hörprobe beim Kauf von CDs, Hintergrundmusik einer Homepage, Peer-to-Peer Tauschbörsen, …)
- Internet-Telefonie
- Übertragung von Vorträgen im Web

1.2.2 Eigenschaften von Video

Als Video bezeichnet man jede Art von Information, die der Mensch über das Auge wahrnehmen kann. Video lässt sich in verschiedene Klassen aufteilen. Die Abbildung 1-4 zeigt die wichtigsten Klassen von Video, die bei der Übertragung über das Internet von großer Bedeutung sind. Diese Klassen unterscheiden sich vorrangig durch ihr erzeugtes Datenvolumen und ihre Kompressionsfähigkeit. Grundsätzlich fällt unter den Begriff Video jegliche Wiedergabe von bewegten Bildern. Im einfachsten Fall ist dies ein Fernsehbild bzw. Vollbild-Video, in dem komplette Bilder pro Zeiteinheit übertragen werden. Anders z. B. bei Präsentationen und Multimedia-Shows. Hier werden zwar auch Filme übertragen, allerdings sehr viel mehr statische Bildbereiche (Titel, Text) oder Animationen verwendet. Animationen bestehen ihrerseits aus bewegten grafischen Primitiven (Rechtecke, Kreise, Linien usw.). Dabei nimmt die entstehende Datenmenge bei Präsentation und Animation im Vergleich zum Vollbild-Video stark ab.

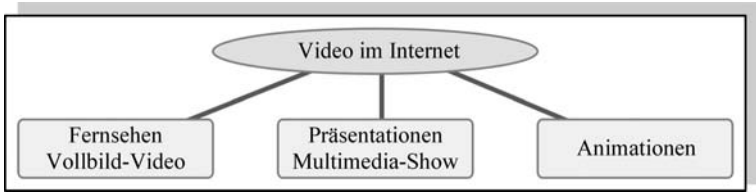

Abb. 1-4: Aufteilung von Video-Klassen im Internet

Ein Zuschauer stellt zwei Anforderungen an ein übertragenes Video:

- Anforderungen an die Güte (Qualität) des Videos
- Anforderungen an die Übertragung des Videos (QoS)

Die Anforderungen an die Güte des Videos sind abhängig vom Anwendungsfall. Beim Fernsehen erwartet der Zuschauer beispielsweise weniger Qualität als beim Verfolgen einer Online-Vorlesung, bei der er auch den Tafelanschrieb erkennen möchte. Diese Anforderungen können bereits unabhängig von der Kompression die Datenmenge stark reduzieren. Dabei lassen sich z. B. Auflösung, Farbtiefe und Bilder pro Sekunde reduzieren.

Die Anforderungen an die Übertragung des Videos lassen sich in erster Linie auf die in Abschnitt 1.1.1 aufgezeigten QoS-Parameter zurückführen. Insbesondere sind für Video folgende QoS-Parameter wichtig:

- Bitrate (ausreichend für die Anforderungen an die Güte)
- Delay (besonders bei bidirektionaler Übertragung, z. B. Video-Konferenzen)
- Jitter (damit keine „Aussetzer“ bei der Wiedergabe entstehen)

Die wichtigsten Anwendungen für Video-Streaming im Web sind:

- Internet-TV (Live-Stream, Berichte, …)
- Übertragung von einzelnen Videos (Video-Filme, Kino-Trailer, …)
- Internet-Videokonferenzen
- E-Learning (z. B. Übertragung von Vorlesungen im Web)

1.3 Audio-Streaming im Internet

Das Audio-Streaming im Internet basiert auf einem einheitlichen System, das von nahezu allen Media-Encodern und Media-Playern eingehalten wird

(siehe Abschnitt 1.3.1.). Dabei spielt neben der einheitlichen Struktur auch ein einheitliches Audio-Format eine große Rolle. Solche Audio-Formate beschreiben nicht nur die Eigenschaften des transportierten Audio-Signals, sondern auch dessen Kompression. Wichtige Audio-Formate, die im Web verwendet werden, sind in der Tabelle 1-2 im Abschnitt 1.3.3 aufgeführt. Bei der Kompression der Signale verwenden die einzelnen Audio-Formate häufig die gleichen Verfahren. Einige dieser Verfahren werden im Abschnitt 1.3.2 exemplarisch beschrieben.

1.3.1 Logische Struktur eines Audio-Streaming-Systems

Die in Abbildung 1-5 gezeigte logische Struktur stellt ein Audio-Streaming-System dar. Das System beschreibt die an der Übertragung des Streams beteiligten Funktionskomponenten vom Sender zum Empfänger (von links nach rechts).

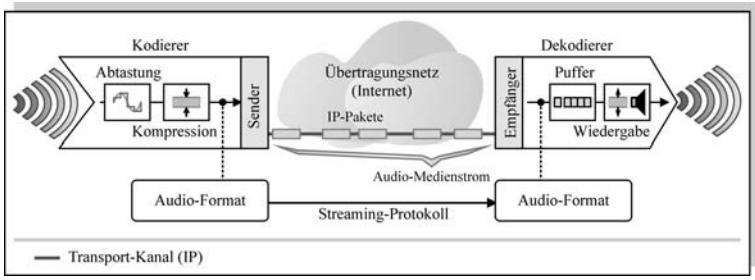

Abb. 1-5: Logische Struktur der Audio-Streaming-Funktionskomponenten

Der Kodierer auf der Seite des Senders (Media-Encoder) digitalisiert zunächst die Eingabedaten mittels *Abtastung*. Diese werden schließlich komprimiert (*Kompression*) und in ein *Audio-Format* überführt. Dieses Audio-Format stellt einen standardisierten Container für Audio-Daten dar, der es ermöglicht, dass auch Media-Player sowie -Encoder unterschiedlicher Hersteller den Stream verarbeiten können. Wichtige Formate in diesem Zusammenhang zeigt die Tabelle 1-2 im Abschnitt 1.3.3.

Vom Sender wird das Audio-Format schließlich in *IP-Pakete* verpackt und unter Verwendung eines Streaming-Protokolls über das *Übertragungsnetz* an den Empfänger (Media-Player) gesendet. Die IP-Pakete bilden dabei einen *Audio-Medienstrom* in Richtung des Empfängers. Der Empfänger reiht zunächst alle empfangenen Pakete in einen Empfangspuffer ein. Dabei erhält er aus den einzelnen Paketen das vom Sender verwendete Audio-Format zurück. Der Eingangspuffer des Empfängers ermöglicht es, Verzögerungen

beim Versand der Pakete über das Übertragungsnetz zu reduzieren (bzw. zu glätten). Die Wiedergabe erfolgt schließlich direkt aus dem Puffer des Media-Players.

Digitalisierte Audiodaten können in unterschiedlicher Qualität erstellt werden. Beispielsweise bestimmen die Abtastrate und die Abtasttiefe bereits den größten Teil des Umfangs der Daten.

Für eine 6 Sekunden lange Audio-Information in CD-Qualität entsteht bei einer Abtastrate von 44.1 kHz und 16 Bit in Stereo ein Datenvolumen von:

*44100 Hz * 16 Bit * 6 Sek * 2 (Stereo) = 8 467 200 Bit = 1 058 400 Byte*

Die Tabelle 1-1 zeigt die bei der Digitalisierung von Audio-Informationen entstehenden Datenmengen und die dazugehörigen subjektiven Qualitätsbeschreibungen. Dabei werden keine Kompressionsverfahren verwendet:

Audio-Qualität	Bandbreite	Kanäle	Bitrate
Telefon	2,5 KHz	1 (mono)	8 KBit/s
Kurzwellenradio	4,5 KHz	1 (mono)	16 KBit/s
Mittelwellenradio (AM)	7,5 KHz	1 (mono)	32 KBit/s
UKW-Radio (FM)	11 KHz	2 (stereo)	56-64 KBit/s
CD	>15 KHz	2 (stereo)	112-128 KBit/s

Tabelle 1-1: Subjektive Audio-Qualität - Bandbreite und Bitrate

1.3.2 Besonderheiten bei der Kodierung von Audio

Aufgrund der großen Menge an Rohdaten setzen nahezu alle Kodierungsverfahren für Audio bereits bei der Abtastung an. In dem etwa durch Pulse-Code-Modulation-(PCM)-Verfahren die Daten bereits bei ihrer Digitalisierung begrenzt werden. Eine besonders hohe Verbreitung hat dabei das Verfahren ADPCM (Adaptive Delta PCM). Beim DPCM (Delta PCM), das in Abbildung 1-7 gezeigt ist, wird für die nächst folgende Quantisierungsstufe jeweils eine Prädiktion erstellt. Die Abweichung der Vorhersage zum tatsächlichen Wert wird codiert und übertragen. Da ein in der Regel sinusförmiges Audiosignal leicht vorherzusagen ist, reduziert sich die Datenmenge sehr stark. Das A für „Adaptive“ steht beim ADPCM für variable Quantisierungsstufengrößen. In Bild 1-7 könnte man sich beispielsweise vorstellen,

dass die einzelnen Stufen auf der Y-Achse logarithmisch abgetragen wären, somit würde sich für kleinere Pegel und höhere Frequenzen eine sehr viel genauere Quantisierung ergeben, als bei einer gleichmäßigen Verteilung der Stufen. Dieses Vorgehen macht man sich etwa bei der CD zu Nutze, um die Qualität der mittleren und hohen Frequenzen zu steigern, ohne sie für tiefe Frequenzen unnötig zu verschlechtern.

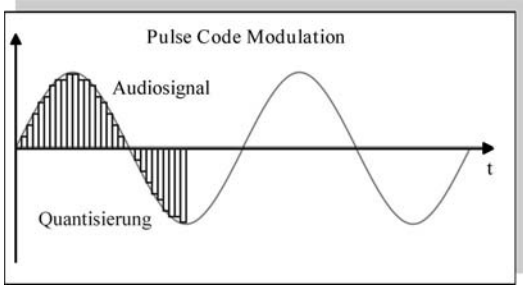

Abb. 1-6: Digitalisierung (Quantisierung) eines Audiosignals nach dem PCM-Verfahren

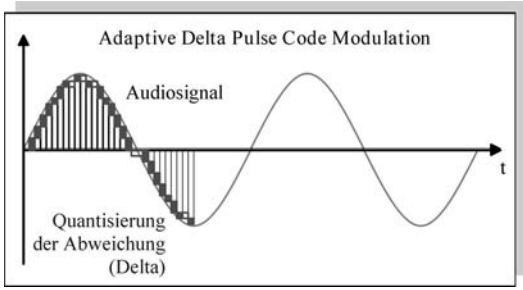

Abb. 1-7: Abtastung des Signals aus 1-6 nach dem DPCM-Verfahren

Weitere Informationen zum Thema Audio-Digitalisierung finden sich in [BBSZ_00].

Nach der Reduktion der Daten beim Abtasten setzen die meisten Kompressionsalgorithmen an der Tatsache an, dass das menschliche Ohr akustische Signale nur begrenzt auflösen kann. So lässt sich z. B. durch eine Hör-

schwellenmaskierung der benötigte Frequenzbereich bereits deutlich reduzieren. Das menschliche Ohr nimmt Frequenzen bzw. Töne zwischen 2 kHz und 4 kHz besonders gut auf. Frequenzen ober- und unterhalb dieser Grenze müssen sehr viel lauter sein, damit das Ohr sie wahrnimmt.

Hinzu kommt, dass das menschliche Ohr Frequenzen nur begrenzt gut filtern kann. So werden beispielsweise hohe Frequenzen leicht von tieferen überdeckt, was dazu führt, dass die hohen Frequenzen gar nicht mehr getrennt vom Ohr wahrgenommen werden. Daher werden solche überlagerte Frequenzbereiche bei vielen Kompressionsverfahren direkt aus dem Audio-Signal entfernt.

Außerdem lässt sich z. B. die Trennung von Stereo-Informationen aufbrechen. Ist etwa ein bestimmter Frequenzbereich, z. B. eine Gesangsstimme, sowohl rechts als auch links hörbar, so kann diese Information in Mono übertragen werden, was die benötigte Bandbreite für dieses Audio-Signal halbiert.

Grundsätzlich werden die Audio-Signale durch diese Maskierungen zusammen mit weiteren Kompressionsverfahren, die in der Regel ebenso wie die oben genannten Verfahren verlustbehaftet sind, so lange in ihrer Größe reduziert, bis die gewünschte Bandbreite erreicht wird. Eine Kombination dieser Verfahren stellt ein bestimmtes Kompressionsformat wie etwa das populäre *MP3* dar. Näheres zum Thema Audio-Kompression findet sich in [BBSZ_00].

1.3.3 Wichtige Audio-Formate

Durch die Festlegung von bestimmten Audio-Formaten wird sichergestellt, dass auch Media-Encoder und -Player verschiedener Hersteller gegenseitig zusammenarbeiten können. Die Tabelle 1-2 enthält einige wichtige Audio-Formate, die im Web verwendet werden, sowie deren Besonderheiten.

Format (Standard)	Abtast-rate	Bitrate	Kanäle	Besonderheiten
PCM	beliebig	beliebig	1	Rohdaten (verlust-frei)
ADPCM	beliebig	beliebig	1	Rohdaten (verlust-frei)

Tabelle 1-2: Gegenüberstellung wichtiger Audio-Formate (LFE = low frequency effects)

Format (Standard)	Abtast-rate	Bitrate	Kanäle	Besonderheiten
A-Law (G.711)	8 KHz	64 KBit/s	1	für Sprachkodierung, z. B. ISDN (Europa)
µ-Law (G.711)	8 KHz	64 KBit/s	1	Für Sprachkodierung, z. B. ISDN (USA, Japan)
G.726 (ADPCM)	8 KHz	16,24,32,40 KBit/s	1	A-Law und µ-Law
GSM 6.10	8 KHz	13 KBit/s	1	für Sprachkodierung, z. B. beim GSM-Mobilfunk

Tabelle 1-2: Gegenüberstellung wichtiger Audio-Formate (LFE = low frequency effects)

populäre Kompressionsformate	Kanäle	Besonderheiten
MP3 (MPEG Layer 3 Audio)	2	variable Bitrate
MP3pro	2	variable Bitrate
AAC (MPEG 2 Audio)	48 + LFE	DVD Audio
RealAudio	2 (5.1)	[REAL]
Windows Media Audio	7.1	variable Bitrate, 2-pass-encoding [WMA]
Ogg Vorbis	255	variable Bitrate [OGGV]

Tabelle 1-3: Gegenüberstellung wichtiger Audio-Formate (LFE = low frequency effects)

1.4 Video-Streaming im Internet

Beim Video-Streaming im Internet wird von nahezu allen Media-Encodern und Media-Playern ein einheitliches System eingehalten (siehe Abschnitt 1.4.1.). Neben der einheitlichen Struktur nutzen die Systeme in der Regel auch einheitliche Video-Formate. Video-Formate beschreiben nicht nur die Eigenschaften des transportierten Video-Signals, sondern auch dessen Kompression. Wichtige Video-Formate, die im Web verwendet werden, sind in der Tabelle 1-3 im Abschnitt 1.4.3 aufgeführt. Die für die Kompression der Signale verwendeten Verfahren sind bei vielen Video-Formaten gleich. Einige dieser Verfahren sind im Abschnitt 1.4.2 exemplarisch beschrieben.

1.4.1 Logische Struktur eines Video-Streaming-Systems

Die Abbildung 1-8 zeigt die logische Struktur eines Video-Streaming-Systems. Das System beschreibt die an der Übertragung des Streams beteiligten Funktionskomponenten vom Sender zum Empfänger (von links nach rechts).

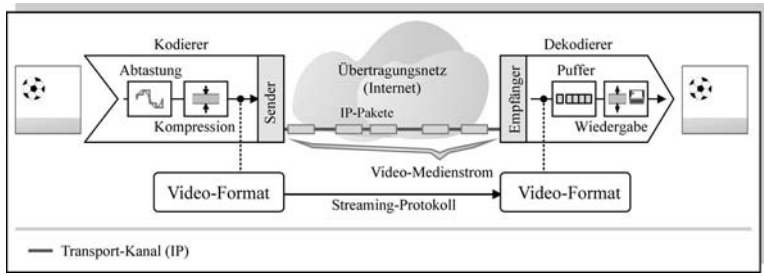

Abb. 1-8: Logische Struktur der Video-Streaming-Funktionskomponenten

Die Eingabedaten (in Form eines Bildes) werden zunächst vom Kodierer (Media-Encoder) digitalisiert (Abtastung). Danach werden sie komprimiert (*Kompression*) und in ein *Video-Format* eingebettet. Dieses Video-Format stellt einen standardisierten Container für Video-Daten dar, der es ermöglicht, dass auch Media-Player sowie -Encoder unterschiedlicher Hersteller den Stream verarbeiten können. Wichtige Formate in diesem Zusammenhang zeigt die Tabelle 1-4 im Abschnitt 1.4.3.

Vom Sender wird das Video-Format schließlich in *IP-Pakete* verpackt und unter Verwendung eines Streaming-Protokolls über das *Übertragungsnetz* an den Empfänger (Media-Player) gesendet. Dadurch entsteht ein *Video-Medienstrom* in Form von mehreren IP-Paketen in Richtung des

Empfängers. Die empfangenen Pakete werden vom Empfänger zunächst in einen *Empfangspuffer* eingereiht. Dabei gewinnt er aus den einzelnen Paketen das vom Sender verwendete Video-Format zurück. Der Eingangspuffer des Empfängers ermöglicht es, Verzögerungen beim Versand der Pakete über das Übertragungsnetz zu reduzieren. Die *Wiedergabe* erfolgt schließlich direkt aus dem Puffer des Media-Players.

Die Digitalisierung von Video-Informationen führt je nach Qualität zu sehr großen Datenmengen. Dabei bestimmen bereits vor der Kompression der Daten die *Auflösung*, die *Farbtiefe* und die *Bildrate* (Bilder pro Sekunde) die Größe der entstehenden Datenmenge. Geringere Auflösungen, Farbtiefen und Bildraten reduzieren die entstehenden Rohdaten um ein Vielfaches.

Als Beispiel soll eine 10 Sekunden lange Video-Sequenz in PAL-Auflösung (Fernseh-Qualität) übertragen werden. Die Farbinformation soll dabei mit 24 Bit kodiert werden (ca. 16 Millionen mögliche Farben). Um eine flüssige Wiedergabe des Films zu erreichen, soll die Bildrate 25 Bilder pro Sekunde betragen (beim Fernsehen, werden 50 Halbbilder pro Sekunde übertragen, was effektiv ebenfalls 25 Bildern entspricht). Es ergibt sich ein Datenvolumen von:

*768 * 576 Pixel * 24 Bit (Farbe) * 25 Bilder/s * 10 Sekunden = 2 654 208 KBit = 331 776 KByte = ca. 332 MByte!*

Für die Übertragung der oben genannten Videoinformation würden somit ca. 33 MByte/s benötigt. Weitere Informationen zum Thema Video-Digitalisierung finden sich in [BBSZ_00].

1.4.2 Besonderheiten bei der Kodierung von Video

Bei der Kodierung der Farbinformation macht man sich häufig den Effekt zu Nutze, dass das menschliche Auge, bedingt durch die Verteilung von Zäpfchen und Stäbchen auf der Netzhaut, Helligkeit sehr viel besser wahrnimmt als Farbinformationen. Die 4:2:2-Abtastung bedient sich dabei des YUV-Modells. Wobei Y die Helligkeit und U sowie V die Farbinformation kodieren. Das Verhältnis zwischen Y, U und V wird mit 4 zu 2 zu 2 definiert. Damit wird die Farbinformation im Vergleich zur Helligkeit halbiert und damit die Datenmenge reduziert.

Weitere Kompressionsverfahren gliedern sich in:

- Entropiekodierung (Redundanzreduktion) - verlustfrei
- Statistische Kodierung - verlustfrei
- Quellenkodierung (Irrelevanzreduktion) - verlustbehaftet

Entropie- und Statistische Kodierung (Redundanzreduktion)

Die Entropiekodierung bezieht sich auf den digitalisierten Datenstrom. In ihm existieren sehr viele Redundanzen, die ohne Verlust reduziert werden können.

Als erstes lassen sich zeitliche Redundanzen in der Videoinformation verringern. Häufig ändert sich von einem Bild zum nächsten nur ein sehr kleiner Teilbereich des Gesamtbilds. So bleibt etwa bei einer Nachrichtensendung im Fernsehen der Hintergrund weitgehend gleich, während sich die Lippen des Sprechers bewegen, oder einzelne Textzeilen eingeblendet werden. In diesem Beispiel würde der Hintergrund in den fortlaufenden Bildern redundant übertragen. Eine zeitliche Differenzkodierung ermöglicht es, unveränderte Bildausschnitte zu erkennen, und diese im Folgebild nicht erneut zu kodieren. Abbildung 1-9 zeigt einen Fußball, der über einer grünen Fläche fliegt. In diesem Beispiel müsste auf dem zweiten Bild nur der Fußball an seiner neuen Position übertragen sowie an seiner alten Position entfernt werden.

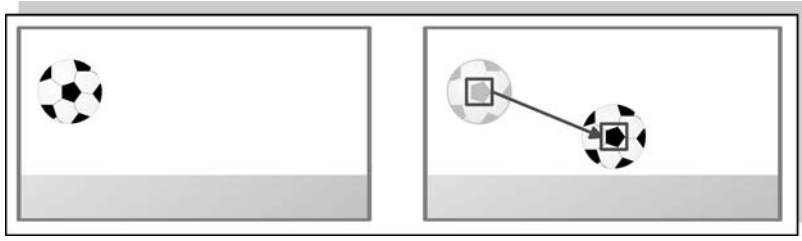

Abb. 1-9: Zeitliche Differenzkodierung - Beispiel eines fliegenden Fußballs

Nicht nur im Bild selbst stecken Redundanzen, sondern auch in dem Bitstrom, der aus seiner Kodierung entsteht. Diese Redundanzen können ebenfalls reduziert werden.

Im folgenden Beispiel soll eine Lauflängenkodierung RLE (run length encoding) durchgeführt werden. Dabei werden alle Sequenzen separiert, in denen gleiche Werte aufeinander folgen (Redundanzen).

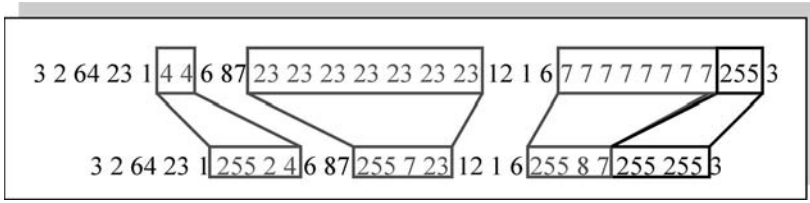

Abb. 1-10: RLE-(run length encoding)-Lauflängenkodierung – verlustfrei

Wie in Abbildung 1-10 gezeigt, werden Sequenzen mit gleichen aufeinander folgenden Werten zu Blöcken zusammengefasst. Beispielsweise wird der Block „4 4“ kodiert als „255 2 4“. Dabei steht die 255 als Sonderzeichen (Header), die eine wiederholte Folge gleicher Werte einleitet. Die 2 steht für die Länge der Wiederholung und die 4 schließlich für den Wert, der wiederholt wird.

Während dieses Verfahren bei diesem Block noch zu einer Verlängerung des codierten Ausgabestroms führt, wird z. B. der Block mit dem Wert „23“ sehr stark verkürzt. Eine besondere Rolle spielt in diesem Beispiel der Wert 255 im Eingabestrom. Dieser ist im Ausgabestrom als Sonderzeichen für die Einleitung einer wiederholten Wertefolge definiert. Daher muss er im Ausgabestrom gesondert kodiert werden. Dies wird realisiert, indem das Sonderzeichen doppelt im Ausgabestrom codiert wird.

An die in Abbildung 1-10 gezeigte RLE-Kodierung schließt sich häufig eine statistische Kodierung an: die Huffman-Kodierung. Dabei werden Blöcke nach ihrer Häufigkeit im Eingabestrom mit unterschiedlich langen Bitfolgen kodiert. Der Block, der am häufigsten im Eingabestrom enthalten ist, erhält die kürzeste Bitfolge, was die Effizienz der Übertragung erneut steigert, ohne Verluste zu erzeugen. Abbildung 1-11 zeigt die Abbildung einer Folge von RLE-Blöcken nach ihrer Häufigkeit auf eine Baumstruktur (Huffman-Kodierung).

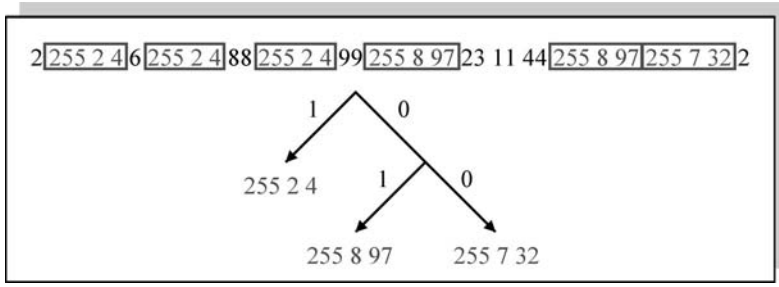

Abb. 1-11: Huffman-Kodierung – einzelnen Blöcke werden in einer Baumstruktur verzeichnet

Der Block „255 2 4“ kommt im Eingabestrom am häufigsten (insgesamt dreimal) vor. Da insgesamt sechs RLE-Blöcke im Beispiel unterschieden werden, ergibt sich eine Häufigkeit von 50 %. Somit bekommt er die kürzeste Kodierung und das erste Blatt des Huffman-Baums. Für die anderen Blöcke ergeben sich die in der Tabelle 1-3 zusammengetragenen Werte.

RLE-Block	Häufigkeit	Binäre Huffman-Kodierung
255 2 4	50 %	1
255 8 97	33 %	01
255 7 32	17 %	00

Tabelle 1-4: Huffmann-Kodierung der RLE-Blöcke

Die Entropiekodierung sorgt zusammen mit der statistischen Kodierung für eine effiziente Reduktion der Datenmenge, ohne dabei Verluste zu erzeugen.

Quellenkodierung (Irrelevanzreduktion)

Die meisten Kompressionsverfahren und vor allem sämtliche MPEG (Motion Picture Experts Group)-Standards setzen auf die Reduktion von irrelevanten Informationen. Das menschliche Auge besitzt neben der in der 4:2:2 genutzten Schwäche bei der Farbwahrnehmung eine Schwäche bei der Wahrnehmung von Details. Während es etwa 25 Bilder pro Sekunde erwartet, um eine Sequenz als „flüssig“ einzustufen, nimmt es längst nicht alle Details dieser 25 Bilder auf. Die in diesem Rahmen irrelevanten Informationen lassen sich durch zwei Verfahren reduzieren.

Das erste Verfahren stellt ein Prädiktionsverfahren dar, das auch beim MPEG-Standard zum Einsatz kommt. Grundsätzlich könnte man sagen, dass das Auge allein zeitlich nicht in der Lage ist, 25 Bilder pro Sekunde aufzunehmen. Entfernt man jedoch auch nur wenige Bilder wirkt dies auf den Betrachter schnell als „Ruckeln“ in der gesamten Sequenz. Daher entfernt man beim Prädiktionsverfahren die Bilder nicht komplett sondern versucht, aus ihren vorherigen und folgenden Bildern ein Mischbild zu erstellen. Dabei stellen Bilder, die ihren vollen Informationsumfang enthalten und keiner Prädiktion unterliegen, so genannte I-Frames (Intra-Frames) dar. Ein Bild, das als Prädiktion aus beiden I-Frames entsteht, nennt man P-Frame (Predicted-Frame). In der Abbildung 1-12 lässt sich erkennen, dass eine Vorhersage (Prädiktion) häufig sehr leicht möglich ist. Die Position des Balls zwischen den beiden I-Frames ist leicht zu ermitteln.

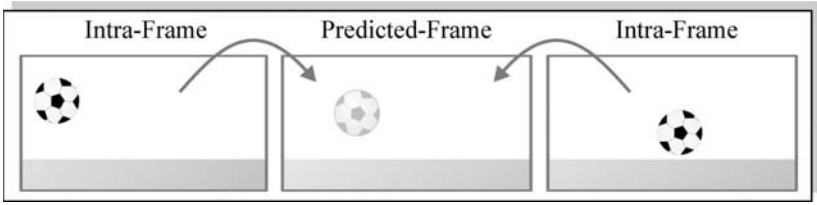

Abb. 1-12: Prädiktion: Flugbahn des Balls – zwischen zwei I-Frames

Beim MPEG-Standard werden zwischen I-Frame und P-Frame zusätzlich zwei B-Frames (Bidirectional Predicted Frames) eingefügt. Diese B-Frames ergeben sich aus der Prädiktion zum einen vom P-Frame in Richtung des I-Frames und zum anderen in umgekehrter Richtung. Durch diese Verteilung von I-, P- und B-Frames ergibt sich beim MPEG-Standard ein monotones Muster von I,B,B,P,B,B,I,B,B,P,B,B,…, wie in Abbildung 1-13 gezeigt wird. Für den Betrachter entsteht ein flüssiger Film, der in Wirklichkeit sehr viele Details, insbesondere auf Flächen, die sich wenig verändern oder bewegen, reduziert. Beim MPEG-Verfahren wird somit auch die Zeitliche Differenzkodierung erreicht.

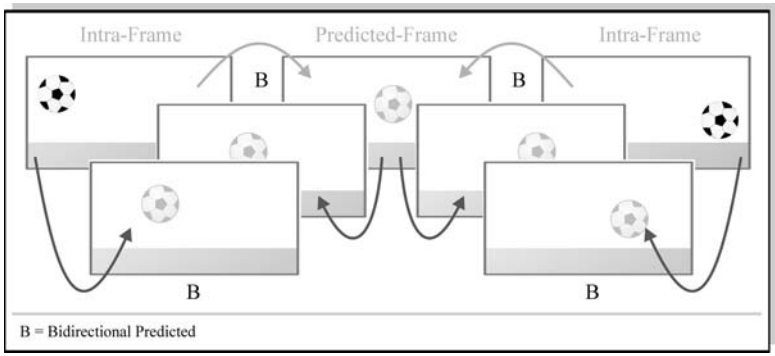

Abb. 1-13: I,B,B,P,B,B,I… Frame-Reihenfolge

Auch die Bilder selbst beinhalten Irrelevanzen. Diese werden sowohl bei der JPEG- als auch bei der MPEG-Kodierung vorrangig durch eine diskrete Cosinus Transformation (DCT) reduziert. Dabei wird das Bild zunächst in Blöcke mit jeweils 8 x 8 Pixeln eingeteilt. Diese Blöcke werden dann durch die DCT von ihrer örtlichen Information (Pixel) in eine Frequenzdarstellung überführt. Theoretisch wäre dieser Vorgang verlustfrei, in der Realität jedoch bewirken Rundungsfehler, dass insbesondere hohe Frequenzen im Bild (scharfe Kanten) ungenau codiert werden. Daraus resultieren die für JPEG und MPEG typischen „Schlieren“ im Bild.

Abbildung 1-14 zeigt die grundsätzliche Verfahrensweise beim Kodieren der einzelnen Blöcke. Zunächst werden die Farben im Bild vom RGB- in das YUV-Modell übertragen, was zu einer 4:2:2-Kompression führt, die bereits am Anfang des Abschnitts 1.4.2 vorgestellt wurde. Danach werden einzelne Blöcke von jeweils 8 x 8 Pixeln aus dem Bild geschnitten. Diese Blöcke werden von Pixeln nach Frequenzen gewandelt und schließlich in einer Quantisierungsmatrix, die genau diese Frequenzen enthält, gespeichert. Beim Auslesen wird die Matrix im unten gezeigten Schema von links oben nach rechts unten durchlaufen. Der Vorteil dieses Zickzack-Verfahrens ist, dass sich sehr viele redundante Bereiche ergeben, die die gleichen Werte im Bitstrom besitzen. Genau dieser Effekt bewirkt in der anschließenden RLE, dass sehr große Blöcke als kurze RLE-Sequenz kodiert werden können. Die anschließende Huffman-Codierung bewirkt eine zusätzliche Kompressionssteigerung. Dabei werden in diesem Beispiel die extrem langen Nullfolgen mit dem kürzesten Wert des Huffman-Baums kodiert. Das dargestellte Verfahren beschreibt die Grundzüge der klassischen *MPEG-Kompression*. Näheres dazu findet man z. B. in [BBSZ_00].

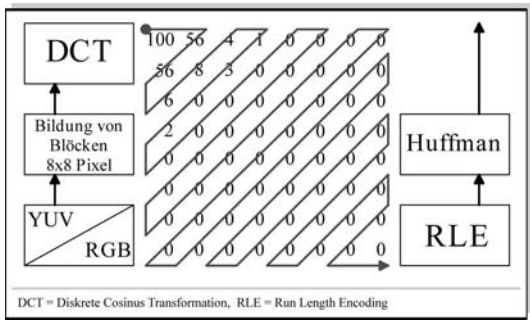

Abb. 1-14: Die DCT-Kodierung bei JPEG und MPEG

1.4.3 Wichtige Video-Formate

Durch die Festlegung von bestimmten Video-Formaten wird sichergestellt, dass auch Media-Encoder und -Player verschiedener Hersteller gegenseitig zusammenarbeiten können. Die Tabelle 1-4 enthält einige wichtige Video-Formate, die im Web verwendet werden, sowie deren Besonderheiten.

Format	Auflösung	Bitrate	Besonderheiten
PAL-DV	720 x 576	249 MBit/s	Europäisches TV (768 x 576)
CIV	360 x 288	62 MBit/s	Videokonferenzstandard
QCIV	180 x 144	16 MBit/s	

Populäre Kompressionsformate	Kodierung	Besonderheiten
MPEG	DCT	Videokonferenzen, variable Bitrate, 2-pass-encoding
MPEG-2	DCT	DVD-Video, variable Bitrate, 2-pass-encoding
RealVideo	DWT	variable Bitrate, [REAL]

Tabelle 1-5: Gegenüberstellung wichtiger Video-Formate

Populäre Kompressionsformate	Kodierung	Besonderheiten
Windows Media Video	DCT	Variable Bitrate und 2-pass-encoding (ab Version 8), [WMV]
Sorenson	VQ	Variable Bitrate, 2-pass-encoding, [QUIC]
DivX	DCT	

Tabelle 1-5: Gegenüberstellung wichtiger Video-Formate

1.5 Streaming mittels HTTP

Beim HTTP (Hypertext Transfer Protocol) [HTTP] werden die Web-Seiten als Stream zum Benutzer übertragen. Daher kann das universelle HTTP auch als Streaming-Protokoll eingesetzt werden, wobei man von Streaming mittels HTTP spricht. In diesem Fall werden die Streaming-Inhalte (Audio,Video) direkt per HTTP vom Web-Server versendet. Die Streaming-Inhalte liegen somit als Dateien auf dem Web-Server vor, was nur einen Konserven-Stream zulässt. Die Nutzung von HTTP als Streaming-Protokoll birgt einige Nachteile in sich:

- Da das HTTP auf TCP basiert, unterliegt es den typischen Nachteilen dieses Protokolls: große Verzögerungszeiten, anfänglich langsame Übertragungen (sog. „slow-start“-Problem), siehe [BADA_01].
- Obwohl die Daten als Stream an den Browser verschickt werden, stellt dieser sie erst nach dem kompletten Empfang dar. Die Daten werden zunächst heruntergeladen und dann an den Media-Player übergeben.
- HTTP definiert keine Kontrollmöglichkeiten, wie z. B. Pause oder Vor- bzw. Zurückspulen im Stream.

Während der erste Nachteil unzertrennlich mit dem HTTP-Standard verbunden ist, kann der Nachteil des kompletten Empfangs der Inhalte vor der Wiedergabe durch eine Beschreibungsdatei (auch Meta-Datei genannt) behoben werden. Dabei empfängt der Browser in diesem Fall per HTTP lediglich eine Datei, die die Web-Adresse, den sog. URL (Uniform Resource Locator [RFC_1738]), und die Eigenschaften des angewählten Streams enthält. Diese Datei ist schnell übertragen und kann direkt an den Media-Player weitergegeben werden, der den URL umgehend öffnen und wiedergeben kann.

Der Nachteil der fehlenden Kontrollmöglichkeiten des Streaming bleibt natürlich ebenso bestehen wie der Nachteil der Nutzung des TCP. Diese Probleme werden gewissermaßen lediglich weitergereicht, da nun der Media-Player per HTTP über den URL den Stream öffnet und wiedergibt.

Die übrig gebliebenen Nachteile lassen sich nur durch die Definition eines gesonderten Protokolls beheben. Es stellt sich die Frage: Warum wird HTTP überhaupt als Streaming-Protokoll verwendet? Zum einen bietet sich der Vorteil der einheitlichen Verwaltung. Sollen z. B. Videos oder Audio-Inhalte generell erst heruntergeladen werden, so lassen sie sich direkt mit den HTML-Seiten auf dem Web-Server speichern. In diesem Fall liegt die Webseite zentral auf einem Server, was die Verwaltung vereinfacht. Der eigentliche Vorteil liegt jedoch darin, dass auch Benutzer hinter einer sehr restriktiven Firewall (z. B. im Unternehmen) in der Regel ohne Probleme HTTP-Nachrichten senden und empfangen dürfen. Andere Streaming-Protokolle benutzen gesonderte Ports für Ihre Verbindungen. Diese sind evtl. nicht in der Firewall des Unternehmens freigegeben. HTTP bietet somit den Vorteil, dass es theoretisch von jedem am Internet angeschlossenen Rechner ohne Einschränkungen nutzbar ist. Für die Übertragung von Streaming-Inhalten mittels HTTP werden gesonderte Pragma-Eigenschaften im HTTP-Header verwendet, um die Daten annähernd in Echtzeit transferieren zu können.

Abbildung 1-15 zeigt den Ablauf einer Streaming-Media-Sitzung unter Verwendung des HTTP als Streaming-Protokoll. Dabei wird vom Web-Browser eine Meta-Datei an den Media-Player übergeben. Dieser ruft schließlich den in der Meta-Datei enthaltenen URL ab und bekommt als HTTP-Response den gewünschten Stream.

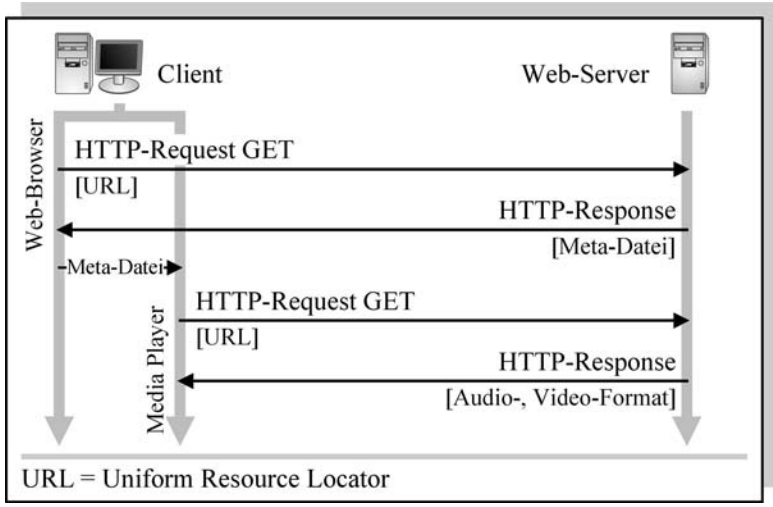

Abb. 1-15: Streaming mittels HTTP und Meta-Datei

1.6 RTSP – Real Time Streaming Protocol

Das RTSP (Real Time Streaming Protocol) wurde im April 1998 veröffentlicht [RFC_2326] und unterstrich damit den Boom von Streaming-Media im Internet. Ziel war es, ein übergreifendes Protokoll zu entwickeln, das frei und herstellerunabhängig verwendet werden konnte. Anders als beim Streaming mittels HTTP aus Abschnitt 1.5 verwendet das RTSP separate Protokolle zur Übertragung der Streaming-Inhalte (sog. *Transportprotokolle*). Somit können Protokolle verwendet werden, die durch den Einsatz von UDP die Schwächen der TCP-Verbindung, wie sie bereits in Abschnitt 1.5 erwähnt wurden, unterbinden.

RTSP selbst definiert explizit kein solches Transportprotokoll. Es dient vielmehr als Steuerungs- und Kontrollprotokoll, das zur eigentlichen Übertragung der Daten andere Protokolle verwendet. Daher spricht man beim RTSP auch von einem „out-of-band"-Protokoll, da die Daten nicht im Übertragungsband von RTSP transferiert werden (im Gegensatz zum HTTP als „in-band"-Protokoll, das die Nutzdaten in HTTP-Responses verpackt).

Durch den „out-of-band"-Aufbau des RTSP wird die Unterstützung von mehreren parallelen Streams möglich, die verwaltet (z. B. synchronisiert) werden können. Um die Streaming-Inhalte anzufordern und zu senden, wird neben dem Web-Server ein sog. Streaming-Server benötigt. Abbildung 1-16

zeigt die Integration eines Streaming-Servers in den Kommunikationsablauf. Streaming-Server und Web-Server können auf demselben Rechner als Funktionskomponenten eingesetzt werden oder, wie hier gezeigt, auf separaten Rechnern installiert werden.

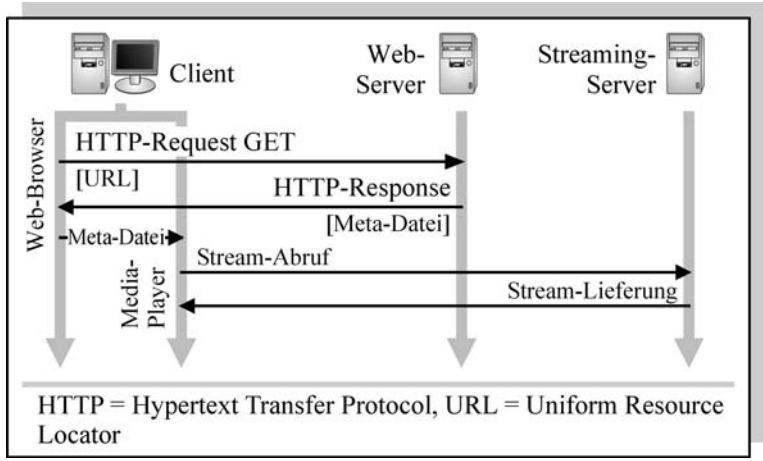

Abb. 1-16: Streaming über einen Streaming-Server

Das Protokoll RTSP weist eine dem HTTP sehr ähnliche Struktur auf. Auch beim RTSP werden die Nachrichten als plain text (Klartext) mit einer einfachen Kommandostruktur verschickt. Auch die Statuscodes sind denen des HTTP angepasst. Anders als beim HTTP kann beim RTSP neben TCP auch das Protokoll UDP für die Übertragung der Nachrichten benutzt werden. Als „well-known"-Port wurde dem RTSP serverseitig sowohl für UDP als auch für TCP der Port 554 zugewiesen. Wie HTTP funktioniert auch das RTSP nach dem Client/Server-Prinzip, wobei die von ihm übermittelten Nachrichten in Requests und Responses unterschieden werden können. Beim RTSP kann, im Gegensatz zum HTTP, sowohl der Server als auch der Client Requests verschicken bzw. mit Responses darauf antworten.

1.6.1 Phasen einer RTSP-Sitzung

Innerhalb einer RTSP-Sitzung kann zwischen verschiedenen Phasen unterschieden werden. Abbildung 1-17 zeigt den Ablauf der einzelnen Phasen beim RTSP. Folgende Phasen werden dabei unterschieden:

- Web-Verbindung

  Diese erste Phase beginnt bereits vor der RTSP-Sitzung. In ihr wird eine

Verbindung zum Web-Server hergestellt. Der Web-Browser erhält dabei eine Meta-Datei, die die Adresse des gewünschten Streams enthält.

- Aufbau der Kontrollverbindung

  Das RTSP stellt ein Kontrollprotokoll für Streaming dar. Daher wird in der folgenden Phase seitens des Clients eine RTSP-Verbindung als Kontrollverbindung zum Server aufgebaut. Diese Verbindung bleibt bis zum Ende der Sitzung bestehen, wobei der Stream „out-of-band“ über ein Transportprotokoll zum Client gesendet wird. Beim Aufbau der Kontrollverbindung erhält der Client zusätzlich alle nötigen Informationen, wie z. B. verschiedene Tonspuren, vom Server.

- Medienauswahl

  Anhand der Informationen über den Stream kann der Client schließlich den gewünschten Stream auswählen. So kann er z. B. für einen Stream die Tonspur in seiner Landessprache auswählen oder einen Stream in einer niedrigen Auflösung anfordern, um sich an die aus seiner Sicht größtmögliche Übertragungsrate anzupassen.

- Aufbau der Transportverbindung

  Im Anschluss an die Auswahl des gewünschten Streams kann in dieser Phase die Transportverbindung aufgebaut werden.

- Medienkontrolle

  Während der Übertragung kann der Betrachter auf unterschiedliche Art und Weise Einfluss auf die Wiedergabe des Streams nehmen. So kann er den Stream z. B. kurzzeitig unterbrechen (PAUSE) oder vor- bzw. zurückspulen.

- Transportüberwachung

  Der Transport zum Client wird vom Server ständig überwacht. Dabei werden statistische Daten über den übertragenen Stream ausgewertet. Kommt z. B. das Transportprotokoll RTP (Real Time Transport Protocol) zum Einsatz, so kann über das Protokoll RTCP (Real Time Transport Control Protocol) eine Auswertung z. B. der empfangenen Daten beim Client erfolgen sowie der zeitliche Versatz (Verzögerung und Jitter) ausgewertet werden. Dadurch kann ggf. auf einen Stream mit einer niedrigeren Übertragungsrate zurückgeschaltet werden, um automatisch eine verbesserte, z. B. flüssigere, Wiedergabe beim Client zu erreichen.

- Abbau der Transportverbindung

  Am Ende des Streams wird in einer neuen Phase die Transportverbindung geschlossen. Dies kann auch durch den Benutzer initiiert werden, in dem er den Stream beendet. Auch ein kurzzeitiges Unterbrechen (PAUSE) des Streams führt zum Ab- und späteren Wiederaufbau der Transportverbindung.

- Abbau der Kontrollverbindung

  Nachdem die Transportverbindung geschlossen und Anwender oder Server den Stream beendet haben, wird schließlich auch die Kontrollverbindung geschlossen. In dieser Phase wird damit auch die RTSP-Sitzung beendet.

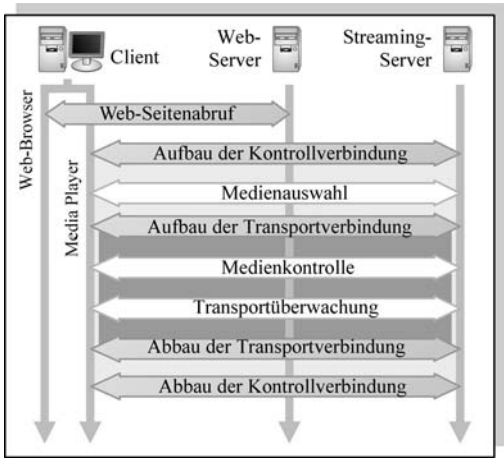

Abb. 1-17: Phasen einer RTSP-Sitzung

1.6.2 Methoden des RTSP

Der Kernbestandteil eines RTSP-Request ist ein Kommando, das auch als Methode bezeichnet wird. Beim RTSP werden vorrangig folgende Methoden genutzt:

- SETUP: Regelt die Initialisierung einer Verbindung. Vorrangig gibt der Parameter Transport in einer SETUP-Nachricht an, wie der Client zu erreichen ist und welches Protokoll für den Transport verwendet werden soll. Die SETUP-Methode darf auch während einer bestehenden Verbindung erneut ausgeführt werden, etwa um den Transport an neue Gegebenheiten im Netz anzupassen.

Grundsätzlich gibt der Client im Request SETUP an, welchen Transport des Streams er akzeptieren kann, und der Server bestätigt ihm die schließlich verwendete Transport-Methode in der Response SETUP.

- PLAY: Die PLAY-Methode sendet nach dem erfolgreichen SETUP einen Stream an den Client. Dabei kann über den Parameter Range ein bestimmter Zeitraum für die Wiedergabe ausgewählt werden. Dieser Zeitraum kann per SMPTE-Time-Code gestellt werden, so dass eine sehr exakte zeitliche Synchronisierung möglich wird. Näheres zu SMPTE siehe [SMPTE].
- RECORD: Diese optionale Methode ermöglicht es dem Client, einen Stream auf den Server zu laden. Dabei kann der Stream auch live eingespeist werden. Diese Methode spielt vor allem bei Videokonferenzen (die mit dem Parameter Conference angegeben werden können) eine Rolle.
- PAUSE: Die Methode PAUSE ist optional und bietet die Möglichkeit, einen empfangenen Stream kurzzeitig zu unterbrechen.
- TEARDOWN: Mit TEARDOWN initiiert der Client den Abbau einer RTSP-Verbindung beim Server.

Neben diesen Methoden, die die Grundfunktionalität gewährleisten, existieren beim RTSP folgende weitere Methoden:

- OPTIONS: Die Response auf einen OPTIONS-Request enthält analog zu ihrem Pendant beim HTTP die vom Server unterstützten Methoden. So kann ein Client mittels OPTIONS erfahren, welche Methoden er an den Server senden kann.
- DESCRIBE: Beschreibt einen auf dem Server abrufbaren Stream. Die Response auf einen DESCRIBE-Request enthält beispielsweise die Dauer (Länge) eines Streams, eventuelle Kommentare, Adresse des Autors usw.
- ANNOUNCE: Der ANNOUNCE-Request stellt das Gegenstück zum Request DESCRIBE dar und ermöglicht Autoren eines Streams, Beschreibungen oder Titel anzufügen. Diese Funktion hängt insbesondere mit der Methode RECORD zusammen.
- GET_PARAMETER: Mit dem Request GET_PARAMETER lassen sich einzelne Parameter bzw. Variablen des Servers abfragen. Die Implementierung der einzelnen Variablen kann bei unterschiedlichen RTSP-Servern variieren. Häufig lassen sich jedoch Informationen, wie z. B. Jitter oder Anzahl der gesendeten Pakete usw., vom Server abfra-

gen. Anhand dieser Informationen kann der Client schließlich evtl. die Empfangsparameter des Streams mittels SETUP anpassen.

- SET_PARAMETER: Setzt als Gegenstück zu GET_PARAMETER Variablen auf dem Server. Die Implementierung ist offen, es sind keine Variablen fest vorgegeben.
- REDIRECT: Über die Methode REDIRECT kann ein Server dem Client mitteilen, dass er den von ihm gewünschten Stream von einem anderen RTSP-Server abrufen soll. Dadurch wird die Anfrage des Clients nach einem Stream umgeleitet.

Zur eindeutigen Identifizierung einer RTSP-Sitzung wird in allen Methoden außer SETUP und OPTIONS der Parameter Session angegeben, der eine Sitzung auf dem Server eindeutig identifiziert. Dadurch können Requests von verschiedenen aktiven Clients unterschieden werden. Außerdem enthalten alle Requests den Parameter CSeq, der eine fortlaufende, bei jedem neuen Request inkrementierte, Zahl enthält. Anhand der CSeq (Command Sequence) kann die chronologisch korrekte Abfolge von Requests und Responses gesichert werden.

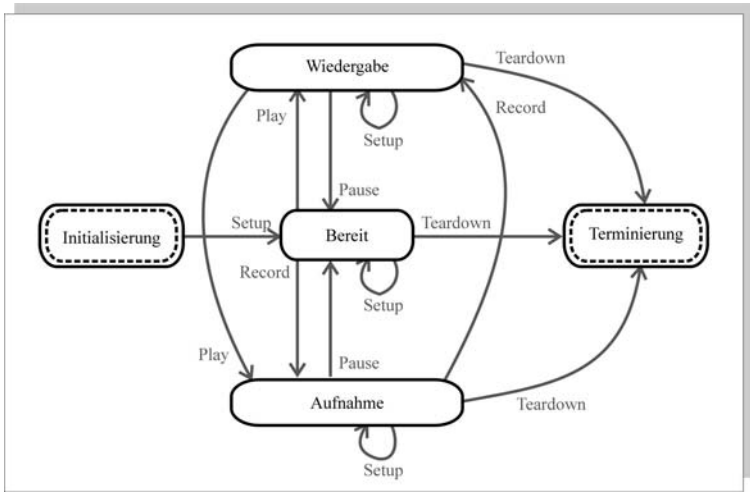

Abb. 1-18: Reihenfolge der Methoden und Zustände des RTSP

Das in Abbildung 1-18 gezeigte Zustandsdiagramm zeigt die beim RTSP möglichen Zustände sowie deren zugehörige Methoden. Die Methode SETUP nimmt dabei eine Sonderrolle ein, da sie bei den Zuständen Wiedergabe, Bereit und Aufnahme ausgeführt werden kann, ohne dass sich dabei

der Zustand des Protokollablaufs ändert. Wie bereits bei der Methode SETUP eingangs in diesem Abschnitt beschrieben, kann durch diese Methode z. B. der Stream während der Wiedergabe ohne Unterbrechung angepasst werden. Dadurch kann z. B. auf einen Bandbreitenengpass reagiert werden, indem die Auflösung des übertragenen Videos reduziert wird, um die flüssige Wiedergabe aufrechtzuerhalten. Beim RTSP können folgende Zustände unterschieden werden:

- Initialisierung

  In diesem Zustand befindet sich ein RTSP-Client bzw. -Server vor dem Aufbau einer RTSP-Sitzung. Z. B. ein Media-Player direkt nach seinem Start und vor der Eingabe der gewünschten URL mit dem Streaming-Media-Inhalt.

- Bereit

  Dieser Zustand wird nach dem Aufbau der Verbindung zwischen Client und Server erreicht. Dieser Zustand ist lediglich ein Übergang. In ihm wird z. B. der gewünschte Stream etwa die gewünschte Landessprache oder Qualität ausgewählt.

- Wiedergabe

  In diesem Zustand wird der vom Server bereitgestellte Stream auf dem Client wiedergegeben. Dieser Zustand ist der häufigste eines Media-Players.

- Aufnahme

  Im Aufnahme-Zustand speist der Client einen Stream auf den Server. Dies ist zum einen für Live-Streaming von Bedeutung, wird aber auch z. B. bei Video-Konferenzen benötigt.

- Terminierung

  In diesem Zustand wird die RTSP-Sitzung geschlossen.

1.6.3 Aufbau von RTSP-Nachrichten

Abbildung 1-19 zeigt die Struktur von RTSP-Requests. Die entscheidende Information des Requests steht in der ersten Zeile, der sog. Request-Line. Das Feld *Methode* enthält dabei den Namen einer der in Abschnitt 1.6.2 erwähnten RTSP-Methoden. *RTSP-Version* enthält Angaben über die Version des verwendeten RTSP-Protokolls. Bis zum Erscheinen einer neueren RTSP-Version steht hier RTSP/1.0. Der *Request-URI* enthält die absolute Adresse des Streams bzw. der Streamdefinition, auf die sich der Request

bezieht. Anders als beim HTTP muss beim RTSP im Request-URI zwingend das zu verwendende Protokoll angegeben werden (in der Regel mit rtsp://)

Im *Request-Header* stehen Angaben über den verwendeten Media-Player (User-Agent), die akzeptierte Kodierung und Sprache (Accept-Encoding, Accept-Language) sowie mögliche Autorisierungen (Benutzername, Passwort). Außerdem wird hier die für die Methode PLAY notwendige Range (der Bereich des Streams) angegeben. Eine ausführliche Beschreibung des RTSP-Request-Headers findet sich in [RFC_2326].

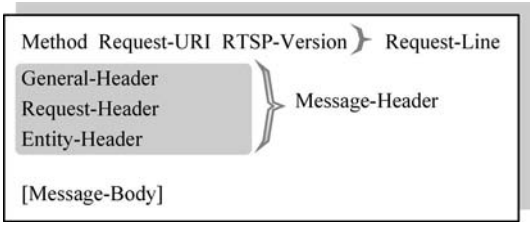

Abb. 1-19: Struktur der Requests beim RTSP

Die Struktur der Responses beim RTSP zeigt die Abbildung 1-20. Analog zu den Requests steht auch hier die wichtigste Information in der ersten Zeile, der sog. Status-Line. Die *RTSP-Version* ist bis dato RTSP/1.0. Der *Status-Code* wird ähnlich wie beim HTTP durch einen dreistelligen Zahlencode repräsentiert, deren erste Stelle die Zugehörigkeit zu einer Klasse definiert. Die Klassen sind beim RTSP analog zum HTTP definiert, also z. B. 2xx für positive Server-Rückmeldungen, 5xx für Serverfehler usw. Der *Reason-Phrase* gibt schließlich eine kurze Fehler- bzw. Statusbeschreibung in Klartext. Häufige Statusmeldungen beim RTSP sind: 200 OK, 250 Low on Storage Space (insbesondere beim RECORD) und 404 Not Found (für die Auswahl eines nicht existenten URI). Eine genauere Beschreibung der Status-Codes bietet [RFC_2326].

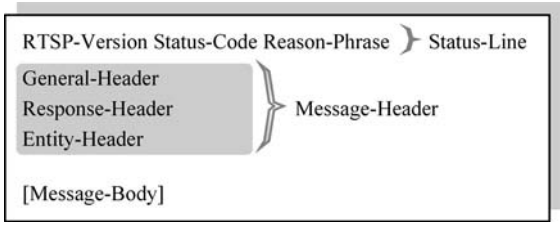

Abb. 1-20: Struktur der Responses beim RTSP

Der *General-* sowie der *Entity-Header* ist bei Requests und Responses gleich. Im General-Header lassen sich z. B. das Datum der Message angeben sowie Reglementierungen für ein Caching treffen.

Für eine genauere Beschreibung der möglichen Parameter in den RTSP-Headern sei erneut auf [RFC_2326] verwiesen.

1.6.4 Typischer Verlauf einer RTSP-Sitzung

Einen typischen Verlauf einer RTSP-Sitzung zeigt die Abbildung 1-21.

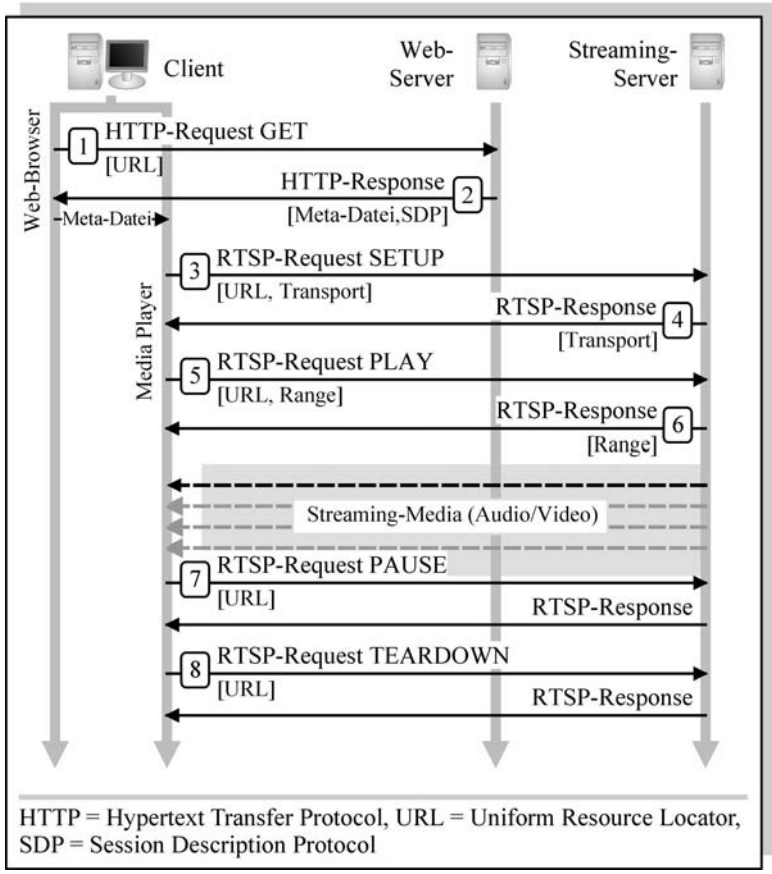

Abb. 1-21: Typischer Verlauf einer RTSP-Sitzung

Bei dem in Abbildung 1-21 gezeigten Ablauf sind die folgenden Schritte zu unterscheiden:

1. Nachdem der Benutzer beispielsweise einen Link zu einem auf einer Webseite enthaltenen Video ausgewählt hat, sendet der Web-Browser eine Anfrage nach diesem Link an den Web-Server; z. B.:

```
GET /fhfulda.sdp HTTP/1.1
Host: www.fh-fulda.de
Accept: application/sdp
```

2. Der Web-Server beantwortet die Anfrage mit einer Meta-Datei, die den URL für den Stream enthält. Sie kann auch mehrere Stream-URLs enthalten, zwischen denen der Client wählen kann (z. B. nach Größe und Landessprache). Der Web-Browser gibt diese Datei nach dem Empfang an einen Media-Player, der auf dem Rechner des Anwenders installiert ist, weiter; z. B.:

```
HTTP/1.1 200 OK
Content-Type: application/sdp
v=0
o=- 1234567890 1234567890 IN IP4 192.168.78.1
s=RTSP Session
m=audio 0 RTP/AVP 0
a=control:rtsp://stream.fh-fulda.de/audio/start-german
m=video 0 RTP/AVP 31
a=control:rtsp://stream.fh-fulda.de/video/start-german
```

   Der Body der Response (ab v=0) stellt eine SDP-(Session Description Protocol)-Datei dar. SDP definiert die Meta-Datei einer RTSP-Sitzung und ist in [RFC_2327] ausführlich beschrieben.

3. Der Media-Player startet seinerseits eine Verbindung zum Streaming-Server, auf den der URL aus der Meta-Datei verweist. Er teilt diesem in der Methode SETUP mit, wie er zu erreichen ist und welchen Transport der Daten er wünscht, z. B. für die Übertragung des Audio-Streams:

```
SETUP rtsp://stream.fh-fulda.de/audio/start-german RTSP/1.0
CSeq: 1
Session: 12345678
Transport: RTP/AVP/UDP;unicast;client_port=3089-3090
```

4. Der Server quittiert die Anfrage des Client mit einer SETUP-Response. In dieser Nachricht sendet der Server außerdem die für die Sitzung ver-

wendeten Übertragungsparameter, unter Berücksichtigung der Wünsche des Clients im SETUP-Request, z. B. für den Audio-Stream:

```
RTSP/1.0 200 OK
CSeq: 1
Session: 12345678
Transport:  RTP/AVP/UDP;unicast;client_port=3089-3090;
            server_port=5001-5002
```

5. Nachdem die Sitzung per SETUP vollständig initialisiert ist, sendet der Client einen Request PLAY an den Server, um den Empfang des Streams zu beginnen. Dabei kann er z. B. eine bestimmte Stelle im Stream angeben, die er wiedergeben möchte, z. B.:

```
PLAY rtsp://stream.fh-fulda.de/audio/start-german RTSP/1.0
CSeq: 2
Session: 12345678
Range: smpte=0:30:00-
```

6. Der Server sendet eine kurze Bestätigung mit einer Zeitangabe. Unmittelbar nach der Response auf den Request PLAY beginnt der Server, den Stream nach den in SETUP ausgehandelten Übertragungsverfahren zu verschicken. Dabei kann z. B. das Protokoll RTP für den Versand der Daten zum Einsatz kommen, z. B.:

```
RTSP/1.0 200 OK
CSeq: 2
Session: 12345678
Range: smpte=0:30:00-1:20:00
RTP-Info: url=rtsp://stream.fh-fulda.de/audio/start-german;
          seq=23422234;rtptime=92342112
```

7. Der Client ordnet mit dem Request PAUSE die Unterbrechung des vom Server an ihn gesendeten Streams an. Der Server quittiert diesen Request.
8. Nachdem der Client die Wiedergabe des Streams beendet hat, teilt er dem Server mittels TEARDOWN mit, dass die Sitzung beendet ist. Der Server trennt daraufhin die Verbindung und schließt dadurch die Sitzung.

1.6.5 Transportprotokolle beim RTSP

Obwohl der Standard RTSP kein explizites Transportprotokoll definiert, wird im Zusammenhang mit RTSP meist das RTP (Real Time Protocol) verwendet. Dieses in [RFC_1889] beschriebene Protokoll nutzt für die Übertragung in der Regel UDP. Dabei ist RTP als Erweiterung der Transportschicht anzusehen. Es ermöglicht durch Sequenznummern und Zeitstempel, das Protokoll UDP für eine chronologische Übertragung zu nutzen, ohne dabei die Nachteile einer langsamen Übertragung wie beim TCP zu erzeugen.

Abbildung 1-22 zeigt die Struktur eines RTP-Pakets. Dabei sind vor allem die Felder *Payload Typ*, *Sequenznummer*, *Zeitstempel* und *SSRC* (Synchronization Source / Synchronisationsquelle) wichtig. Im Payload Typ wird angegeben, welches Format die im RTP enthaltenen Nutzdaten aufweisen. Die Sequenznummer sorgt genau wie die CSeq beim RTSP für eine chronologisch korrekte Verarbeitung von aufeinander folgenden RTP-Paketen. Durch die Sequenznummer wird eine Sicherungsfunktion ähnlich dem TCP möglich. So können z. B. stark verzögerte Pakete direkt nach dem Empfang verworfen werden. Außerdem können zu früh ankommende Pakete in einem Puffer bis zur Wiedergabe zwischengespeichert werden. Der zeitaufwendige wiederholte Übertragungsvorgang beim TCP, der im Streaming-Bereich keinen Sinn erfüllt, entfällt. So werden z. B. Audio-Signale, die zu spät ihr Ziel erreichen, gezielt verworfen, anstatt sie wiederzugeben, was zu einem Echo oder zu einer Störung der Wiedergabe führen würde. Neben der Sequenznummer kann der Empfänger eines RTP-Streams außerdem anhand des Zeitstempels des Pakets erkennen, in welchem zeitlichen Kontext dies zum Gesamtsignal steht. Dadurch kann eine Echtzeitwiedergabe beim Client erreicht werden. Die *SSRC-ID* stellt eine Identifikation der Synchronisationsquelle dar. Grundsätzlich handelt es sich um eine ID in Form einer hinreichend großen Zahl, die jeden Stream eindeutig identifiziert. Dadurch wird die Synchronisierung verschiedener Streams, z. B. getrenntes Audio und Video, beim Client möglich. Die SSRC wird zu Beginn einer Sitzung vom Server vergeben. Sollte die gleiche SSRC-ID bereits für einen anderen Stream vergeben sein, wird eine neue SSRC-ID verhandelt.

Die restlichen Felder wie *Version* usw. sind für das Verständnis des Streamings per RTP weniger von Bedeutung. Ihre Beschreibung kann in [RFC_1889] genauer nachgelesen werden.

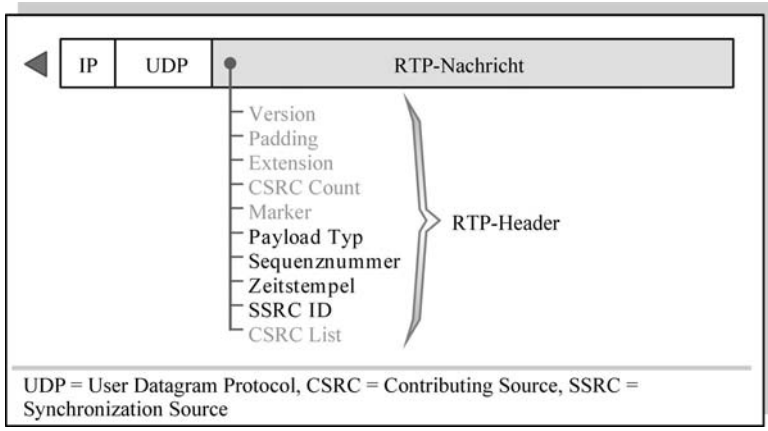

Abb. 1-22: Struktur eines RTP-Pakets

Neben RTSP und RTP wird in einer Streaming-Sitzung häufig das RTCP (Real Time Transport Control Protocol) verwendet. Dieses Protokoll sendet per Multicast oder Unicast an alle Empfänger eines Streams statistische Daten und sammelt selbige von den Clients. Dadurch wird z. B. eine Anpassung der Wiedergabe während der Wiedergabe des Streams möglich. So kann, z. B. wenn sich die Netzwerksituation durch Überlastung in Richtung Client verschlechtert, durch die Auswertung der auftretenden Jitter oder Paketverluste auf ein anderes Format oder eine geringere Auflösung zurück geschaltet werden. Damit das RTCP besonders bei schmalbandigen Übertragungen nicht selbst unnötig wertvolle Bandbreite verschlingt, wird das Intervall und der Umfang der gesendeten Informationen an die verfügbare Bandbreite angepasst.

1.7 MMS – Microsoft Media Server Protokoll

Microsoft bietet in seiner Produktfamilie die Windows Media Services und das damit verbundene Protokoll MMS (Microsoft Media Server) an. Da die gesamte zum Streaming notwendige Infrastruktur monopolistisch von Microsoft gestellt wird, sieht das Unternehmen leider keine Notwendigkeit, diesen proprietären Standard offen zu legen. Dies hat für Microsoft und seine Kunden den entscheidenden Vorteil, dass es z. B. keine Software zum Speichern von Streams per MMS gibt, und somit auch die Entwicklung von Software, die illegalerweise digitaler Rechte von Microsofts DRM (Digital Rights Management) aushebeln könnte, wirkungsvoll unterbunden wird.

Daher arbeitet das Protokoll MMS auch nicht mit Klartext-Nachrichten wie sein Konkurrent RTSP. Stattdessen werden Kommandos und Parameter binär kodiert (als Hex-Werte). Diese kryptische Verschleierung der Übertragung stellt sowohl für Netzwerkadministratoren, z. B. bei der Planung von Firewalls, als auch für Streaming-Media-Architekten bei der Auswahl des geeigneten Formats einen großen Wermutstropfen dar. Trotzdem soll dieses Kapitel auf der Basis von Analysen und Recherchen im Internet einen Vergleich zwischen RTSP und MMS ermöglichen.

MMS regelt sowohl Kontrollmechanismen eines Streams als auch den Transport selbst. Dabei kann ein Stream innerhalb des MMS mittels HTTP, TCP und UDP übertragen werden. Das jeweilige Protokoll wird zu Beginn einer Sitzung optimal für den Client ausgehandelt. Für TCP sowie UDP wurde dem MMS serverseitig der Port 1755 zugewiesen.

1.7.1 Phasen einer MMS-Sitzung

Eine MMS-Sitzung kann in verschiedene Phasen unterteilt werden. Abbildung 1-23 zeigt den Ablauf dieser einzelnen Phasen. Folgende Phasen werden dabei unterschieden:

- Web-Verbindung

  Diese erste Phase beginnt bereits vor der eigentlichen MMS-Sitzung und baut eine Verbindung zu einem Web-Server auf. Dabei empfängt der Web-Browser eine Meta-Datei, die die Adresse des gewünschten Streams enthält.

- Aufbau der Kontrollverbindung

  In der nächsten Phase wird seitens des Clients eine MMS-Verbindung als Kontrollverbindung zum Server aufgebaut. Dadurch beginnt die MMS-Sitzung. Die Sitzung bleibt bis zum Ende der Verbindung bestehen, wobei der eigentliche Stream „out-of-band“ über das Transportprotokoll zum Client gesendet wird. Beim Aufbau der Kontrollverbindung erhält der Client zusätzlich alle nötigen Informationen, wie z. B. verschiedene Tonspuren, Länge des Streams usw., vom Server.

- Test der Verbindungseigenschaften

  Beim MMS werden vor dem Versenden des Streams die Verbindungseigenschaften zwischen Server und Client ermittelt. Dabei versendet der Server einige Testpakete in Richtung Client, aus deren Verzögerung und Empfangsrate der Client die Güte der Verbindung ermitteln kann. Dadurch kann er z. B. einen Stream mit niedrigerer Qualität vom Server

anfordern, um sich evtl. schlechten Verbindungseigenschaften anzupassen.

- Medienauswahl

  Anhand der Informationen über die einzelnen Streams kann der Client schließlich den gewünschten Stream auswählen. So kann er z. B. für einen Stream die Tonspur in seiner Landessprache auswählen oder einen Stream in einer niedrigen Auflösung anfordern, um sich an die aus seiner Sicht größtmögliche Übertragungsrate anzupassen.

- Aufbau der Transportverbindung

  Im Anschluss an die Auswahl des gewünschten Streams kann in dieser Phase die Transportverbindung aufgebaut werden.

- Medienkontrolle

  Während der Übertragung kann der Betrachter auf unterschiedliche Art und Weise Einfluss auf die Wiedergabe des Streams nehmen. So kann er den Stream z. B. kurzzeitig unterbrechen oder vor- bzw. zurückspulen.

- Transportüberwachung

  Der Transport zum Client wird vom Server ständig überwacht. Dabei werden statistische Daten über den übertragenen Stream ausgewertet. Beim MMS wird ein proprietäres Protokoll verwendet, das eine Auswertung der empfangenen Daten beim Client sowie den zeitlichen Versatz (Verzögerung und Jitter) ermöglicht. Dadurch kann der Client während der Übertragung auf einen Stream mit einer niedrigeren Übertragungsrate zurückschalten, um automatisch eine verbesserte, z. B. flüssigere, Wiedergabe beim Client zu erreichen.

- Abbau der Kontrollverbindung

  Beim MMS werden Transport- und Kontrollverbindung in einem Schritt abgebaut. Dies ist möglich, da MMS ein eigenes Transportprotokoll definiert (siehe Abschnitt 1.7.3). In dieser Phase wird damit auch die MMS-Sitzung beendet.

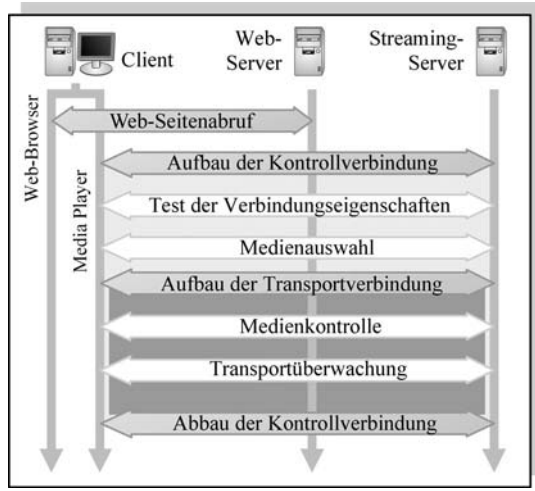

Abb. 1-23: Phasen einer MMS-Sitzung

1.7.2 Typischer Verlauf einer MMS-Sitzung

Abbildung 1-24 zeigt die relevanten Teile des Ablaufs einer MMS-Verbindung. Die konkreten Kommandos können, da ein offener Standard hierfür nicht verfügbar ist, bei unterschiedlichen Versionen des Windows Media Players bzw. Windows Media Servers variieren. Das Beispiel lässt bereits erkennen, dass das MMS ein relativ komplexes Protokoll darstellt.

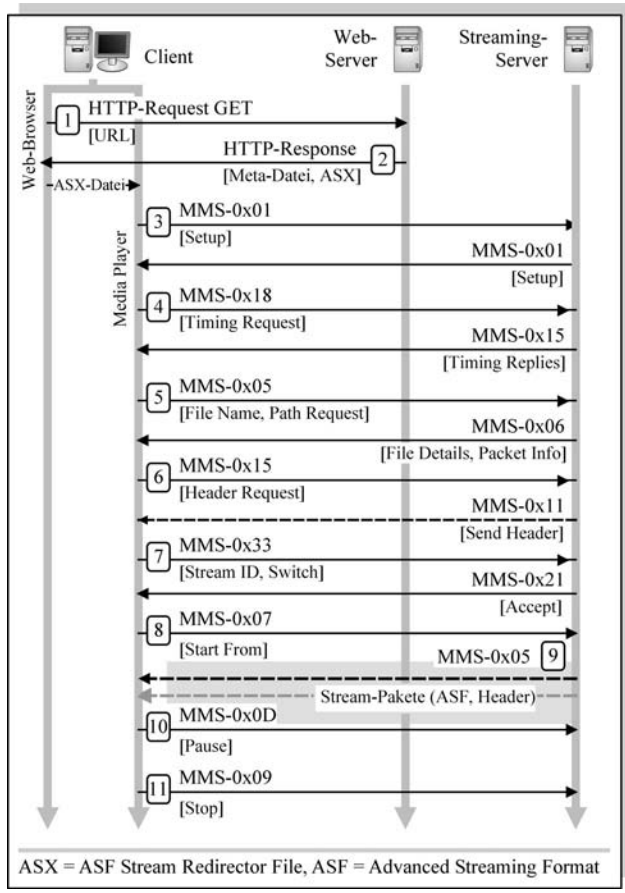

Abb. 1-24: Struktur des Ablaufs einer MMS-Verbindung

Bei dem in Abbildung 1-24 gezeigten Ablauf werden die folgenden Schritte unterschieden:

1. Durch Auswahl eines Links im Web-Browser wird ein Video angefordert.

2. Der Web-Server antwortet auf die Anfrage mit einem ASX (ASF Stream Redirector File). Dabei handelt es sich um eine XML-Datei, die das Pendant zur Meta-Datei beim RTSP darstellt. Z. B. kann diese enthalten:

```
<ASX version="3.0">
 <Entry>
  <Ref href="mms://stream.fh-fulda.de/video/start-ger.wmv"/>
 </Entry>
</ASX>
```

3. Das ASX-File wird an den Media-Player übergeben, der seinerseits eine Verbindung zum enthaltenen URI herstellt. In einem ASX-File können auch mehrere Streams enthalten sein, die synchronisiert wiedergegeben werden. Dabei wird per Kommando 0x01 z. B. Version, Hostname usw. zwischen Client und Server ausgetauscht. Außerdem sendet der Client Informationen über den verwendeten Internetzugang (Providername, Benutzer-Login).
4. Im Anschluss initiiert der Client mit 0x18 einen Timing Request, den der Server mit ein paar zufälligen Paketen in 0x15 beantwortet. Dadurch können Latenzzeiten und Jitter der Verbindung geschätzt werden.
5. Mit 0x05 sendet der Client eine Anfrage nach dem gewünschten Dateinamen und dessen Pfad. Dies kann für mehrere im ASX enthaltene Streams erfolgen. Der Server antwortet per 0x06 und mit Paketinformationen sowie Länge und Kodierung des Streams.
6. Durch 0x15 fordert der Client den Server zum Senden des Headers auf. Der Header enthält umfassende Zusatzinformationen zum gewünschten Stream.
7. Schließlich entscheidet der Client, welchen Stream aus dem ASX-File er erhalten will. Hier kann durch den sog. Switch z. B. ein Stream in der Landessprache oder in gesonderter Qualität automatisch angefordert werden.
8. Durch das Kommando 0x07 entscheidet der Client, ab welchem Frame er die Wiedergabe beginnen möchte. Dies kann mit dem Parameter Range der Methode PLAY beim RTSP verglichen werden.
9. Ab dem Kommando 0x05 vom Server wird der Stream in Form von MMS-Paketen mit ASF-Inhalt per HTTP, TCP oder UDP zum Client übertragen.
10. Ein 0x0D entspricht in ungefähr der Methode PAUSE beim RTSP.
11. 0x09 beendet die Wiedergabe und schließt die Verbindung.

Ähnlich wie beim RTCP sendet der Windows Media Server etwa jede Minute eine Statistik über den Verlauf des Streams an den Client und fragt Daten von diesem ab, wodurch eine Anpassung des Streams an aktuelle Netzwerksituationen möglich wird. Erhält der Server keine Nachricht vom Client, trennt er die Verbindung automatisch.

1.7.3 Transportprotokolle beim MMS

Streng genommen definiert das MMS kein eigenes Transportprotokoll. Es kann sowohl TCP, UDP als auch HTTP für den Transport seiner Daten nutzen. Die übertragenen Pakete weisen jedoch eine RTP-ähnliche Struktur auf. Dabei kommen ASF(Advanced Streaming Format)-Pakete beim MMS zum Einsatz. Dieses Format wurde von Mircosoft offen gelegt und kann unter [MSASF] eingesehen werden. Grundsätzlich lassen sich ASF-Streams als Audio/Video-Komponenten verstehen, die unter anderem als Video in WMV(Windows Media Video)-, als Audio in WMA(Windows Media Audio)-Files gespeichert werden können. Ein ASF-Frame hat in Zusammenhang mit dem Protokoll IP die in Abbildung 1-25 gezeigte Struktur. Dabei können mehrere ASF-Segmente in einem einzigen IP-Paket per TCP oder UDP transportiert werden. Den einzelnen Segmenten ist ein MMS-Header vorangestellt, der vor allem die Sequenznummer analog zum RTP enthält. In den Segmenten selbst ist zusätzlich ein ASF-Zeitstempel gespeichert, der die Synchronisierung von einzelnen Streams wie beim RTP ermöglicht.

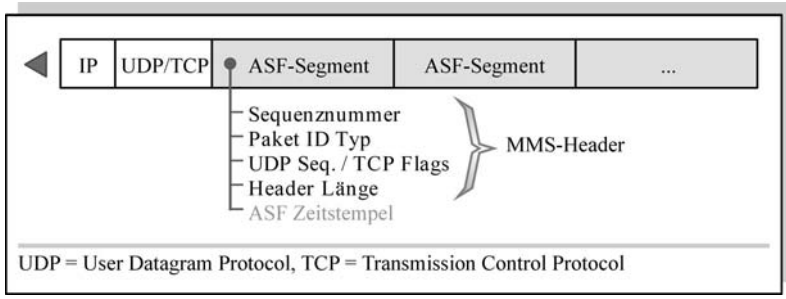

Abb. 1-25: Struktur eines ASF-Pakets

1.8 Streaming-Media-Perspektiven

Streaming-Media bietet ein großes Potential für die Zukunft des Internet. Es bietet gewissermaßen die Verschmelzung sämtlicher Medien, wie z. B. Radio und Fernsehen, zu einem einzigen - dem Internet. Auch wenn diese Verschmelzung vielleicht nie vollends verwirklicht wird, bieten sich durch Streaming-Inhalte im Web völlig neue Möglichkeiten und Dienste. Im Fol-

genden sollen einige dieser Möglichkeiten fixiert werden. Der interessierte Leser kann aus ihnen vielleicht auch Denkanstösse für völlig andere Ansätze gewinnen. Streaming-Media bietet durch die multimediale Verschmelzung eine regelrechte Plattform für kreative Ideen.

Internet-Radio

Bereits heute senden viele Radiostationen ihre Hörfunk-Programme auch über das Internet, was häufig als Web-Radio oder Internet-Radio bezeichnet wird. Während diese Möglichkeit derzeit häufig als Begleitmusik beim Surfen o. ä. genutzt wird, könnte sich in Zukunft eine Art interaktives Radio abzeichnen. Viele Internet-Radios bieten zusätzliche Informationen wie aktuelle Nachrichten, Quiz-Sendungen oder einfach nur den Interpreten des laufenden Titels auf einer zugehörigen Web-Seite. Die Möglichkeiten von Streaming-Media im Internet gehen allerdings weit darüber hinaus. So könnte z. B. in der Zukunft theoretisch jeder Radio-Hörer sein eigenes Musik-Programm wählen, das dann individuell an ihn als Stream gesendet wird. Dies würde natürlich die Übertragung als Multicast aushebeln, käme aber für eine zahlende Kundschaft durchaus in Frage. Aber auch Gewinnspiele oder Begleitinformationen lassen sich über das Internet sehr viel einfacher an die Zuschauer verteilen, als über eine aufwendige Hotline. Außerdem ist diese Möglichkeit sehr viel kostensparender. Beim Radio per Streaming-Media wäre auch möglich (siehe RECORD Methode des RTSP), dass einzelne Hörer direkt und ohne den Umweg über das Telefon live in die Sendung geschaltet werden. Eine Möglichkeit, die für Talk-Shows usw. interessant ist. Neben diesen Möglichkeiten ist ebenfalls interessant, dass man per Streaming-Media auch Radio-Sender aus der ganzen Welt an seinem Rechner empfangen kann.

E-Education

Einer der größten Wachstumsmärkte könnte das Umfeld des E-Education werden. So könnten Vorlesungen an Universitäten oder Vorträge aller Art über ein hochauflösendes Video z. B. nach Hause oder direkt an die Teilnehmer im Hörsaal verschickt werden. Die Teilnehmer könnten auch ebenfalls per Mikrophon über ihren eigenen Laptop eine Frage an den Dozenten stellen. Eine interessante Möglichkeit stellt auch die individuelle Aufnahme einer solchen Sitzung oder zumindest (um Kopien der Aufzeichnung zu vermeiden) die Bereitstellung des Streams zu einem späteren Zeitpunkt für die Teilnehmer dar. Diese könnten dann beim Nacharbeiten der Vorlesung z. B. in derselbigen hin- und herspulen, um den Stoff korrekt aufzuarbeiten. Auf diese Weise könnten auch Folien, Tafelanschriebe usw. als Stream gesendet werden. Auch wenn diese Möglichkeiten wohl kaum den regulären Lehrbe-

trieb ersetzen können, bieten sie vielleicht in naher Zukunft bereits eine interessante Erweiterung.

Internet-TV und interaktives Fernsehen

Lange Zeit wurden Begriffe wie Video-On-Demand und interaktives Fernsehen als Zukunft der betagten Mattscheibe proklamiert. Diese Schlagwörter und teilweise sicherlich auch Luftblasen sind weitgehend verschwunden. Die Aufwertung des Fernsehens durch Begleitinformationen im Internet zum laufenden Programm ist jedoch gängige Praxis geworden. Kaum eine TV-Sendung schmückt sich nicht mit dem dezenten Hinweis auf eine Web-Seite am Ende der Sendezeit. In der Zukunft ist ein interaktives Fernsehen denkbar, bei dem Zuschauer per Web-Seite zur Sendung nicht nur Informationen über diese erhalten, sondern auch deren Verlauf (z. B. bei einer Talk-Show durch Fragen der Surfer) mitbestimmen könnten. Auch die Auswahl des gewünschten Spielfilms (z. B. von einigen verschiedenen per Multicast oder einen individuellen per Unicast) im Abendprogramm wäre denkbar. Außerdem wird durch Streaming-Media praktisch das weltweite Fernseh-Programm verfügbar, direkt am heimischen Fernseher bzw. Rechner.

Internet-Telefonie und Videokonferenzen

Bereits heute nutzen viele Menschen das Internet für Videokonferenzen und Internet-Telefonie. Dabei hat sich diese Domäne längst vom unternehmensnahen Umfeld in die Privathaushalte erweitert. Die Kostenvorteile sind besonders für die Privatnutzer interessant, wobei die Unternehmen durch die zusätzliche Nutzung von Gruppenkonferenzen, Whiteboards oder „application sharing“ eine sehr viel effizientere Kommunikation mit den Geschäftspartnern oder z. B. Außenstellen erhalten. Die zentrale Aufgabe des Streaming-Media-Umfelds in der Zukunft könnte daher sein, sich in kleinen mobilen Geräten integrieren zu lassen. So hätte man Video-Telefon, Radio, Fernseher, aber auch die Kamera immer verfügbar.

2. Multicasting-Anwendungen

Die Übertragung von Streaming-Media setzt sehr große Bandbreiten voraus. Insbesondere stellt die Tatsache, dass das Übertragungsvolumen proportional zur Anzahl der Teilnehmer wächst, ein großes Problem dar. Der Kernbestandteil dieses Problems ist dabei die separate Übertragung der Streams an die einzelnen Teilnehmer (als Unicast – Punkt-zu-Punkt). Als Lösungsansatz existiert im Netzwerkumfeld das Multicasting-Verfahren. Dabei werden die Streams nicht separat, sondern als ein einziger Stream an alle Teilnehmer gleichzeitig (als Multicast – Punkt-zu-Mehrpunkt) verschickt. Aufgrund der vielschichtigen und hierarchischen Struktur des Internet, die ein Übertragen der Daten an mehrere Teilnehmer nicht direkt unterstützt, ist die Akzeptanz von Multicasting derzeit noch relativ gering. Es existieren nur wenige „Multicast-Inseln“ im Internet, die per Unicast-Tunnel untereinander verbunden sind. Trotzdem bieten vor allem Media-Encoder und Media-Player nahezu geschlossen Multicasting-Fähigkeiten an. Die Übertragung von Streaming-Media an mehrere Teilnehmer stellt daher eine der wichtigsten Multicasting-Anwendungen dar.

2.1 Multicasting-Strukturen

Um den Einsatz von Multicasting in einem bestehenden Netzwerk zu realisieren, sind einige zusätzliche Konfigurationen notwendig. Diese Konfigurationen hängen alle mit den Multicasting-Strukturen zusammen. Sie beziehen sich damit auf die Verteilung von identischen Daten an mehrere (eine Gruppe) Empfänger in einem Netzwerk.

2.1.1 Klassifizierung von Multicasting

Bei der Adressierung von Übertragungen in einem Netzwerk können am Sender verschiedene Arten der Informationsverteilung und deren zugehörige Verfahren unterschieden werden. Dabei bezieht sich die Informationsverteilung auf die Zuordnung von Sendern zu Empfängern. Sowohl Sender als auch Empfänger können dabei einzeln oder auch als Gruppe von mehreren Sendern oder Empfängern verstanden werden. Abbildung 2-1 zeigt die beiden klassischen Arten der Informationsverteilung (Adressierung) in einem Netzwerk.

Abbildung 2-1 a) zeigt die einfachste Art der Adressierung einer Übertragung – den Unicast. Dabei überträgt genau ein Sender Informationen zu genau einem Empfänger. Diese Art der Übertragung wird auch als Punkt-zu-Punkt bezeichnet, da der Sender als Punkt genau eine Verbindung (Kante) besitzt, die zur Gegenstelle der Übertragung führt. Eine Unicast-Übertragung lässt sich z. B. mit einem Telefongespräch vergleichen, bei dem sich zwei Personen „Punkt-zu-Punkt“ unterhalten.

Abbildung 2-1 b) stellt die Adressierung von mehreren Empfängern gleichzeitig dar – den sog. Broadcast. Dabei werden die zu übertragenden Informationen gleichzeitig an mehrere Empfänger verteilt. Diese Übertragung wird auch als Punkt-zu-Mehrpunkt bezeichnet, da der Sender als Punkt mehrere Verbindungen (Kanten) besitzt, die zu einer Vielzahl von Gegenstellen führen. Genauer gesagt müsste man beim Broadcast von Punkt-zu-Allen sprechen, da der Sender hier alle möglichen Empfänger adressiert. Eine Broadcast-Übertragung kann z. B. mit der Ausstrahlung von Radio- und Fernsehprogrammen verglichen werden, bei denen ein Sender gleichzeitig an mehrere Empfänger (Haushalte) „Punkt-zu-Mehrpunkt“ überträgt.

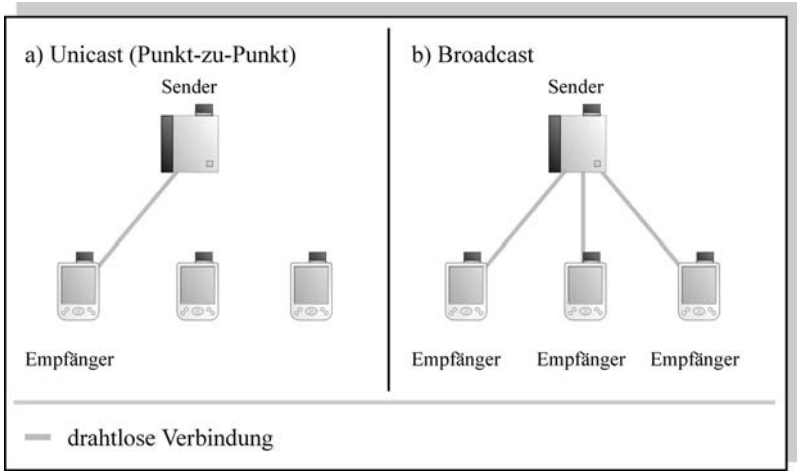

Abb. 2-1: Unicast a) und Broadcast b)

Anhand des Beispiels der Übertragung von Radio- und Fernsehprogrammen lässt sich auch die Multicast-Übertragung verstehen, die für diese Arbeit von zentraler Bedeutung ist.

Ein Fernsehgerät in einem Haushalt kann in der Regel genau ein Fernsehprogramm wiedergeben, wobei dieses Programm jedoch aus einer Vielzahl von Programmen gewählt werden kann. Somit überträgt ein Fernsehsender zwar rein technisch sein Programm an alle angeschlossenen Haushalte (zwischen Sender und Empfänger), bezieht man die Übertragung der Information jedoch auf Sender und Zuschauer, so findet keine Broadcast-Übertragung statt, bei der jeder Zuschauer, an jedem Fernsehgerät jedes Programm sehen müsste. Man könnte diesen Fall als selektives Broadcast-Verfahren (*Broadcasting*) bezeichnen. Dabei wird das Fernsehprogramm selektiv von den Zuschauern empfangen, die den Kanal gewählt haben, auf dem der Sender überträgt. In diesem Zusammenhang spricht man von einem *Multicasting* oder einer Multicast-Übertragung. Dabei kann ein Multicast als selektiver Broadcast verstanden werden.

Die Abbildung 2-2 a) zeigt eine solche Übertragung an eine Gruppe von mehreren Empfängern.

Beim Multicasting können, im Gegensatz zum vorherigen Beispiel mit dem Zuschauer eines Fernsehprogramms, auch mehrere Sender ein und denselben Empfänger adressieren. Dieser Fall wird in Abbildung 2-2 b) gezeigt

und beschreibt einen Multicast mit mehreren Sendern. Dabei bildet jeder Sender separat mit seinen Empfängern eine Multicast-Gruppe. D. h., dass bei einer Multicast-Übertragung pro Gruppe genau ein Sender existieren kann, während eine Vielzahl von Empfängern möglich ist.

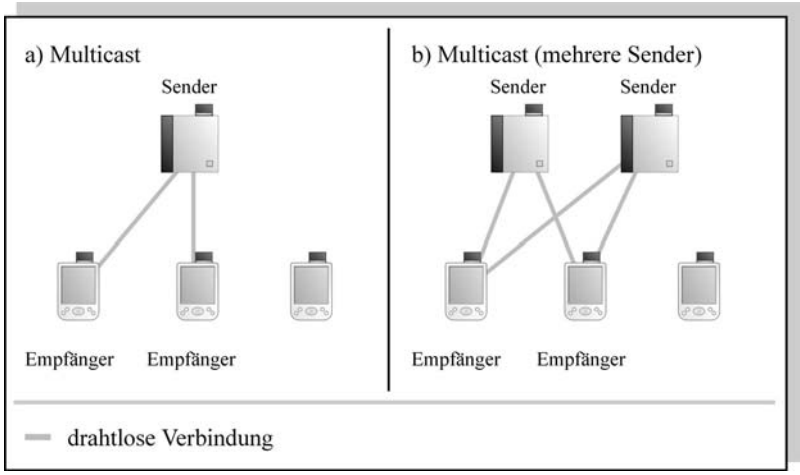

Abb. 2-2: Multicast mit a) einem und b) mehreren Sendern

2.1.2 IP-Multicasting

Die Übertragung der Multicast-Struktur aus dem Abschnitt 2.1.1 auf das Protokoll IP, das als Basis für die in dieser Arbeit verwendeten Netzwerke dient, stellt einige Anforderungen. Da Multicasting eine bestimmte Form der Adressierung darstellt, wird zunächst eine gesonderte Form von Adressen benötigt. Normale IP-Adressen verweisen auf genau einen Empfänger. Multicast-IP-Adressen müssen jedoch eine Gruppe von möglichen Empfängern adressieren. Um dies zu erreichen, wurde innerhalb der IP-Adressen ein gesonderter Adressbereich (224.0.0.0 bis 239.255.255.255) für sog. Multicast-Adressen reserviert [RFC_1700] (siehe auch [RFC_3232]). Diese Adressen können für Gruppen vergeben werden, in denen ein Sender per Multicasting Nachrichten an alle anderen Teilnehmer der Gruppe versendet.

In [RFC_1700] sind außerdem feste Multicast-Gruppen definiert. So adressiert die Gruppe 224.0.0.1 z. B. alle Multicast-Rechner, die Gruppe 224.0.0.2 alle Multicast-Router. Dadurch kann ein Router relativ einfach alle seine Multicast-Rechner (mittels der Gruppe 224.0.0.1) erreichen und ein einzelner Rechner (über die Gruppe 224.0.0.2) alle Multicast-Router in seinem Bereich finden. Wichtig ist in diesem Zusammenhang, dass

der Router z. B. für die Abfrage aller Hosts mittels 224.0.0.1 eine TTL (Time To Live) von 1 für das Paket vergibt. Damit kann das Paket nur im eigenen Subnetz, nicht aber über weitere Router hinweg beantwortet werden. Ohne diese Angabe würde der Router zusätzlich Antworten von Rechnern (im schlimmsten Fall aus der ganzen Welt) von seinen Nachbar-Routern bekommen.

Die Gruppenzugehörigkeit muss dynamisch zugeordnet werden können und insgesamt verwaltbar sein. Beim IP-Multicasting ist dafür ein gesondertes Verwaltungsprotokoll vorgesehen, das Protokoll IGMP.

IGMP – Internet Group Management Protocol

Das Protokoll IGMP (Internet Group Management Protocol) ist in der Version 2 nach [RFC_2236] spezifiziert. Die Version 3 des IGMP befindet sich bereits in der Spezifikation, wird allerdings bis dato nicht verwendet. IGMP operiert zwischen einem Rechner und dem aus seiner Sicht nächstliegenden Router. Der Router dient als Verteiler für die Daten, die von außen in das Subnetz gelangen. Dabei kann der Router unter Verwendung des IGMP bestimmen, welche Rechner in seinem Subnetz Multicast-Nachrichten empfangen möchten.

In Abschnitt 2.1.2 wurde bereits eingangs erwähnt, dass Multicast-Adressen Gruppen von Hosts repräsentieren. Somit kann der Router bestimmte Multicast-Gruppen verwalten und die an sie adressierten Pakete in sein Subnetz versenden. Obwohl der Zusatz „Group Management“ im Namen des Protokolls eine Gruppenverwaltung suggeriert, kann diese nur bedingt erreicht werden. Da das IGMP lediglich zwischen einem Router und den Rechnern in dem an ihn grenzenden Subnetz genutzt wird, ist eine Verwaltung der Gruppenzugehörigkeiten über Router hinweg nicht möglich. Somit kann z. B. keine Aussage darüber getroffen werden, welcher Rechner weltweit derzeit Mitglied in welcher Multicast-Gruppe ist. Ebenso ist eine Verwaltung z. B. durch Entfernen eines bestimmten Rechners nur im Subnetz (d. h. nicht über einen Router hinweg) möglich.

Beim IGMP (in der Version 2) werden drei Nachrichtentypen unterschieden:

- Membership Query

  Mittels der Nachricht Membership Query kann ein Router Teilnehmer einer Multicast-Gruppe in seinem angeschlossenen Subnetz feststellen. Bei einer Membership Query wird zwischen einer selektiven und einer allgemeinen Abfrage unterschieden. Die selektive Abfrage ermittelt die Teilnehmer einer bestimmten Multicast-Gruppe, während die allgemeine Abfrage alle aktiv genutzten Multicast-Gruppen im Subnetz zurück lie-

fert. Dabei ermittelt die selektive Abfrage jedoch nicht sämtliche Adressen der Teilnehmer im Subnetz. Sie stellt lediglich die Information für den Router zur Verfügung, ob die gewählte Multicast-Gruppe überhaupt im Subnetz empfangen wird.

Der Router sendet die selektive Abfrage direkt an die Adresse der Multicast-Gruppe. Aufgrund dieser Tatsache müssten auf diese Anfrage theoretisch alle in diesem Subnetz teilnehmenden Rechner dieser Gruppe eine Antwort an den Router zurücksenden. Da dies jedoch vor allem in großen Netzen eine sehr große Bandbreitenverschwendung zur Folge hätte, die obendrein vollkommen redundante Daten übertragen würde, antwortet beim IGMP nur ein Rechner auf die Anfrage des Routers. Dies wird erreicht, indem jeder Empfänger eines Membership Query eine zufällige Zeit vor dem Versenden der Antwort abwartet.

Die maximale Dauer dieses zufälligen Zeitraums gibt das Feld *MRT* (maximale Antwortzeit) im IGMP-Header, der in Abbildung 2-3 gezeigt ist, an. Empfängt er während des Abwartens dieses Zeitraums bereits eine Antwort auf der Adresse der Multicast-Gruppe, so kann er davon ausgehen, dass bei einem anderen Teilnehmer die zufällige Wartezeit früher abgelaufen ist und die Anfrage des Routers damit beantwortet ist. In diesem Fall verwirft er seine eigene Antwort, ohne eine weitere Nachricht an die Multicast-Gruppe resp. den Router zu senden. Der Router hat somit keinerlei Information darüber, wie viele oder gar welche Rechner in seinem Subnetz Teilnehmer einer bestimmten Multicast-Gruppe sind. Dies ist jedoch auch nicht für ihn von Belang. Um die gewünschten Multicast-Inhalte von seinen angrenzenden Nachbar-Routern bzw. über das Netzwerk zu routen, benötigt er lediglich die Information, ob überhaupt Empfänger der Multicast-Gruppe in seinem Subnetz existieren, was er durch die Membership Query bestimmen kann. Antwortet auf einen Membership Query kein einziger Rechner aus dem angeschlossenen Subnetz, so kann der Router die Multicast-Gruppe schließen und etwaige zugehörige Verbindungen zu seinen Nachbar-Routern schließen.

- Membership Report

  Die Membership Response stellt die Antwort auf eine Membership Query dar. Zusätzlich kann ein Rechner jedoch auch ohne vorheriges Empfangen einer Membership Query mittels Membership Report einer gewünschten Multicast-Gruppe beitreten. Über eine Membership-Report-Nachricht kann ein Rechner somit eine bestimmte Multicast-Gruppe bei dem nächstgelegenen Router seines Subnetzes „bestellen“. Jede Membership Response enthält die Adresse einer Multicast-Gruppe,

die ein Rechner bereits empfängt oder der er beitreten möchte.

- Leave Group

  Durch das Versenden der Nachricht Leave Group kann ein Rechner explizit aus einer Multicast-Gruppe austreten. Die Verwendung dieser Nachricht ist dabei optional, da der Router eine Multicast-Gruppe ebenfalls schließt, wenn (wie bereits bei der Membership Query beschrieben) kein Rechner auf eine Anfrage der entsprechenden Multicast-Gruppe antwortet. Ein Multicast-Router fragt daher in einem Intervall von ca. 100 Sekunden alle seine aktiven Gruppen ab. Damit wird eine Multicast-Gruppe auch dann geschlossen, wenn der teilnehmende Rechner sich nicht korrekt abmeldet (z. B. ausgeschaltet wird).

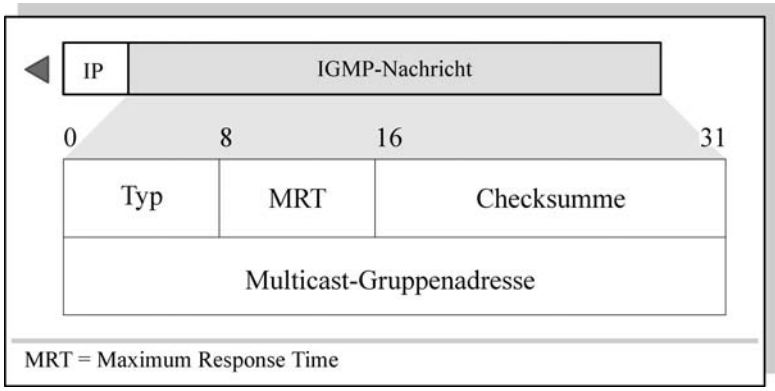

Abb. 2-3: Aufbau einer IGMP-Nachricht

Abbildung 2-3 zeigt die Struktur von IGMP-Nachrichten. Das Feld Typ enthält dabei einen Hex-Wert, der den verwendeten Nachrichttyp darstellt (z. B. 0x11 für eine Membership Query). Das Feld *MRT* (Maximum Response Time) gibt die maximale Wartezeit des Senders auf eine Antwort vor. Damit kann durch zufällige Wartezeiten die Anzahl von Antworten z. B. auf eine Membership Query vermindert werden. Die im Header enthaltene *Checksumme* bietet die Möglichkeit, fehlerhafte IGMP-Nachrichten zu erkennen und ggf. zu verwerfen. Der wichtigste Bestandteil einer IGMP-Nachricht ist die Multicast-Gruppenadresse, auf die sich die Nachricht bezieht. In diesem Feld steht z. B. die Adresse der gewünschten Multicast-Gruppe bei einer selektiven Membership Query.

Das IGMP regelt nur die Verwaltung von Multicast-Gruppen in ein- und demselben Subnetz. Für die Verwaltung von Multicast-Gruppen über Router hinweg wird ein Multicast-Routing-Protokoll benötigt, und der Router selbst

muss das Multicast-Routing unterstützen. Multicast-Routing-Protokolle werden in Abschnitt 2.2.1 genauer beschrieben. Eine ausführliche Beschreibung des IGMP sowie des IP-Multicasting allgemein findet sich in [BADA_01].

Nachteile des IP-Multicasting

Der augenscheinlich größte Nachteil des IP-Multicasting ist, dass vor allem bedingt durch das Protokoll IGMP weder ermittelt werden kann, welche Rechner zu einer bestimmten Multicast-Gruppe gehören, noch einzelne Rechner entfernt oder verwaltet werden können. Dieser Nachteil wird jedoch in der Praxis auf der Anwendungsebene implementiert. Es ist trotzdem zu beachten, dass das IP-Multicasting kein klassisches Vermittlungsprotokoll in IP-Netzen bietet. Es existiert praktisch kein eigenes Protokoll auf der Vermittlungsschicht (Schicht 3).

2.1.3 IP-Multicasting im LAN

Damit das IP-Multicasting in LANs möglich ist, müssen die IP-Multicast-Adressen auf Multicast-MAC-Adressen abgebildet werden. Ohne eine Umsetzung der IP-Multicast-Adressen müssten im LAN erneut mehrere separate Unicasts vom Router verschickt werden, und die Vorteile des IP Multicasting könnten nur zwischen den Routern zum Tragen kommen. In der Praxis ist auch für MAC-Adressen ein Multicast-Bereich eingeteilt. MAC-Adressen, die mit 01-00-5E [RFC_1700] beginnen, sind als Multicast-Adressen reserviert. Der OUI (Organisation Unique Identifier) stellt in gewöhnlichen MAC-Adressen den eindeutigen Herstellernamen des jeweiligen Controllers dar. Er bildet die ersten 24 Bit der insgesamt 48 Bit langen MAC-Adresse. Somit bleiben theoretisch 24 Bit zur Kodierung der IP-Multicast-Adresse über.

In der Praxis wird jedoch das erste Bit der verbleibenden 24 für die Unterscheidung zwischen Internet Multicast (1. Bit = 0) und einem von der IANA reservierten Adressbereich (1. Bit = 1) verwendet. Abbildung 2-4 zeigt die Abbildung der IP-Multicast-Adresse 224.0.0.1 (alle Multicast-Rechner) auf ihr zugehöriges MAC-Multicast-Äquivalent. Durch die Verkürzung des Multicast-Adressbereichs auf 23 Bit innerhalb der MAC-Adresse entstehen bei diesem Verfahren Adressüberschneidungen, die in der Praxis in Kauf genommen werden müssen.

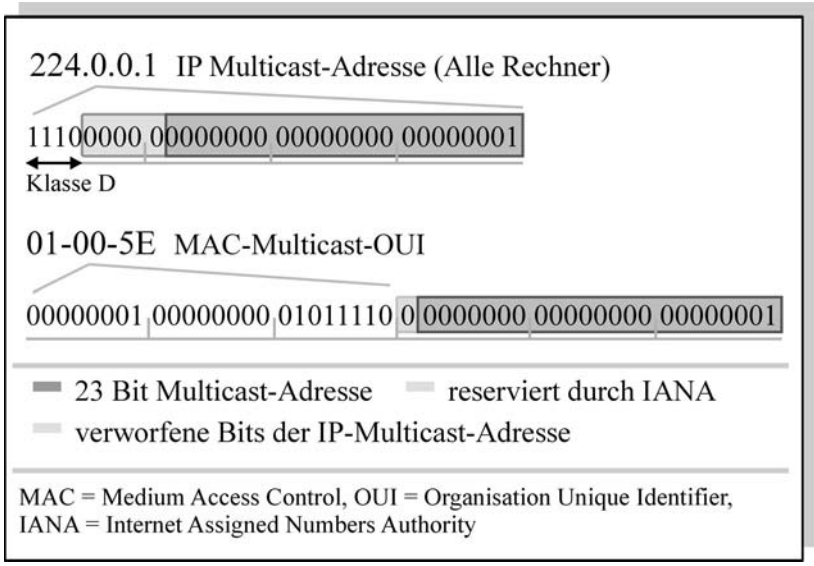

Abb. 2-4: Umsetzung von IP-Multicast-Adressen auf MAC-Multicast-Adressen

2.1.4 Zuverlässiges Multicasting für Streaming-Media

Bei der Übertragung von Multicasts in der in den vorherigen Abschnitten beschriebenen Art und Weise kann keine Gewährleistung für eine korrekte Übertragung der versendeten Daten übernommen werden. So kommen z. B. per Multicast versendete Pakete bei einem Teilnehmer, der von der Anzahl von Hops (Routern) her näher am Sender liegt, schneller an, als bei einem weiter entfernt liegenden Teilnehmer. Dies äußert sich mitunter in einer Verzögerung bei der Wiedergabe z. B. von Streaming-Media auf dem Rechner des Teilnehmers. Im Extremfall kann sogar eine Verfälschung der Daten durch Bitfehler während der Übertragung beim IP-Multicasting nicht festgestellt werden, was ebenfalls zu Fehlern bei der Wiedergabe führt.

Einzelne Entwicklungen und RFCs im Internet beschreiben daher die Spezifikation von sog. Reliable Multicast-Transportprotokollen (zuverlässigen Multicast-Protokollen), bei denen die Pakete z. B. Sequenznummern zugewiesen bekommen, durch die sich Verzögerungen und Laufzeitschwankungen (Jitter) korrigieren lassen.

In der Praxis wird ein zuverlässiges Multicasting insbesondere im Streaming-Media-Bereich jedoch durch die Anwendungsprotokolle erreicht. So kann z. B. das Protokoll RTP beim Streaming, wie im Abschnitt 1.6.5 beschrieben, verwendet werden, um per Multicasting übertragene Video- oder Audio-Daten beim Empfänger gezielt zu verwerfen oder zu puffern. Dadurch kann die Wiedergabequalität deutlich erhöht werden, ohne dabei auf zuverlässige Multicast-Protokolle aufsetzen zu müssen. Diese Protokolle werden zudem in den meisten IP-Netzen (auch im Internet) derzeit noch kaum unterstützt.

2.1.5 Multicasting-Anwendungen für Streaming-Media

Bereits in Abbildung 1-2 im Abschnitt 1.1.2 wurden die Nachteile aufgezeigt, die die paketvermittelnde Struktur des Internet auf Streaming-Media ausübt. Insbesondere die Bandbreitenverschwendung durch die separate Übertragung jedes Streams als Unicast wurde in Abschnitt 1.1.2 bemängelt. Diese Bandbreite lässt sich auf einen Bruchteil reduzieren, indem statt Unicast ein Multicast-Verfahren verwendet wird. Beim Multicasting bilden mehrere Empfänger, die sich im gleichen Netz befinden und den gleichen Stream empfangen wollen, eine Gruppe. Durch den Einsatz von Multicasting am Router, der das Netz, in dem sich die Gruppe befindet, beliefert, wird erreicht, dass an Stelle von mehreren Unicast-Übertragungen zu den einzelnen Mitgliedern der Gruppe nur eine einzige Multicast-Übertragung in das Netzwerk durchgeführt werden muss.

Der Vorteil von Multicasting für Streaming-Media liegt auf der Hand. Beim Streaming-Media werden identische, große Datenmengen in kurzen Zeiträumen an sehr viele Zuhörer bzw. Zuschauer verschickt. Nutzt man ausschließlich Unicasts, um diese Teilnehmer zu erreichen, so wächst die benötigte Bandbreite am Server und an den Routern unterwegs proportional zur Anzahl der ihnen untergeordneten Teilnehmer. Beim Multicasting bleibt die benötigte Übertragungsbandbreite sowohl für den Server als auch für alle Router auf dem Weg zum Client dagegen konstant. Dies illustriert die Abbildung 2-5.

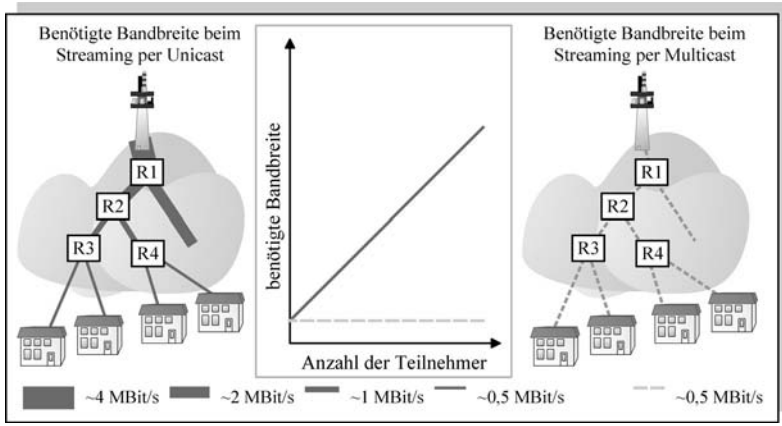

Abb. 2-5: Bandbreitenverteilung Unicast- und Multicast-Streaming an vier Haushalte

Anwendungsszenarien

- Verteilung von Verwaltungsinformationen beim Streaming

  Während der Übertragung eines Streams ermitteln Streaming-Protokolle bzw. deren Server in der Regel Statistiken über die Verteilung des Streams. Beim RTSP wurde in diesem Zusammenhang bereits das Protokoll RTCP vorgestellt. Damit solche Auswertungsverfahren, die lediglich statistische Daten übermitteln, nicht kostbare Bandbreite für die Streams selbst verschlingen, werden die Pakete des RTCP z. B. per Multicasting übertragen. Dadurch wird auch bei vielen Teilnehmern bzw. Empfängern des Streams nicht zu viel Bandbreite für statistische Informationen verschwendet.

- Verteilung der Streams am Rande des Backbone (Transitnetzes)

  Gerade am Rande eines Transitnetzes, z. B. bei einem ISP (Internet Service Provider), der ein großes Netz betreibt, in dem viele Benutzer potentiell den gleichen Stream empfangen wollen, kann sehr viel Bandbreite eingespart werden. Dafür müssen die Router beim ISP, die den Zugang zum Backbone herstellen, über IP-Multicasting verfügen. Von ihnen aus können dann sämtliche Anwender innerhalb des eigenen Netzes mit einem einzigen Stream versorgt werden, sofern die Router über ein Multicast-Routing verfügen.

Diese Möglichkeit ist besonders in großen Netzen mit verhältnismäßig geringer Bandbreite interessant. Als Beispiel sei ein drahtloses Netzwerk in einem Hörsaal genannt, das neben den Kunstwerken an der Tafel einen Stream, z. B. mit einer Präsentation, an die Zuhörer übertragen soll. Drahtlose Netzwerke verfügen meist über eine sehr geringe Bandbreite. Würden die Streams per Unicast verschickt würden schnell einige Zuhörer nur noch Artefakte der Präsentation sehen. Per Multicast dagegen würde der Stream nur einen Bruchteil der Bandbreite verschlingen, obendrein noch unabhängig von der Anzahl der Zuhörer. Ein ähnliches Szenario kann man sich z. B. am Flughafen in der Wartehalle der Zukunft mit Streaming besonderer Flughafen-Informationen usw. vorstellen.

Dabei ist Multicasting unabhängig von den verwendeten Streaming-Protokollen, da alle auf das Protokoll IP zurückgreifen.

Verteilung des Streams im Backbone

Der Idealfall wäre selbstverständlich der auch in Abbildung 2-5 dargestellte Zustand, in dem praktisch alle Router weltweit Streaming per IP-Multicasting unterstützen. Dadurch würde sehr viel Bandbreite gespart und schließlich die Qualität auch z. B. von Fernsehübertragungen im Internet von anderen Kontinenten in guter Qualität ermöglicht. Diese Infrastruktur ist jedoch heute kaum möglich. Es existieren nur wenige Multicast-Inseln, meist an Universitäten und Forschungseinrichtungen, und damit weit vom Massenmarkt des Endkunden entfernt. Zukünftige Internet-Technologien unterstützen das Multicasting jedoch ohne Einschränkung und es ist zu erwarten, dass die Verteilung von Streams per Multicasting auch im Backbone-Bereich weiter zunimmt.

2.2 Multicast Routing

Bereits in Abschnitt 2.1.2 wurde erwähnt, dass für eine Multicast-Übertragung über mehrere Router hinweg spezielle Multicast-Routing-Protokolle benötigt werden. Diese Protokolle müssen in der Lage sein, unterschiedliche Subnetze mit einzelnen Teilnehmern in verschiedenen Multicast-Gruppen zu verbinden. Die in IP-Netzen verwendeten Multicast-Routing-Protokolle stehen teilweise in direktem Zusammenhang mit den verwendeten Unicast-Protokollen. Im Internet existieren derzeit nur wenige Router (vorrangig an Universitäten und Forschungseinrichtungen), die ein Multicast Routing unterstützen. Ein Beispiel für die Unterstützung von Multicast Routing im Internet bildet das MBone [MBONE].

Am Rande sei bemerkt, dass Multicast-Routing-Protokolle in dieser Arbeit keine vorrangige Rolle spielen, da der Multicast zum Verteilen von

Streaming-Media in drahtlosen Netzwerken in der Regel direkt im Subnetz des Access Point per IGMP (siehe Abschnitt 2.1.2) zwischen dem Teilnehmer und einem Server, der z. B. zu einem CDN (*Content Delivery Network*) gehört, geregelt wird.

2.2.1 Grundlagen

Die vorrangige Anforderung an das Multicast Routing ist die Verteilung von Multicast-Nachrichten über mehrere Subnetze hinweg. Dabei können in den verschiedenen Subnetzen völlig unterschiedliche Multicast-Gruppen existieren. Ziel des Multicast Routing ist es, Teilnehmer einer solchen Gruppe mit Teilnehmern der gleichen Gruppe aus anderen Subnetzen (d. h. über mehrere Router hinweg) zu verbinden. Wie bereits in Abschnitt 2.1.1 erwähnt, stellt Multicasting einen selektiven Broadcast dar. Router können in ihrer Eigenschaft als Verbindungsstelle zwischen Subnetzen jedoch grundsätzlich keinen Broadcast übertragen. Es würde auch keinen denkbaren Nutzen erfüllen, auf eine Anfrage aus einem Subnetz eine Antwort von allen Rechnern aus einem IP-Netz (z. B. dem Internet) über mehrere Router hinweg zu erhalten. Außerdem würde sich der Broadcast theoretisch in jedem Subnetz vermehren, da er in diesem erneut an alle angeschlossenen Router weitergeleitet werden würde. Einen weiteren, wenn nicht größeren Nachteil würde die Verschwendung von Bandbreite und ggf. Übertragungskosten in kostspieligen, schmalbandigen WAN-Anbindungen durch die Vermehrung der Broadcasts darstellen.

Das skizzierte Szenario zeigt klar, dass ein Router grundsätzlich nicht in der Lage ist bzw. sein darf, eine Gruppe von Hosts bzw. Routern zu adressieren. Multicast-Routing-Protokolle lassen sich an Hand der Lösung der genannten Probleme beurteilen und klassifizieren.

Es existieren mehrere theoretische Verfahren, die als Basis für Multicast-Routing-Protokolle dienen:

- Flooding
- Improved Flooding

Multicast-Bäume:

- Steiner-Bäume
- Bäume mit Rendezvous-Stellen
- Quellenbasiertes Routing

Flooding

Das einfachste Verfahren zur Übertragung von Multicast-Nachrichten über mehrere Router hinweg stellt das sog. Flooding oder auch Fluten dar. Die Nachricht wird in diesem Fall analog zum Broadcast an alle Router im Netz weitergeleitet. Dabei entstehen die oben genannten Nachteile bei der Weiterleitung von Broadcasts über Router. Vor allem ist dabei die Bildung von Schleifen zu vermerken, bei der ein Router die Multicast-Nachricht an seine Nachbar-Router weiterleitet, die ggf. über weitere Router diese Nachricht schließlich wieder an ihn versenden. Ein Multicast-Paket würde somit so lange über diesen Weg zirkulieren, bis die TTL [BADA_01] (Lebenszeit des Pakets, wird in jedem Router dekrementiert) im IP-Header überschritten würde. Dieses Problem wird beim Improved Flooding vermieden.

Improved Flooding

Das Improved Flooding oder auch „verbesserte Fluten“ bezeichnet Ansätze, die die vom Flooding bekannten Probleme der Schleifenbildung vermeiden. Dabei werden in der Regel in jedem Router separate Listen der empfangenen Multicasts gespeichert. Anhand dieser Listen kann ein bereits empfangenes Multicast-Paket identifiziert und bei erneutem Empfang gezielt verworfen werden. Es ist zu bemerken, dass diese Erweiterung des Flooding lediglich die Schleifenbildung kontrolliert, jedoch die anderen Nachteile völlig unangetastet lässt. So werden die Multicasts weiterhin analog zum Broadcast im gesamten Netz verteilt und Bandbreite sowie ggf. Kosten im WAN-Bereich verschwendet. Außerdem ist die Pflege einer Liste der empfangenen Multicasts nur ein theoretisches Konzept. In der Praxis würde diese Liste sehr schnell einen sehr großen Umfang erlangen und zu inakzeptablen Wartezeiten bei deren Abarbeitung und damit der Paketweiterleitung führen.

Steiner-Bäume

Um das Problem der Bandbreitenverschwendung zu lösen, wird bei der Übertragung von Multicasts in Netzen ein Multicast-Baum gebildet. Dieser Baum verbindet alle Router, deren Subnetze Teilnehmer einer Multicast-Gruppe beinhalten, ohne dabei Schleifen zu erzeugen. Es handelt sich somit gewissermaßen um die Verbindung mit den geringstmöglichen Kosten bzw. die geringste Anzahl von nötigen Kanten (unter Berücksichtigung von deren Kosten).

Die Bildung eines solchen Baums ist auch als Steiner-Problem bekannt [HAKA_71], weshalb man den Baum auch als Steiner-Baum bezeichnet. Abbildung 2-6 zeigt die Bildung eines Steiner-Baums zur Verbindung einer Multicast-Gruppe über mehrere Subnetze hinweg.

Dabei bilden die Router R4, R2, R7 und R5 einen Baum (wie in der rechten Abbildung gezeigt). Der Baum wird relativ trivial gebildet, indem zwischen allen Routern, in deren Subnetzen Teilnehmer der Multicast-Gruppe existieren, die kostengünstigste gemeinsame Verbindung ermittelt wird. Die Kosten der einzelnen Kanten werden auch als Metrik bezeichnet. In Abbildung 2-6 sind die Kosten als Zahl neben den Kanten angemerkt. Hohe Kosten können z. B. eine langsame Übertragungsstrecke oder eine teure Anbindung symbolisieren, die Kosten sparend verwendet werden soll. Für die Ermittlung einer kostengünstigen Strecke (als Ast des Baums) werden die Kosten der Teilstrecken addiert. Somit ergibt sich z. B. für die Strecke R4 nach R7 die Strecke über R3 mit dem Kostenfaktor 1 + 1 = 2. Die Verbindung von R4 nach R7 über R3 und R6 würde z. B. Kosten von 1 + 2 + 2 = 5 verursachen.

Eine Verbindung von R4 über R3, R6 und schließlich R5 hätte mit 1 + 2 + 3 + 3 = 9 die höchsten Kosten. Somit stellt die Verbindung direkt über R3 die günstigste Verbindung dar. Die Gesamtkosten des Baums in dem in Abbildung 2-6 gezeigten Beispiel betragen als Summe der im Baum enthaltenen Kanten 1 + 2 + 2 + 1 + 3 = 9.

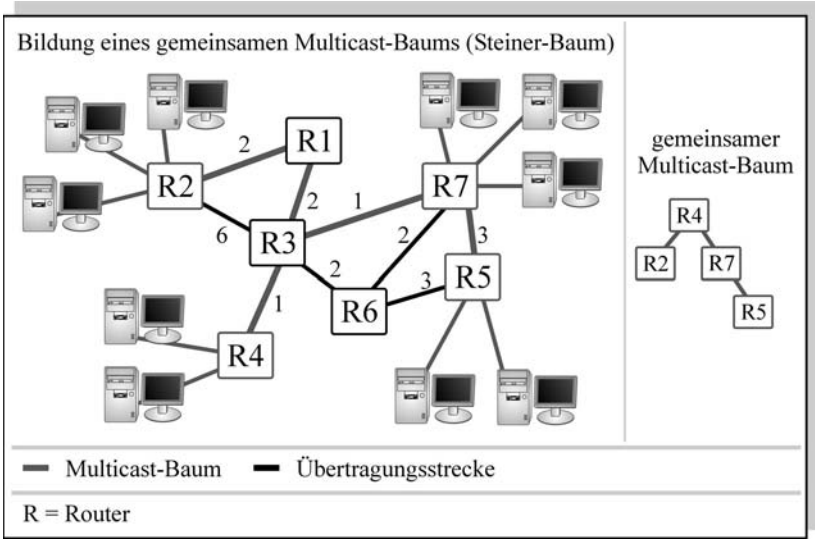

Abb. 2-6: Gemeinsamer Multicast-Baum als Steiner-Baum

Für die Ermittlung eines Steiner-Baums existieren unterschiedliche Approximationsalgorithmen, die z. B. in [WWW_80] beschrieben werden. Weitere Ausführungen zum Steiner-Baum-Problem finden sich in [KURO_02]. Trotz der existierenden Algorithmen und der relativ simplen

Ermittlung der Steiner-Bäume wird diese Art als Multicast-Baum im Internet von keinem Multicast-Routing-Protokoll verwendet. Stattdessen werden z. B. Bäume mit Rendezvous-Stelle verwendet, wie sie im nächsten Abschnitt beschrieben werden.

Bäume mit Rendezvous-Stelle

Die Abbildung 2-7 zeigt die Bildung eines Multicast-Baums über die Verwendung einer Rendezvous-Stelle. Der dabei verwendete Algorithmus ist ähnlich trivial wie der zur Bildung eines Steiner-Baums. In Abbildung 2-7 wurde angenommen, dass der Router R4 als Rendezvous-Stelle dient. Dies kann z. B. dadurch begründet sein, dass R4 den vorrangigen Sender (Quelle) der Multicast-Gruppe beinhaltet und von daher als erster in dieser Gruppe existiert. Der Router R4 wird auch als Kern bezeichnet, weshalb das Verfahren auch den Namen Kernbasiertes Routing trägt.

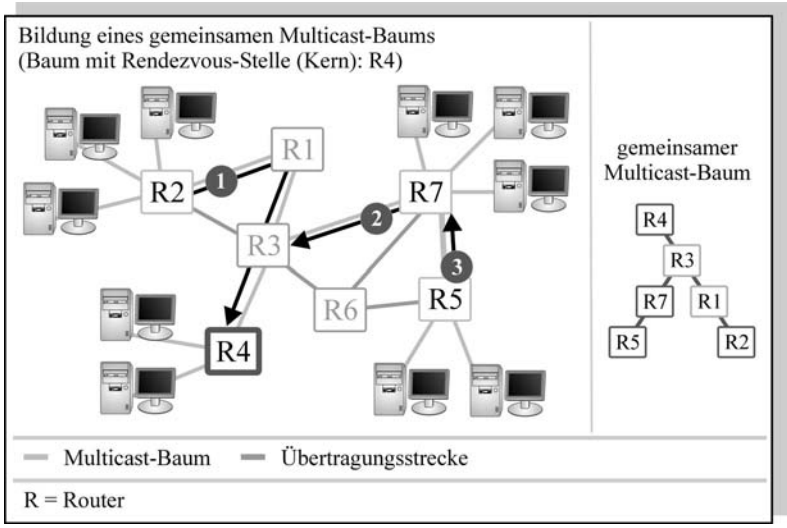

Abb. 2-7: Gemeinsamer Multicast Baum mit Rendezvous-Stelle (Kern)

Bei der in Abbildung 2-7 gezeigten Bildung eines Multicast-Baums mit Rendezvous-Stelle können folgende Schritte unterschieden werden:

1. Nachdem R4 zum Kern (Rendezvous-Stelle) des Multicast-Baums erklärt wurde, möchte zuerst R2 dem Baum beitreten. R2 schickt daher eine Anfrage an R4. Da der Weg von R2 nach R4 über R1 und R3 führt, wird dem Baum an R4 der Ast R4 nach R3 nach R1 nach R2 hinzuge-

fügt. Dies bildet den rechten Ast des in Abbildung 2-7 rechts gezeigten Multicast-Baums.

2. Als nächstes möchte R7 der Multicast-Gruppe beitreten. Er sendet über R3 eine Anfrage an R4. Da R3 bereits am Multicast-Baum von R4 hängt, wird der Ast von R7 zu R3 direkt auf den Baum aufgepfropft.
3. Schließlich möchte auch R5 der Multicast-Gruppe beitreten. Er sendet die Anfrage direkt an R7. Da R7 bereits ein Knoten im Multicast-Baum mit der Rendezvous-Stelle R4 ist, wird der Ast zu R5 direkt am Knoten R7 auf den Baum aufgepfropft. Damit sind alle Router, die Teilnehmer der Multicast-Gruppe in ihren Subnetzen besitzen, Knoten des Baums geworden.

Auch ein Multicast-Baum mit Rendezvous-Stelle besitzt keine Schleifen. Er ist sehr einfach zu ermitteln und wird z. B. beim Multicast-Routing-Protokoll CBT (Core Based Trees) verwendet.

Eine weitere Lösung für die Erstellung eines Multicast-Baums, die in Multicast-Routing-Protokollen eingesetzt wird, ist das Quellenbasierte Routing, das im nächsten Abschnitt beschrieben wird. Eine ausführliche Beschreibung der Bäume mit Rendezvous-Stellen findet sich z. B. in [KURO_02].

Quellenbasiertes Routing

Die bisher beschriebenen Verfahren benutzen alle einen gemeinsamen Multicast-Baum. Das Quellenbasierte Routing geht hier einen anderen Weg, in dem es ausgehend von der Quelle die jeweils günstigste Route zum Empfänger als separate Route vermerkt. Der jeweilige Router, der der Multicast-Gruppe beitreten möchte, versendet daher eine Join-Nachricht (Beitritt) an den Multicast-Router, der das Subnetz der Quelle (des Senders) verwaltet. Dieser ermittelt dann rückwärts den kürzesten Pfad zum Empfänger. Bei der Ermittlung dieses Pfades wird ein sehr einfacher Algorithmus verwendet, der wie folgt beschrieben werden kann:

Ein Router, der ein Multicast-Paket empfängt, leitet dieses Paket genau dann an alle seine Ausgangsleitungen, außer der, auf der er es empfangen hat, weiter, wenn er das Paket auf der Schnittstelle empfangen hat, die aus seiner Sicht die kürzeste Verbindung zur Quelle darstellt.

Konkret bedeutet dies, dass, wie in Abbildung 2-8 gezeigt, z. B. der Router R3 das von R4 empfangene Paket weiterleitet, da dies auf der Schnittstelle eingetroffen ist, die R3 direkt mit R4 verbindet. Im Gegensatz dazu verwirft z. B. R2 die von R1 erhaltene Kopie des Originalpakets von R4, da die von

ihm aus kürzeste Route zu R4 nicht über R1, sondern über R3 verläuft, und das Paket somit auf der falschen Schnittstelle eingetroffen ist. Auf diese Art und Weise wird, wie in Abbildung 2-8 gezeigt, ein vollständiger Multicast-Baum erzeugt, der keine Schleifen enthält.

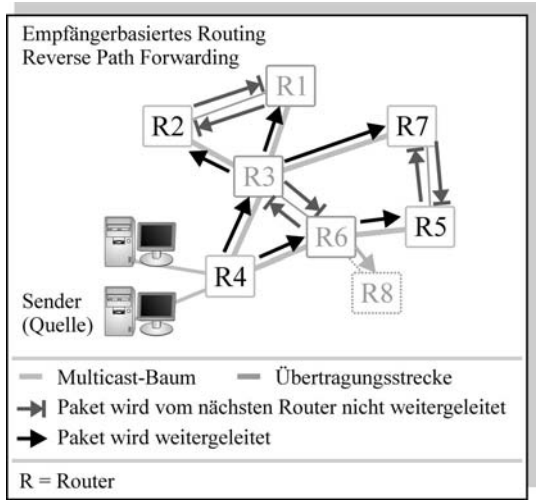

Abb. 2-8: Quellenbasiertes Routing: Reverse Path Forwarding

Dieses Verfahren wird auch als Reverse Path Forwarding (RPF) bezeichnet. Es gibt viele Erweiterungen dieses Verfahrens, wie z. B. das RPM (Reverse Path Multicasting), und einige weitere. In dieser Arbeit wird jedoch ausschließlich auf die gemeinsame Basis dieser Derivate, das RPF, eingegangen. Die Erweiterungen des RPF werden in [WIZI_99] ausführlich beschrieben.

Ein großer Nachteil dieses Verfahrens liegt in der Verteilung der Multicast-Pakete an R8. Da R8 Mitglied des in Abbildung 2-8 erstellten Multicast-Baums ist, jedoch selbst gar keine Teilnehmer der Multicast-Gruppe in seinem Subnetz unterhält, würde er unnötig mit Multicast-Paketen überschwemmt. Diese Tatsache ist besonders gravierend, wenn an R8 abwärts noch weitere Router hängen, die ebenfalls keine Teilnehmer der Gruppe unterhalten.

Um dieses Problem zu lösen, wurde beim quellenbasierten Routing das sog. Pruning (Beschneiden) eingeführt. Über eine Prune-Nachricht kann ein Knoten, der unerwünscht Multicast-Pakete einer Gruppe erhält, diese baumauf-

wärts abbestellen. Damit wird der Ast des Baumes beschnitten (engl. pruning). Abbildung 2-9 zeigt das Pruning des Multicasts-Baums von R8 aus. Angenommen der Router R5 sendet nun, wie in Abbildung 2-9 gezeigt, ebenfalls eine Prune-Nachricht an R6, da seine Multicast-Gruppen-Teilnehmer aus Abbildung 2-8 sich nun per IGMP mittels Leave Group abgemeldet haben, so sendet R6 diese Prune-Nachricht direkt an R4 weiter, da er nun ebenfalls keine Multicast-Pakete für diese Gruppe mehr weiterleiten muss.

In der Praxis wird ein Pruning nach einem vordefinierten Zeitraum vom Router, der die Prune-Nachricht erhalten hat, rückgängig gemacht. Dies ermöglicht, dass Teilnehmer, die in der Zwischenzeit bei den abgeschnittenen Routern zur Gruppe beigetreten sind, erneut Multicast-Pakete erhalten. Somit muss ein Router jeweils erneut eine Prune-Nachricht nach dem Verstreichen dieses Zeitraums versenden, sofern er nach wie vor keine Pakete dieser Multicast-Gruppe empfangen möchte. Dieses nachträgliche Aufheben eines Pruning wird auch als Grafting (Aufpfropfen) bezeichnet.

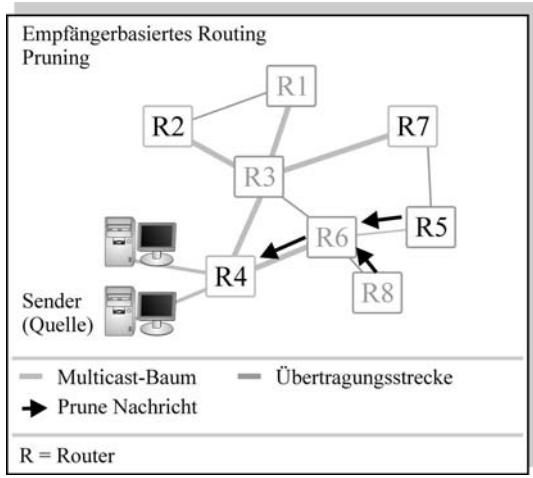

Abb. 2-9: Pruning (Beschneiden des Multicast-Baums)

Eine ausführlichere Beschreibung der genannten Verfahren lässt sich in [WIZI_99] sowie [KURO_02] nachlesen.

2.2.2 Protokolle

In IP-Netzen existieren in der Praxis mehrere Multicast-Routing-Protokolle. Dies führt unter anderem dazu, dass diese nicht untereinander kompatibel

sind. Als gemeinsame Schnittmenge aller Protokolle hat sich das DVMRP (Distance Vector Multicast Routing Protocol) herausgestellt, das weltweit die größte Akzeptanz findet.

Folgende Multicast-Routing-Protokolle werden in IP-Netzen verwendet:

DVMRP – Distance Vector Multicast Routing Protocol

Das DVMRP besitzt eine große Akzeptanz. Es wird unter anderem auch im MBone (dem Multicast Backbone on the Internet) verwendet. Die Spezifikation des DVMRP ist [RFC_1075]. Sie beinhaltet die Implementierung von quellenbasierten Multicast-Bäumen (nach RPF) inkl. Pruning und Grafting, wie in 2.2.1 beschrieben. DVMRP stützt sich auf das Unicast-Routing-Protokoll RIP (Routing Information Protocol). Es ermittelt anhand eines Distanzenvektors die jeweilige kürzeste Route eines Routers zur Quelle (Sender), was vom RPF-Algorithmus benötigt wird. Der große Nachteil von RIP ist, dass es nicht flexibel auf Engpässe auf dem Pfad zwischen Sender und Empfänger reagieren kann. So wird vom RIP in jedem Fall die Route gewählt, die die geringste Anzahl von Hops (Routern) enthält, ohne dabei z. B. langsame Verbindungsleitungen zu berücksichtigen. Dadurch können mitunter Routen ermittelt werden, die in puncto maximalem Durchsatz einer Route mit evtl. mehr Hops unterlegen sind. Eine ausführliche Beschreibung des DVMRP ist in [BADA_01] sowie [KURO_02] und [WIZI_99] enthalten.

MOSPF – Multicast Open Shortest Path First

Die beim DVMRP beschriebenen Nachteile des zugrunde liegenden Unicast Routing-Protokolls RIP werden beim OSPF (Open Shortest Path First) behoben, indem über Link-State-Algorithmen neben der reinen Anzahl von Hops als Metrik auch Kenngrößen wie maximale Verbindungsrate oder z. B. Verbindungskosten in die Ermittlung der günstigsten Router zwischen Sender und Empfänger einbezogen werden. Dadurch werden mittels OSPF in der Regel gerade in punkto maximaler Durchsatz effizientere Routen gefunden als beim RIP. Die Multicast-Variante des OSPF wird in [RFC_1584] beschrieben und MOSPF (Multicast Open Shortest Path First) genannt. Damit stellt das MOSPF grob gesagt eine Erweiterung des DVMRP um Link-State-Algorithmen dar.

CBT – Core Based Trees

Das bereits in Abschnitt 2.2.1 beschriebene Verfahren der Bildung von Multicast-Bäumen über Rendezvous-Stellen, auch Kernbasiertes Routing genannt, wird beim CBT (Core Based Trees) Multicast-Routing-Protokoll verwendet. CBT wird in [RFC_2201] sowie [RFC_2189] beschrieben. Dabei

sendet ein Router, der einer Multicast-Gruppe beitreten möchte, eine Nachricht JOIN_REQUEST an den Kern-Router (Redezvous-Stelle), der diese Nachricht mit einem JOIN_ACK beantwortet. Um den Baum aufrecht zu erhalten, werden in regelmäßigen Abständen ECHO_REQUEST-Nachrichten von den Downstream-Routern zu ihren unmittelbaren Upstream-Routern gesendet, die mit einer ECHO_REPLY-Nachricht beantwortet werden. Bleibt eine solche Antwort aus, wird der Baum downstream (also von dem Router aus abwärts) durch das Versenden der Nachricht FLUSH_TREE aufgelöst.

PIM – Protocol Independent Multicast

Während DVMRP und MOSPF von den jeweiligen Unicast-Routing-Protokollen RIP und OSPF abhängen, stützt sich das in [RFC_2362] beschriebene PIM (Protocol Independent Multicast) auf kein gesondertes Unicast-Routing-Protokoll. Ein Nachteil beider Verfahren (DVMRP sowie MOSPF) ist die verhältnismäßig hohe Belastung eines Netzes mit wenigen Multicast-Routern über einen großen Raum (ggf. mit einer großen Anzahl von Unicast-Routern zwischen den Multicast-Routern). Für diesen Fall werden beim PIM zwei verschiedene Modi unterschieden. Der Dense-Mode funktioniert analog zum DVMRP mit RPF, Pruning und Grafting in Netzwerken, die eine hohe Dichte von Multicast-Routern aufweisen. Für Netzwerke mit einer geringeren Dichte von Multicast-Routern (und ggf. einer hohen Dichte von Unicast-Routern) wird der sog. Sparse-Mode definiert. Im Sparse-Mode wird eine dem CBT ähnliche Technik mit Kernbasiertem Routing (Rendezvous-Stelle) verwendet. Dabei melden sich die einzigen Multicast-Router beim Kern-Router explizit an, was die Belastung der unterwegs liegenden Unicast-Router stark verringert.

3. Drahtlose Netzwerk-Technologien

Unter drahtlosen Netzwerk-Technologien versteht man im Allgemeinen die Übertragung von Informationen zwischen mehreren Endgeräten ohne die Verwendung von Kabeln. Alle derzeit verfügbaren Techniken in diesem Bereich berufen sich auf die Übertragung per Funk. Das Medium Kabel wird daher durch das Medium Luft ersetzt. Alle Verfahren machen sich den Effekt zu Nutze, den der britische Physiker James Clerk Maxwell bereits 1865 voraussagte und der schließlich 1887 vom deutschen Physiker Heinrich Hertz zuerst genutzt wurde. Sich bewegende Elektronen senden durch ihre Bewegung eine elektromagnetische Welle aus, die sich frei im Raum bewegen kann. Die ausgesendete Welle wird durch Materialien (wie Mauerwerk, Erdboden, Wasser (auch Regen!) usw.), aber auch von der Luft zum Teil absorbiert bzw. gedämpft. Diese Dämpfung bestimmt maßgeblich die Reichweite der (Funk-)Welle. Nur im Vakuum ist die Dämpfung einer elektromagnetischen Welle gleich null. Hohe Frequenzen werden stärker gedämpft als niedrige. Die elektromagnetische Welle kann nicht nur gedämpft, sondern auch an bestimmten Materialien (z. B. Metall) reflektiert oder gestreut werden [TANE_00].

Neben der reinen drahtlosen Bitübertragung, die per Funk möglich ist, sind jedoch weitere Verfahren notwendig, um von einem drahtlosen Netzwerk zu sprechen. Diese Verfahren beschreiben z. B. die Frequenznutzung, das Zugriffsverfahren und die Topologie und sichern damit die Interoperabilität zwischen verschiedenen Komponenten unterschiedlicher Hersteller. Sie sind in internationalen Standards definiert.

3.1 Strukturen und Eigenschaften

Alle drahtlosen Netzwerke besitzen unabhängig von den ihnen zu Grunde gelegten Standards ähnliche Strukturen und Eigenschaften. Neben der Übertragung per Funk anstelle von Kabeln decken die einzelnen Knoten in den Netzwerken immer die gleichen Funktionen ab.

3.1.1 Definition von drahtlosen Netzwerken

Im Zusammenhang mit der Beschreibung von drahtlosen Netzwerken in der vorliegenden Arbeit werden folgende Begriffe verwendet:

Station (node)

Eine Station stellt einen Knoten im Netzwerk dar. Sie besitzt daher in der Regel eine Adresse. Bei der Übertragung von Daten im Netz besitzt eine Station entweder die Rolle Sender oder Empfänger. Dies beschreibt eine unidirektionale Verbindung (der Sender sendet zum Empfänger). Bei Multicast sendet ein Sender an mehrere, bei Broadcast an alle Empfänger.

Im Netzwerk können Stationen folgende Funktionen haben:

- Endpunkt (client)

  Ein Endpunkt stellt eine Station dar, die mit genau einem Netzwerk verbunden ist. Ein Endpunkt kann Verbindungen zu mehreren Stationen unterhalten. Diese Stationen können jedoch nicht über den Endpunkt miteinander kommunizieren.

- Verteiler (access point)

  Eine Station, die eine Verteiler-Funktion übernimmt, vermittelt zwischen zwei Stationen. So kann z. B. eine Station eine andere Station außerhalb ihrer Reichweite über einen Verteiler erreichen. Eine gesonderte Form von Verteiler ist die **Brücke**, die mit mindestens zwei Netzwerken verbunden ist. Neben zwei drahtlosen Netzwerken, zwischen denen diese Brücke vermittelt, kann eines der beiden Netze auch ein drahtgebundenes Netz sein. Durch eine Brücke wird vielfach der Zugriff von drahtlosen Stationen auf drahtgebundene realisiert.

Übertragungsmedium

Durch die Nutzung des Übertragungsmediums können die einzelnen Stationen untereinander kommunizieren. Es stellt die gemeinsame Basis für ein drahtloses Netzwerk dar. Das Übertragungsmedium besitzt dabei zwei Eigenschaften:

- Übertragungscharakteristik

  Auf dem Weg zwischen Sender und Empfänger befinden sich häufig mehrere Hindernisse für die bei der Übertragung genutzten elektromagnetischen Wellen. Jedes Material dieser Hindernisse übt eine unterschiedliche (charakteristische) Dämpfung auf die Welle aus. Einzelne Materialien können die Welle sogar komplett absorbieren. Wieder andere Materialien reflektieren die Welle teilweise oder sogar komplett. Gerade dann, wenn der Sender onmidirektional sendet, lässt sich daher nur schwer beantworten, in welchen Bereichen der Empfang seines Signals möglich ist. Hinzu kommt, dass die Ausbreitung der Welle im Raum nahezu nicht deterministisch ist. In der Abbildung 3-1 haben beispielsweise die beiden mittleren Wellen eine unterschiedliche Laufzeit, obwohl beide den Empfänger erreichen und das gleiche Signal transportieren. Diesen Effekt bezeichnet man auch als Near-Far- oder Hidden-Station-Problem.

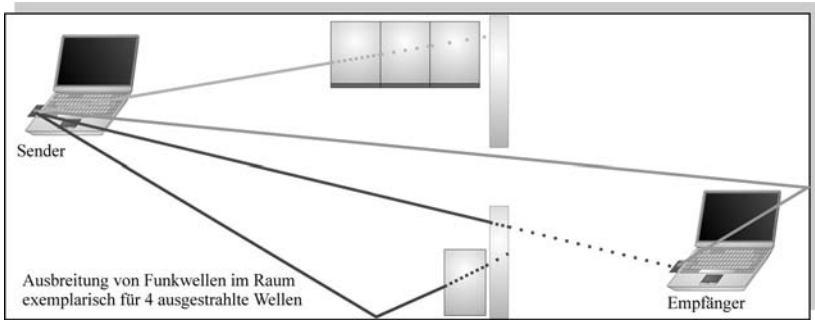

Abb. 3-1: Exemplarische Funkwellenausbreitung in geschlossenen Räumen

- Übertragungstechnik

  Die Nutzung des Übertragungsmediums kann mit verschiedenen Techniken erreicht werden. So können z. B. unterschiedliche Frequenzen genutzt werden oder ständig wechselnde Kanäle innerhalb der Frequenzen definiert werden. Diese Techniken dienen nicht der Bit-Übertragung, sondern der reinen Anpassung an die Gegebenheiten des Übertragungsmediums (Luft). Beim „spread spectrum“ z. B. wird ein Bit in eine Folge von mehreren Signalen per Funk aufgeteilt. Damit kann, auch wenn ein Großteil der Signale durch Dämpfung verloren gegangen ist, das übertragene Bit erkannt werden.

- Funkzelle

  Der gesamte Abdeckungsbereich, den ein Sender bedingt durch die Übertragungscharakteristik ausstrahlen kann, wird Funkzelle genannt. Andersherum stellt dies den Bereich dar, in dem ein Empfänger diesen Sender empfangen kann. Dabei ist die Funkzelle abhängig von Ihrer Übertragungstechnik. Eine Funkzelle ist immer an eine bestimmte Frequenz und damit an einen bestimmten Kanal gebunden. Diesen Kanal nennt man auch Kommunikationskanal.

Definition: Drahtlose Netzwerke

Ein drahtloses Netzwerk bezeichnet einen Verbund von mehreren Stationen, die nicht auf ein Kabel als gemeinsames Übertragungsmedium angewiesen sind.

3.1.2 Topologie

Die Kombination der in 3.1.1 beschriebenen Stationen führt zu den in Abbildung 3-2 gezeigten Topologien:

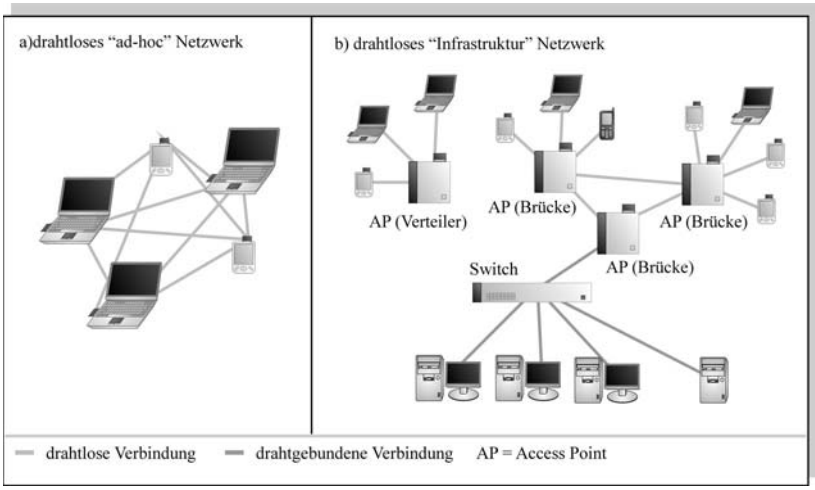

Abb. 3-2: Drahtlose Netzwerk-Topologien

- Drahtloses „Ad-hoc"-Netzwerk (Endpunkt-Netzwerk)

  Die einfachste Form eines drahtlosen Netzwerks stellt ein Netzwerk dar, das ausschließlich aus drahtlosen Endpunkten besteht. Da ein solches Netzwerk „Ad-hoc" entsteht, somit nicht gesondert installiert werden muss, spricht man hierbei auch von einem „Ad-hoc"-Netzwerk. Dabei

unterhält jede Station Verbindungen zu sämtlichen restlichen Stationen. Alle Stationen nutzen denselben Kanal zur Übertragung per Funk.

- Drahtloses Infrastruktur-Netzwerk (Verteiler / Access-Point-Netzwerk)

  In einem drahtlosen Netzwerk, das Access Points (kurz APs) für die Anbindung der Stationen verwendet, kommunizieren die Endpunkte über den Access Point (als Verteiler) miteinander. Bei der Verwendung von mehreren APs wird so eine Infrastruktur für das drahtlose Netzwerk geschaffen; man spricht daher auch von einem Infrastruktur-Netzwerk. Anders als beim „Ad-hoc"-Netzwerk können die einzelnen Stationen nicht direkt, sondern nur über den AP miteinander kommunizieren. Der Vorteil dieser Topologie ist auch, dass Stationen auch an Endpunkte senden können, die sich nicht in ihrem Sendebereich befinden. In einem Infrastruktur-Netzwerk arbeiten die APs außerdem nicht alle auf dem gleichen Kanal. Dadurch werden sowohl Störungen vermieden als auch das Roaming (automatischer Wechsel) zwischen zwei Funkzellen ermöglicht.

3.2 Standards

Für die Übertragung von Daten in drahtlosen Netzwerken existieren verschiedene Standards. Diese Standards sichern die Interoperabilität von Komponenten unterschiedlicher Hersteller und bilden den Rahmen für sämtliche zur Übertragung notwendigen Protokolle und Verfahren. In dieser Diplomarbeit wird vor allem auf die beiden Verfahren IEEE 802.11 (WLAN) [802.11] sowie Bluetooth [BTSIG] eingegangen. Diese beiden Verfahren haben eine sehr große Verbreitung und eine stetig ansteigende Akzeptanz. Während 802.11 dabei in LAN(Local Area Network)-, MAN(Metropolitan Area Network)- und teilweise sogar WAN(Wide Area Network)[TANE_00]- Verbindungen eingesetzt wird, ist für Bluetooth von dem IEEE der Bereich PAN (personal area network) klassifiziert. Damit bietet Bluetooth per Definition Verbindungen im Kurzstreckenbereich (ca. 10 Meter) an. Auch wenn für beide Verfahren Erweiterungen existieren, die andere Bereiche abdecken, wird in dieser Arbeit für Bluetooth implizit von PAN und bei 802.11 von LAN und MAN als Anwendungsbereich ausgegangen. Das WAN-Umfeld von 802.11 bleibt unberücksichtigt, da über große Distanzen in der Regel nur Punkt-zu-Punkt-Verbindungen eingesetzt werden, die weder Multicasts rechtfertigen würden noch für die Übertragung von Streaming-Media interessant sind. Die in Abschnitt 4.1.1 herausgestellten Eigenschaften von 802.11 in puncto Echtzeitfähigkeit können hierbei jedoch aufschlussreich sein.

3.2.1 IEEE 802.11 WLAN

Der WLAN(Wireless Local Area Network)-Standard 802.11 wurde vom IEEE (Institute of Electrical and Electronics Engineers) im Jahr 1997 definiert. Die IEEE-Protokollfamilie 802.x stellt die Grundlage für lokale Netze (LANs) dar. Insbesondere der Standard 802.3 definiert alle heute üblichen Ethernet-Standards. Die 802.x-Familie stellt dabei ein Rahmenwerk dar, dass sowohl den physikalischen Transport der Daten als auch Zugriffsverfahren und Netzwerkmanagement definiert. Dabei deckt die Familie die Schichten 1 und 2 (Physical Layer und Data Link Layer) des OSI-Referenzmodells ab. Genau diese beiden Schichten unterteilen auch die für die Übertragung von Streaming-Media relevanten Eigenschaften des Standards:

- Zugriffsverfahren (Data Link Layer)
- Übertragungstechnik (Physical Layer)

Zusätzlich wurden für 802.11 seit 1997 einige Erweiterungen definiert. In erster Linie stellen diese Erweiterungen die heute üblichen Standards 802.11a sowie 802.11b dar. Die Abbildung 3-3 zeigt, dass 802.11 für alle untergeordneten Standards einheitliche Zugriffsverfahren verwendet, während die Übertragungstechnik bei den einzelnen Standards unterschiedlich realisiert wird.

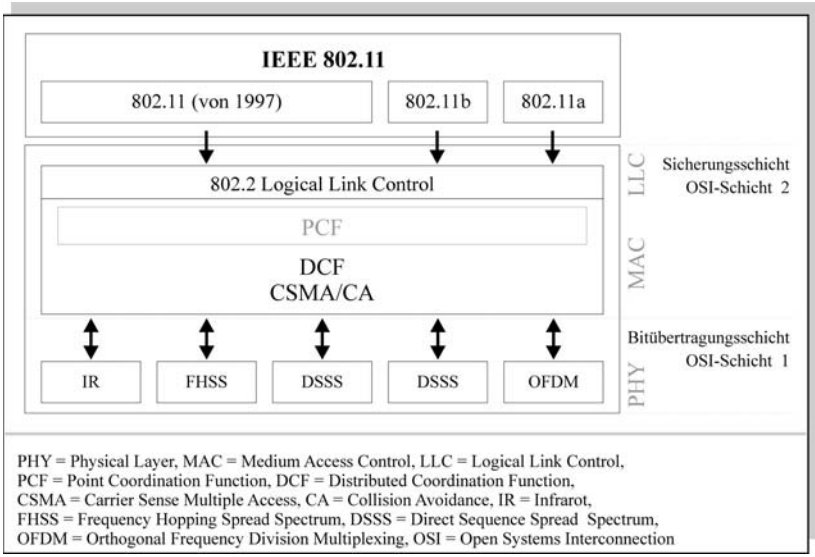

Abb. 3-3: Untergeordnete Standards von 802.11 und deren Übertragungs- und Zugriffsverfahren

Da der ursprüngliche 802.11-Standard von 1997 heute nicht länger in Verwendung ist, wird im folgenden Teil des Kapitels nur auf die beiden Standards 802.11b und dem seit kurzem auch in Deutschland zugelassenen 802.11a eingegangen.

Zugriffsverfahren

Der 802.11-Standard basiert auf dem Zugriffsverfahren CSMA/CA. Dabei steht CSMA für Carrier Sense Multiple Access. Unter Carrier Sense versteht man das Abhören des Kommunikationskanals durch alle Teilnehmer. Das heißt, dass jede Station entweder anhand von Feldstärke-Messungen oder im Rahmen der im folgenden beschrieben Verfahren PCF und RTS/CTS ermittelt, ob der Kommunikationskanal frei ist bzw. keine andere Station gerade über ihn sendet. Multiple Access bedeutet, dass alle Netzteilnehmer den gleichen Kommunikationskanal verwenden. CA steht für Collision Avoidance.

Anders als im drahtgebundenen Ethernet CSMA/CD (wobei CD = Collision Detection) kann eine Kollision auf dem Kommunikationskanal nicht erkannt werden, da sich hierbei die beiden ausgesendeten Funkwellen überlagern. Hinzu kommt, dass die Kollisionen durch die in Abbildung 3-1 gezeigten Effekte wie Laufzeitunterschiede und Dämpfung an einem einzigen Ort nicht erkennbar sind. Die hinter Collision Avoidance stehenden Verfahren bemühen sich insofern nur, die Anzahl von Kollisionen auf dem Kommunikationskanal möglichst gering zu halten. Kollisionen sind genau wie im drahtgebundenen Ethernet eine völlig normale Erscheinung im Netzwerkbetrieb.

Im einfachsten Fall wird für die Verwaltung des Zugriffs auf ein 802.11-WLAN die DCF (Distributed Coordination Function) verwendet. Die DCF setzt in einfacher Art und Weise die bereits beim CSMA/CA beschriebenen Anforderungen um. Dabei entsteht zwischen den einzelnen Stationen eine Art Wettbewerb (Contention Period) um den Zugriff zum Kanal. Abbildung 3-4 zeigt einen klassischen Zugriffsablauf zwischen drei konkurrierenden Stationen nach der DCF.

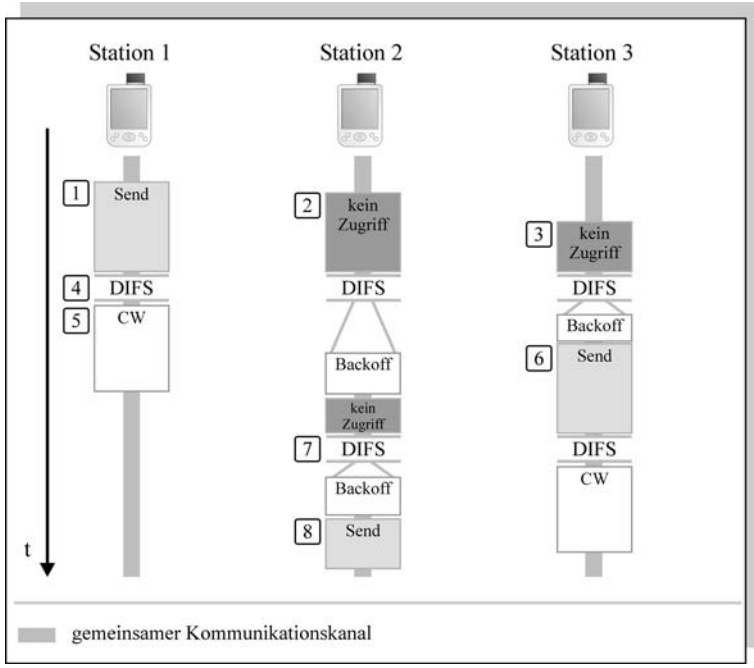

Abb. 3-4: Ablauf beim Zugriffsverfahren nach DCF

Der in Abbildung 3-4 gezeigte Ablauf umfasst folgende Schritte:

1. Zunächst sendet Station 1 auf dem gemeinsam genutzten Kommunikationskanal an eine hier nicht aufgeführte Station.
2. Station 2 möchte auf den Kommunikationskanal zugreifen, erkennt jedoch, dass dieser belegt ist und bricht den Zugriff ab. In der Praxis kann Station 2 aus dem Feld „Dauer“ des von Station 1 gesendeten Frames die Dauer der Kanalbelegung ermitteln. Daher wird der in der Abbildung salopp „kein Zugriff“ genannte Fall in der Realität NAV genannt. Der NAV (Net Allocation Vector) beinhaltet einen Wert, der den Zeitraum definiert, in dem die Station nicht auf den Kanal zugreifen kann.
3. Station 3 möchte ebenfalls auf den Kanal zugreifen und erkennt genau wie Station 2, dass der Zugriff momentan nicht möglich ist.
4. Die Übertragung von Station 1 ist beendet und alle aktiven Stationen warten eine DIFS-Zeit ab. DIFS steht für DCF IFS (InterFrame Space).

Bei 802.11 werden verschiedene IFS definiert, die alle Pausen von einer bestimmten Länge auf dem Kommunikationskanal darstellen.

5. Nachdem Ablauf der DIFS-Zeit startet Station 1 ein CW (Contention Window). Dieses Zeitfenster ermöglicht den anderen Stationen, sich um den Kanal zu bewerben (contention). Station 1 kann erst nach dem Ablauf des CW wieder einen Zugriffsversuch auf den Kanal ausüben. Da Station 1 jedoch keine weiteren Daten zu senden hat, endet ihre aktive Rolle in diesem Beispiel nach dem Ablauf des CW.

   Die anderen Stationen (2 und 3) gehen nach Ablauf der DIFS-Zeit in den Backoff-Modus. Im Backoff-Modus wird eine Zufallszahl ermittelt, die kleiner ist als der Zeitraum eines CW und diese Zahl schließlich kontinuierlich dekrementiert. Damit ergibt sich beim Backoff ein zufälliger Zeitraum, der auf jeden Fall kürzer ist als ein CW und damit einer anderen Station als der zuvor aktiven den Zugriff auf den Kanal ermöglicht.

6. Der Backoff-Prozess von Station 3 wurde (bedingt durch eine kleinere Zufallszahl) früher beendet als der Prozess von Station 2. Somit erkennt Station 2 den Kanal als frei und startet mit dem Sendevorgang.

7. Nach der Beendigung des Sendevorgangs von Station 3 warten wiederum alle aktiven Stationen einen DIFS-Zeitraum ab. Station 1 ist nicht mehr aktiv und beachtet diesen daher nicht.

8. Nachdem der Backoff-Prozess von Station 2 beendet ist, kann diese nun auch ihren Sendevorgang beginnen.

Bei dem dargestellten Vorgang wurden Bestätigungen außer Acht gelassen. Grundsätzlich wird eine Bestätigung (ACK) für den Empfang beim DCF vom Empfänger direkt an den Sender geschickt. Jedoch wird nach dem Empfang und vor dem Absenden des ACK ein SIFS(Short IFS)-Zeitraum abgewartet. Durch die verschiedenen Längen der IFS wird aufgrund der geringeren Wartezeit eine Art Priorität geschaffen. Somit haben ACK-Pakete eine höhere Priorität beim Versenden als Datenpakete, bei denen vor dem Senden erst ein kompletter DIFS-Zeitraum abgewartet werden muss.

Die DCF beinhaltet mehrere Probleme. Ein zentrales Problem entsteht, wenn zwei Stationen nach einem DIFS die gleiche zufällige Zahl für den Backoff-Prozess auswählen. In diesem Fall beginnen beide Stationen gleichzeitig zu senden und eine Kollision entsteht. Die DCF kann nur dann einwandfrei funktionieren, wenn alle Stationen sich gegenseitig erreichen (bzw. abhören) können. Daher entsteht ein weiteres Problem, wenn z. B. Station 1 und 2 Station 3 sehen, sie sich aber nicht untereinander erreichen können. Ein solches Problem entsteht, wenn sich die einzelnen Funkzellen der Sender

nicht alle überlappen. Eine genaue Beschreibung der DCF bietet [NMG_01] sowie [SIKO_01].

Eine optionale Erweiterung zur DCF, die unter anderem dieses Problem beseitigt, stellt der RTS/CTS(Ready To Send / Clear To Send)-Mechanismus dar. Dabei wird nach dem Ablauf des Backoff-Prozesses zunächst ein sehr kurzes RTS-Frame an den Empfänger geschickt und dies, unterbrochen durch ein SIFS, mit einem CTS vom Empfänger an den Sender quittiert, sofern dieser in der Lage ist, mit dem Empfang zu beginnen. Vor der eigentlichen Übertragung wird dann zusätzlich ein SIFS abgewartet. Die anschließende Quittierung (Bestätigung) der empfangenen Daten erfolgt analog zum ACK der DCF. Durch die sehr kurzen RTS- und CTS-Frames, die zwischen den Kommunikationspartnern ausgetauscht werden, und deren Bedingung für den Beginn einer Übertragung werden Kollisionen in diesem Fall effektiv vermieden. Sogar, wenn nicht alle Stationen in unmittelbarem Funkkontakt zu einander stehen.

Über den Nachteil des RTS/CTS-Verfahrens können jedoch auch die relativ kleinen Frames nicht hinwegtäuschen; es wird mehr Protokollverkehr (overhead) auf dem Kanal erzeugt. Damit geht effektiv Bandbreite verloren. Um diesen Verlust zu minimieren wird beim RTS/CTS-Verfahren ein Grenzwert (threshold) für die Größe der Pakete definiert, ab dem das RTS/CTS-Verfahren genutzt werden soll. Somit können kleine Pakete, die einen besonders hohen Protokollverkehr zur Folge hätten, ohne den RTS/CTS-Mechanismus übertragen werden, ohne dass Datenübertragungen, die meist in großen Paketen transportiert werden, gestört werden. Für mehr Informationen zum RTS/CTS-Mechanismus sei [NMG_01] sowie [SIKO_01] empfohlen.

Nicht zuletzt durch die zufälligen Backoff-Zeiten ist das DCF trotz seiner großen Verbreitung nicht besonders gut für Streaming-Media geeignet. Die undeterministische Wartezeit vor dem Senden der Daten führt zu großen Laufzeitunterschieden (Jitter) bei der Datenübertragung. Dieser störende Effekt kommt besonders bei der Echtzeit-Übertragung zum Tragen. Für Echtzeit-Übertragungen definiert der 802.11-Standard daher die PCF (Point Coordination Function). Die PCF eliminiert vorrangig die eingangs geschilderten störenden Backoff-Zeiten. Dabei wird die Übertragung auf dem Kanal kontinuierlich in zwei Perioden aufgeteilt. Die CFP (Contention Free Period) bietet für sendewillige Stationen ein Zeitfenster, indem sie priorisiert und ohne Wettbewerb mit nicht priorisierten Stationen ihre Daten senden dürfen. Die anschließende CP (Contention Period) erlaubt schließlich wieder den Wettbewerb um die Zuteilung des Kanals und nutzt die DCF wie in Abbildung 3-4 beschrieben.

PCF basiert somit auch auf dem CSMA/CA-Verfahren und der DCF. Es teilt allerdings priorisierten Stationen exklusive Übertragungszeit zu. Dabei lassen sich bei der Verwendung der PCF auch Stationen einsetzen, die diese nicht unterstützen. Dies wird dadurch erreicht, dass am Anfang der CFP alle Stationen ihren NAV (bekannt aus Schritt 2 der Abbildung 3-4) auf die maximale Zeit der CFP-Periode setzen. Diese Zeit bekommen die Stationen vom PC (Point Coordinatior) zugewiesen, der in der Regel ein AP ist. Der PC regelt daher zentral den Zugriff auf den Kanal während der CFP. Abbildung 3-5 zeigt einen exemplarischen Ablauf der PCF.

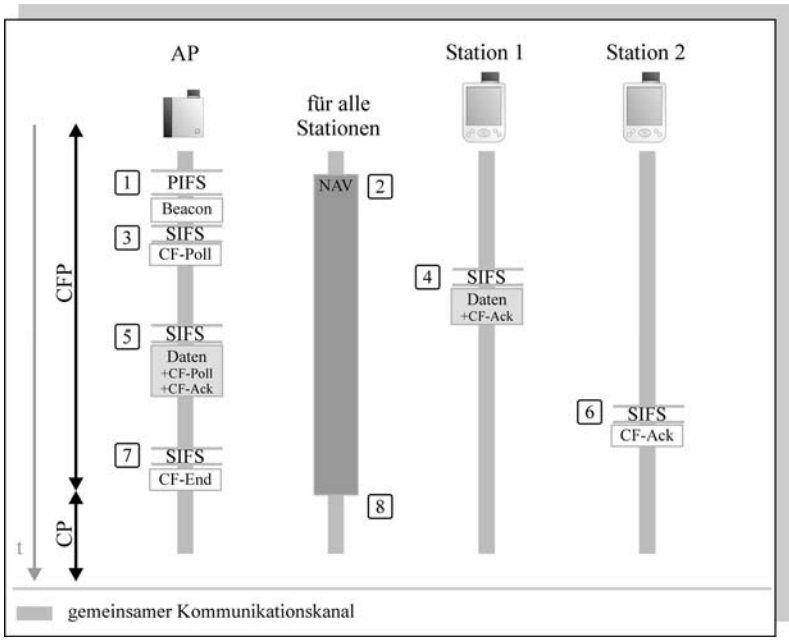

Abb. 3-5: Zugriff auf den Kommunikationskanal nach der PCF

Der in Abbildung 3-5 gezeigte Ablauf gliedert sich in folgende Schritte:

1. Der AP als PC wartet ein PIFS-Intervall ab und sendet dann ein Beacon. PIFS steht für PCF IFS und seine Dauer ist länger als die eines SIFS und kürzer als die Dauer eines DIFS. Damit hat das Beacon Vorrang vor sämtlichen unpriorisierten Übertragungen, die alle vor dem Senden ein DIFS abwarten müssen. Das Beacon wird vom AP im Infrastruktur-Netzwerk benutzt, um den erreichbaren Stationen Informationen über die Funkzelle zu geben, wie z. B. Uhrzeit, Kanal usw. Für die PCF sind dabei besonders zwei Werte interessant; zum einen, dass der AP und damit die

Funkzelle grundsätzlich PCF unterstützt (CF Parameter Feld) und zum anderen das Feld Beacon-Intervall. In diesem Feld steht die eingeplante Zeit bis zum nächsten Beacon. Daraus können die angebundenen PCF-fähigen(CF-Pollable)-Stationen erkennen, wann die nächste CFP beginnt.

2. Alle aktiven Stationen setzen beim Zugriff auf das Netz ihren NAV auf die maximale Dauer der CFP. Dadurch wird der Zugriff ohne CFP auf den Kanal gesperrt.
3. Nach dem Beacon beginnt der AP damit, seine CF-Poll-Liste (bestehend aus allen PCF-fähigen Stationen) abzuarbeiten. Zunächst wartet er ein SIFS ab. Dann sendet er reihum an die CF-Pollable-Stationen einen CF-Poll. Zuerst sendet er in diesem Fall an die Station 1.
4. Station 1 möchte in der ihr angebotenen priorisierten Übertragungszeit Daten versenden. Daher wartet sie ein SIFS ab und sendet schließlich ihre Daten mit CF-ACK (CF-Bestätigung) an den AP (bzw. PC).
5. Nachdem ein SIFS abgelaufen ist, beginnt der AP seinerseits die Daten zu verarbeiten. Er erkennt, dass die von Station 1 gesendeten Daten an Station 2 geschickt werden sollen. Daher schickt er die Daten sofort mit dem folgenden CF-Poll an die Station 2. Dabei sendet er ein CF-ACK mit, das von Station 1 abgehört und als ACK für seine Übertragung gewertet wird.
6. Station 2 sendet nach einem SIFS ein CF-ACK an den AP, um den korrekten Empfang der Daten zu quittieren.
7. Da in diesem Fall keine weiteren Stationen gepollt werden müssen, stoppt der AP den Durchlauf seiner CF-Poll-Liste und beendet nach einem SIFS mit CF-End die CF-Periode (CFP). Der AP kann in der CFP auch länger benötigen, in diesem Fall wird der NAV der Stationen verlängert. Grundsätzlich kann der AP selbst bestimmen, wann er von der CFP zur CP übergeht.
8. Der NAV-Timer der Stationen endet und sie können frei auf das Medium zugreifen. Sendewillige Stationen können nun einen BackOff-Prozess mit anschließendem DIFS starten und im Wettbewerb mit den anderen Stationen Daten in der CP analog zur DCF übertragen.

Die PCF-Implementierung ist im 802.11-Standard relativ offen gehalten. Die Hersteller können beispielsweise selbst entscheiden, nach welchem Scheduler / Dispatcher-Prinzip sie die CF-Poll-Liste abarbeiten wollen. Durch die Vermeidung von Kollisionen ist dieses Verfahren daher zwar sehr effektiv, aber erfüllt nicht die Anforderungen an harte Echtzeit. Insbesondere

lassen sich diese Anforderungen jedoch allein wegen der Fehlerrate bei der Funkübertragung nicht einhalten. Es ist ebenfalls nicht definiert, ob und wann ein verlorengegangenes Paket erneut geschickt oder verworfen werden soll. Das RTS/CTS-Verfahren ist bei der PCF nicht nötig, da die Kollisionen hier durch die Vergabe von Zeitschlitzen (Slots) vermieden werden.

Übertragungstechnik

Während die oben dargestellten Zugriffsverfahren für alle derzeit verfügbaren 802.11-Netzwerke gleich sind, unterscheidet sich die bei den einzelnen Standards benutzte Übertragungstechnik. Wie in Abbildung 3-3 bereits gezeigt wurde, definiert der 802.11-Standard drei konkrete Netzwerkstandards, wobei 802.11b und 802.11a derzeit am weitesten verbreitet sind und der ursprüngliche 802.11-Standard weitgehend nicht mehr verwendet wird. Auch 802.11b und 802.11a unterscheiden sich in ihrer Übertragungstechnik.

- Frequenz

  802.11b nutzt für die Datenübertragung einen Frequenzbereich aus dem ISM(Industrial Scientific Medical)-Band, der den Bereich von 2,4 - 2,4835 GHz abdeckt. Dieser Frequenzbereich kann weltweit weitgehend lizenzfrei genutzt werden. Das ISM-Band wird von einer Fülle von anderen Anwendungsbereichen genutzt. So funken z. B. Video-Überwachungsanlagen im gleichen Band. Die Übertragungstechnik in diesem Band muss daher besonders den störenden Einfluss von anderen Sendern vermindern, auch wenn diese eventuell mit einer ganz anderen Übertragungstechnik arbeiten.

  802.11a wird im 5-GHz-Band betrieben. In Europa steht dafür ein 455 MHz großer Frequenzbereich (5,15 – 5,35 GHz sowie 5,47 – 5,725 GHz), in den USA und Japan stehen mit 300 und 100 MHz jeweils sehr viel kleinere Frequenzbereiche zur Verfügung. Alle Kontinente haben jedoch einen gemeinsam abgedeckten Frequenzbereich von 5,15 – 5,25 GHz. Das 5-GHz-Band ist weitaus weniger ausgelastet als das 2,4-GHz-ISM. Neben WLANs nutzen derzeit kaum andere Anwendungsbereiche dieses Frequenzband.

- Kanalbelegung und Kodierungsverfahren bei 802.11b

  802.11b definiert in dem Frequenzbereich von 2,4 – 2,4835 GHz maximal 14 Kanäle (zwischen 2,412 GHz und 2,484 GHz) mit einem Kanalabstand von 5 MHz. Von diesen 14 Kanälen werden in den USA die ersten 11, in Europa die ersten 13 und in Japan allein der Kanal 14 genutzt. Dabei kommt das DSSS(Direct Sequence Spread Spectrum)-Verfahren für die Kodierung und Übertragung der Bits zum Einsatz.

Spread Spectrum bezieht sich dabei auf die Aufspreizung eines einzelnen Signals in eine größere Bandbreite. Beim DSSS wird (für Datenraten von 1 und 2 MBit/s) ein Bit auf 11 Chips (Barker-Code) aufgespreizt. Diese Chips bilden dann das tatsächlich übertragene Signal. Die Chip-Folge ist im 802.11 definiert als 10110111000, dem so genannten PN-Code (Pseudo Noise). Dieser fest definierte Code ist sowohl dem Sender als auch dem Empfänger bekannt und dient damit zur Filterung von 802.11-Übertragungsdaten aus dem ISM-Band.

Anhand der Spreizung des Signals ist das übertragene Bit weniger störungsanfällig. Außerdem verliert das Signal bedingt durch die Spreizung sehr viel von seiner Intensität. Soviel, dass das übertragene Signal unterhalb der Rauschgrenze liegt. Für die anderen Teilnehmer im ISM-Band ist es somit kaum empfangbar, wodurch Störungen durch 802.11 vermieden werden. Diese Vorteile führen allerdings auch zu einigen Nachteilen. Die für die Übertragung eingesetzten Komponenten müssen beispielsweise sehr hohe Bandbreiten (in diesem Fall 11-mal größer als die eigentliche Bitrate) unterstützen. Wie die Kodierung des Signals beim 802.11 genau funktioniert, zeigt die Abbildung 3-6.

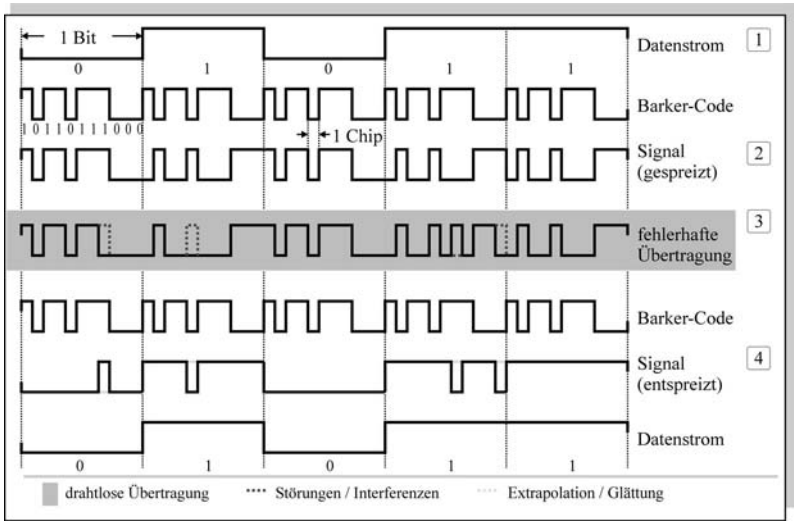

Abb. 3-6: Übertragung einer Bitfolge beim DSSS

Die in Abbildung 3-6 gezeigten Abläufe beim DSSS zwischen Sender und Empfänger lassen sich in folgende Schritte einteilen.

1. Der zu übertragende Datenstrom wird zunächst über eine exklusive Oder-Operation (Modulo-2-Addition) mit dem Barker-Code verknüpft. Dadurch entsteht das ausgehende (gespreizte) Signal.

2. Für die eigentliche Übertragung des gespreizten Signals wird eine DQPSK(Differential Quadrature Phase Shift Keying)-Modulation benutzt. Die DQPSK definiert unterschiedliche Symbole, die für eine bestimmte Phasenverschiebung stehen und Bitfolgen symbolisieren. Dabei wird ein Symbol direkt als Phasenverschiebung übertragen.

   Bei einer Datenrate von 1 MBit/s wird ein 11-Chip-Barker-Code benutzt und per DBPSK (Differential Binary PSK) auf die Welle moduliert. Die Binary-PSK definiert genau zwei Phasen jeweils für 0 (Phasenverschiebung 0) und 1 (Phasenverschiebung ?) als Eingabebitfolge (ein Symbol entspricht genau einem Bit). Bei 2 MBit/s wird die DQPSK mit vier Phasen (0, ?/2, ? und 3 ?/2) für jeweils zwei Bits als Eingabefolge verwendet. Dabei wird das Ausgangssignal durch die Überlagerung der Phasenverschiebung (Q-Kanal – Quadrature-Phase-Channel) und der ursprünglichen Wellenform (I-Kanal – In-Phase-Channel) erzeugt.

3. Im Bild 3-6 wird das übertragene Signal unterwegs gestört bzw. es treten Interferenzen mit anderen Signalen auf. Die „gestrichelten“ Bereiche der Flanken zeigen die verfälschten Bereiche des Signals.

4. Mit Hilfe einer erneuten exklusiven Oder-Operation auf dem empfangenen Signal und dem Barker-Code lässt sich für den Sender die ursprüngliche Bitfolge zurückgewinnen. Dabei können durch Glättungsverfahren auch bei einer Störung von etwas weniger als der Hälfte der das Bit kodierenden Chips noch korrekte Ausgangsdatenströme erzeugt werden.

Für höhere Bitraten von 5,5 oder 11 MBit/s wird ein komplizierteres Verfahren verwendet, das statt 11-Chip-Barker-Codes 8-Chip-Complementary-Codes verwendet. Beim CCK (Complementary Codes Keying) werden die eingehenden Daten gesplittet und auf 4 (bei 11 MBit/s 64) komplexe Codewörter verteilt. Deren Real- und Imaginärteil wird schließlich für den I- und Q-Kanal der DQPSK als Eingang verwendet. Die relative komplizierte CCK-Kodierung wurde für die höheren Bitraten gewählt, da in diesem Fall die gleichen Frequenzen und Datenformate wie bei der 1- und 2-MBit/s-Übertragung genutzt werden können. Außerdem kann die Übertragungshardware weitergenutzt werden, da die Anzahl von Chips pro Sekunde mit 11 MChips/s gleich bleibt. Hohe Geschwindigkeiten führen allerdings zu einer deutlichen Abnahme der Reichweite, da sie weitaus störungs- und damit auch dämpfungsanfälliger sind als die Übertragungsverfahren für geringere Bitraten.

Durch die Spreizung der übertragenen Signale ergibt sich ein Problem. Die sehr breitbandigen Signale lassen sich nicht in der ursprünglich definierten Kanalbreite von 5 MHz übertragen. Für einen 802.11-Kanal ergibt sich durch maximal 11 Chips eine Kanalbreite von 22 MHz. Durch diesen Effekt werden die anfangs erwähnten 13 Kanäle in Europa in ihrer Nutzung stark eingeschränkt. Damit sich die Kanäle nicht untereinander stören, können nur die Kanäle 1,6 und 11 (mit einem Bandabstand von 3 MHz) betrieben werden, da diese sich nicht überlappen. Die einzelnen Kanäle können per FDMA (Frequency Division Multiple Access) durch eine Station am WLAN differenziert werden.

Eine ausführliche Beschreibung der Kanalbelegung von 802.11b bieten [NMG_01] sowie [SIKO_01].

- Kanalbelegung und Kodierungsverfahren bei 802.11a

  Das 5-GHz-Frequenzband ist sehr viel weniger dicht benutzt als beispielsweise das ISM-Band. Daher kann das Kodierungsverfahren in diesem Fall sowohl auf eine große Bandbreite zurückgreifen als auch die Vermeidung von Interferenzen mit anderen Übertragungsverfahren weitgehend ignoriert werden. Dadurch lassen sich wesentlich effizientere Übertragungsverfahren nutzen, die Raten von bis zu 54 MBit/s bieten. Das grundlegende Verfahren zur Erhöhung der Übertragungsrate stellt dabei FDM (Frequency Division Multiplexing), welches die Grundlage für das einheitlich im 5-GHz-Bereich verwendete Übertragungsverfahren OFDM (Orthogonal FDM) bildet. Beim FDM wird ein Signal auf mehreren Unterkanälen parallel übertragen. Es werden somit mehrere Trägerfrequenzen verwendet, auf denen ein Teil des Gesamtsignals übertragen wird. n Unterkanäle bedeuten somit auch eine n-fache Erhöhung der Übertragungsrate. Grundsätzlich könnte man einfach mehrere separate Frequenzbereiche nebeneinander einsetzen. Dabei erweist sich dieses Verfahren jedoch, analog zum DSSS, wo nur 3 Kanäle effektiv nutzbar bleiben, als recht verschwenderisch.

  Beim OFDM wird dieser Nachteil behoben, indem sich die Frequenzbereiche überlappen dürfen und damit 50 % der verfügbaren Bandbreite eingespart wird. Damit sich die verwendeten Unterkanäle in diesem Fall nicht untereinander stören, werden ihre Trägerfrequenzen orthogonal zueinander gewählt. Das heißt, dass ihre Signalrate T genau so gewählt wird, dass sie dem Abstand zwischen den Unterkanälen entspricht. Dadurch sind die jeweiligen Maxima der Unterfrequenzen jeweils um den Faktor 1/T verschoben. Durch die dabei vorliegende Orthogonalität fällt ein Maximum einer Unterfrequenz jeweils auf ein Minimum aller

anderen parallel übertragenen Frequenzen.

Abbildung 3-7 zeigt die Fourier-Transformierte einer OFDM-Übertragung.

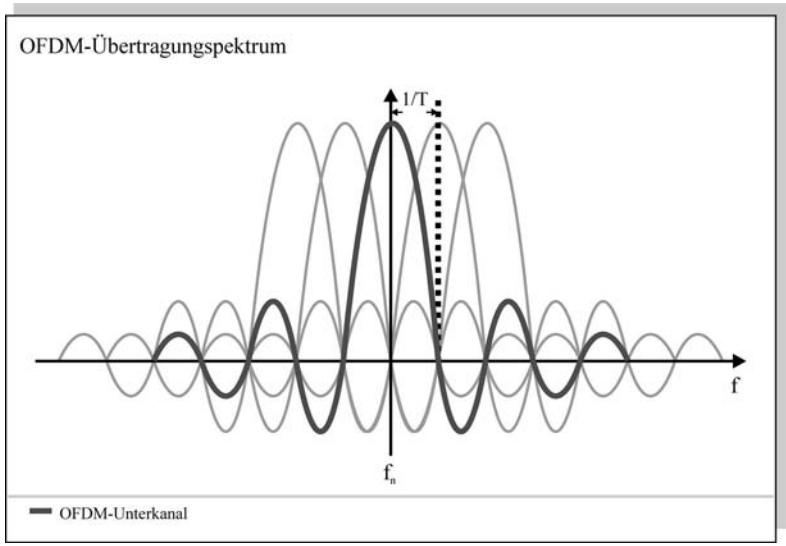

Abb. 3-7: Orthogonale Unterfrequenzen bei einer OFDM-Übertragung

Die Dekodierung des OFDM-Signals beim Empfänger kann durch eine FFT (Fast Fourier Transformation) bzw. IFFT (Inverse FFT) sehr einfach realisiert werden.

Für die eigentliche Übertragung werden die einzelnen Signale beim OFDM schließlich ähnlich wie beim DSSS auf die Unterfrequenzen per BPSK (Binary Phase Shift Keying) und QPSK (Quadrature Phase Shift Keying) moduliert, wobei beim QPSK mit seinen vier möglichen Phasen beim OFDM eine Übertragungsrate von maximal 18 MBit/s möglich ist (eine Steigerung um Faktor 9 im Vergleich zum DSSS ohne CCK). Um auf die maximale Übertragung von 54 MBit/s zu kommen, wird beim OFDM die QAM (Quadrature Amplitude Modulation) eingesetzt. Diese kommt auch in den anderen Anwendungsbreichen, die OFDM nutzen, wie z. B. DAB (Digital Audio Broadcast) oder ADSL (Asynchronous Digital Subscriber Line), zum Einsatz. Bei der QAM werden grundsätzlich wie bei der QPSK Phasenverschiebungen in unterschiedlichen Grö-

ßen eingesetzt, zusätzlich wird hier jedoch auch die Amplitude variiert. Dies bedeutet, dass die einzelnen Phasenverschiebungen eine unterschiedliche Amplitude erhalten können. Dadurch wird die Informationsmenge pro Zeiteinheit vervielfacht.

Ähnlich wie die CCK erreicht die QAM höhere Übertragungswerte nur auf Kosten der Reichweite. Übertragungen mit 54 MBit/s sind beim OFDM nur in einem Bereich von ca. 10 Metern möglich.

Eine detaillierte Beschreibung der Kanalbelegung bei 802.11a kann in [NMG_01] sowie [SIKO_01] nachgelesen werden.

	802.11b	**802.11a**
Frequenzbereich	2,4 GHz	5 GHz
Physical Layer	DSSS	OFDM
Zugriffsverfahren	CSMA/CA	CSMA/CA
Durchsatz (brutto - theoretisch)	11 MBit/s, 5,5 MBit/s, …	54 MBit/s
Durchsatz (netto – praktisch)	5 MBit/s	22 MBit/s
Reichweite	300 Meter (mit spez. Antenne auch mehr)	300 Meter (mit spez. Antenne auch mehr)
Verbreitung / Akzeptanz	groß, viele Implementierungen	wächst, sehr neuer Standard

Tabelle 3-1: Unterschiede der 802.11-Übertragungsverfahren

3.2.2 Bluetooth

Seitdem im Jahr 1998 die BSIG (Bluetooth Special Interest Group) gegründet wurde, um einen einheitlichen „Kurzstrecken“-Funkübertragungsstandard zu definieren, wächst die Akzeptanz für den Bluetooth-Standard stetig an. Das Ziel dieses Standards ist, zum einen auf Kabel für den Datenaustausch zwischen Endgeräten zu verzichten, und zum anderen eine möglichst günstige und kompakte Implementierung zu ermöglichen, die in praktisch jedem Gerät eingebaut werden kann. Geräte die mit einem Bluetooth-Chip ausgestattet sind, sollen schließlich in der Lage sein, unabhängig untereinander kommunizieren zu können. Seit der im Jahr 2001 erschienenen

Version 1.1 des Bluetooth-Standards ist seine Popularität derart gestiegen, dass auch die IEEE einen Standard für PANs (Personal Area Networks) in der Entwicklung hat. 802.15 soll die von Bluetooth definierten Zugriffs- und Übertragungstechniken in die 802-Familie integrieren. Dabei sieht die IEEE explizit die Ergänzung von WLANs nach 802.11 mit PANs nach 802.15 im Nahbereich vor. Die genaue Bluetooth-Spezifikation kann unter [BT] nachgelesen werden. Die IEEE-Definition von PANs nach 802.15 ist unter [802.15] zu finden.

Abbildung 3-8 zeigt die im Bluetooth-Standard definierten Komponenten. Für die Übertragung von Streaming-Media sind dabei die mit einer breiteren Linie verbundenen Komponenten relevant. Diese werden beim Bluetooth-LAP-Profile (LAN-Access-Point), welches Bluetooth-Endgeräten den Zugiff auf ein LAN ermöglicht, eingesetzt. Dabei beschreibt die RF-Spezifikation gewissermaßen die Bitübertragung und damit die Übertragungstechnik. Das Baseband beschreibt Zugriffsverfahren auf den Übertragungskanal. Als Schnittstelle zwischen Übertragungskanal und den eigentlichen Daten steht die L2CAP-Schicht.

Bluetooth definiert mehrere Profiles, die die unterschiedlichen Anwendungsmöglichkeiten beschreiben. Grundsätzlich würde auch die Komponente Audio und das zugehörige Profile für Streaming-Media in Frage kommen. Allerdings ist diese Anwendung zum einen seitens Bluetooth vorrangig für Telefonie (z. B. per Handy) vorgesehen, zum anderen handelt es sich bei einem einzelnen kontinuierlichen Medium wie Audio nicht um eine Multimedia-Anwendung und somit auch nicht um klassisches Streaming-Media.

Netzwerke zwischen Bluetooth Endgeräten bilden immer ein Infrastruktur-Netzwerk. Auch wenn augenscheinlich Bluetooth-Geräte wie z. B. Handy und PDA, sich untereinander „Ad-hoc“ verbinden können, verwaltet auch in dieser Konfiguration eines der beiden Geräte die Verbindung und stellt damit den Master dar. Das andere Endgerät wird in diesem Fall als Slave bezeichnet. Ein Master kann bis zu sieben Slaves verwalten, die aktiv kommunizieren können. Sind mehr als sieben Slaves an einen Master angebunden, so gehen die, die nicht aktiv sind, in den Zustand „geparkt“ über. In diesem Zustand können sie nicht aktiv an der Kommunikation im Netzwerk teilnehmen, wohl aber vom Master mit einem bestimmten Paket geweckt und damit aktiviert werden.

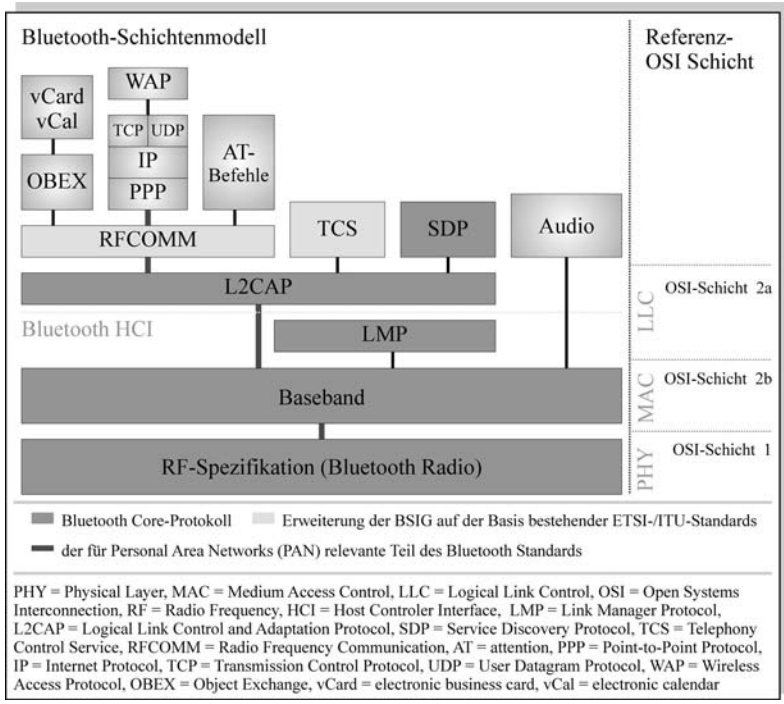

Abb. 3-8: Schichtenmodell des Bluetooth-Standards und Zuweisung der relevanten OSI-Schichten

Aus den Ausführungen ist erkennbar, dass in Bluetooth-Netzwerken maximal acht Endgeräte ein aktives Netzwerk bilden können. Ein solches Netzwerk nennt man ein Piconet, das seinerseits einen Piconet-Master und maximal sieben Slaves besitzt. Dabei kann ein Piconet-Master auch selbst Slave eines anderen Piconet werden. Dadurch entstehen vernetzte Piconets, die man auch insgesamt als Scatternet (verteiltes Netz) bezeichnet.

Ähnlich wie beim 802.11 sind auch hier für Streaming-Media vorrangig interessant:

- Zugriffsverfahren (Data Link Layer)
- Übertragungstechnik (Physical Layer)

Zugriffsverfahren

Anders als beim 802.11 mit PCF und DCF werden bei Bluetooth explizit zwei Verbindungsmodi unterschieden, die jeweils zeitkritische und zeitunkritische Übertragungen unterstützen.

Der SCO (Synchronous Connection-Oriented Link) bietet eine verbindungsorientierte Kommunikation für zeitkritische Übertragungen an. Für diese Verbindung werden vom Master Zeitschlitze (slots) festgelegt, in denen die Slaves per SCO übertragen dürfen. Anders als bei der PCF von 802.11 erfolgt die Zuteilung dieser Zeitschlitze strikt reihum und bietet damit eine quasi leitungsvermittelte Qualität der Verbindung (minimale Verzögerungen, minimaler Jitter). Genau dafür ist der SCO schließlich auch ursprünglich definiert, für Telefon- bzw. Sprach/Audio-Verbindungen über Bluetooth. Die von Bluetooth definierte maximale symmetrische Rate von synchronen Verbindungen entspricht für HV(High Quality Voice)-Pakete gerade 64 KBit/s, was der Bitrate von Telefonverbindungen im ISDN (Integrated Services Digital Network) entspricht.

Der ACL (Asynchronous Connectionless Link) wird im Gegensatz zum SCO nicht in fest zugeteilten Zeitschlitzen übertragen. Er bietet eine paketvermittelte Übertragung, die allerdings nicht für zeitkritische Kommunikation geeignet ist. Die Verzögerungen der Übertragung können in diesem Fall sehr groß werden und durch starke Schwankungen einen hohen Jitter erzeugen. Im ACL-Modus kann der Master auch an alle Slaves gleichzeitig senden, indem er keine Zieladresse im Paket angibt. Diese Versandart entspricht in etwa einem Broadcast beim Ethernet. In der Version 1.1 unterstützt Bluetooth jedoch kein Multicasting über mehrere Hops (über mehrere Master). Somit ist ein Multicast-Routing bei Bluetooth derzeit nicht möglich.

Der eingangs mit Zeitschlitzen beschriebene Kanalzugriff erfolgt bei Bluetooth nach dem TDMA/TDD(Time Division Multiple Access / Time Division Duplex)-Verfahren. Dabei wird ein Kanal, der vorzugsweise eine große Bitrate bietet, nach dem Zeitmultiplexverfahren in mehrere feste Zeitschlitze unterteilt, die schließlich den einzelnen Stationen zugewiesen werden können. Wird bei diesem Verfahren bei der Einteilung der Zeitschlitze zusätzlich der Hin- und Rückkanal der einzelnen Stationen unterschieden, d. h. das Multiplexing in beide Richtungen auf einem Kanal unterstützt, so spricht man zusätzlich von TDD (Time Division Duplex). Nachteil des TDMA/TDD ist, dass es zentral verwaltet werden muss. Es kommt somit nur in Inrastruktur-Netzwerken in Frage, wo ein Access Point (Master) zentral die Zeitschlitze zuteilt. Das TDMA/TDD-Verfahren ist jedoch durch seine

einfache Zugriffskontrolle sehr einfach und damit billig zu realisieren. Eine Forderung, die Bluetooth implizit durch die Integration in alle erdenklichen Peripheriegeräte stellt.

Übertragungstechnik

Bluetooth arbeitet genau wie 802.11b im 2,4-GHz-ISM-Band. Dies führt zum einen zu einer noch höheren Auslastung dieses Bandes und zum anderen zu einer störenden Wechselwirkung zwischen beiden Technologien. In der Praxis kann sich die Fehlerrate bei einer 802.11b-Übertragung am Rande des Empfangsbereichs der jeweiligen Funkzelle erhöhen, sobald ein Bluetooth-Endgerät in der näheren Umgebung zu senden beginnt. Dieser Effekt ist jedoch abhängig von räumlichen Gegebenheiten und der Auslastung der Netzwerke.

Bei Bluetooth wird für die Kodierung des Signals auf dem Kanal FHSS (Frequency Hopping Spread Spectrum) genutzt. Diese auch im ursprünglichen 802.11-Standard verwendete Spread-Spectrum-Technik versucht eine störungsfreie Übertragung der Signale durch schnelles und regelmäßiges Wechseln der Frequenzen zu erreichen. Bei Bluetooth wird die Frequenz dabei mit 1600 Hops/s geändert. Die jeweils verwendeten Sprungfrequenzen werden anhand der Bluetooth-Adresse des Masters und einer Pseudozufallszahlenfolge ausgewählt. Dadurch ist die Sprungfolge allen Bluetooth-Endgeräten, die mit diesem Master verbunden sind, bekannt. In Europa und Nordamerika sind 79 mögliche Sprungfrequenzen definiert. In Frankreich, Spanien und Japan sind es 23.

Das vergleichsweise alte FHSS-Verfahren bietet für die Implementierung bei Bluetooth den entscheidenden Vorteil, dass es sehr einfach und kostengünstig realisierbar ist. Außerdem können beim FHSS bis zu 13 Stationen (bezogen auf die Sprünge über die verfügbaren Frequenzen) gleichzeitig übertragen, ohne das dabei zu viele Kollisionen entstehen. Dieser Wert reicht in einem Bluetooth-Netzwerk mit maximal sieben aktiven Slaves völlig aus. Nachteile von FHSS sind vor allem die maximale Geschwindigkeit von 2 MBit/s (bedingt durch die maximale Bandbreite im 2,4-GHz-Band) und eine schlechte Roaming-Unterstützung. Ersteres lässt sich nur durch andere Übertragungsverfahren wie etwa das von 802.11b bekannte DSSS beheben. Dies ist der Grund, warum zukünftige Bluetooth-Versionen mit höheren Bitraten (in der Entwicklung der IEEE befindet sich z. B. eine 20-MBit/s-Version) nicht auf dem FHSS basieren werden. Der zweite Nachteil ist daher bedingt, dass beim Roaming zwischen zwei Funkzellen zunächst alle Frequenzen nacheinander für einen bestimmten Zeitraum nach einem Beacon (ähnlich wie beim 802.11-Paket mit Status-Informationen der Funkzelle

sowie in diesem Fall Hopping-Sequenz) abgehört werden müssen. Dadurch dauert der eigentliche Roaming-Vorgang mitunter sehr lange.

	Bluetooth
Frequenzbereich	2,4 GHz
Physical Layer	FHSS
Zugriffsverfahren	TDMA/TDD
Durchsatz (brutto - theoretisch)	1 MBit/s (verschiedene feste Raten für Up- und Downstream)
Durchsatz (netto – praktisch)	ca. 300 KBit/s
Reichweite	10 Meter (100 Meter mit spez. Antenne)
Verbreitung / Akzeptanz	groß, viele Implementierungen
Maximale Clients pro Zelle	8 (inkl. Master)

Tabelle 3-2: Eigenschaften von Bluetooth

3.2.3 weitere Standards

Neben den bereits erwähnten Standards für drahtlose Netzwerke, WLAN und Bluetooth, existieren einige weitere, die mehr oder minder den gleichen Funktionsumfang bieten. Z. B. HIPERLAN, dass in der Version HIPERLAN/2 mit OFDM und Übertragungsraten bis zu 54 MBit/s direkt gegen 802.11a antritt. HIPERLAN/2 bietet außerdem eine fest definierte Unterstützung für QoS-Parameter. Dank TDMA/TDD arbeitet es verbindungsorientiert und würde gerade für die Übertragung von Streaming-Media gut geeignet sein. 802.11 ist jedoch der derzeit am weitesten verbreitete Standard und bis auf eine Handvoll Implementierungen existieren praktisch keine Systeme, die HIPERLAN unterstützen.

Ebenso steht es um HomeRF, dass ursprünglich einmal für die drahtlose Vernetzung im Heimbereich propagiert wurde. Auch HomeRF-Endgeräte sind in der Praxis kaum mehr vertreten. Ebenfalls im Heimbereich findet der DECT-Standard derzeit vor allem für schnurlose Telefone eine große Akzeptanz. Auch er bietet Möglichkeiten zur Datenübertragung. Der DPRS (DECT Pakket Radio Service) bietet Bandbreiten von bis zu 1 MBit/s und strebt

mögliche Raten von 20 MBit/s an. Doch auch dieser Standard findet in der Praxis kaum Akzeptanz.

Daher stützt sich die vorliegende Arbeit vor allem auf die Standards 802.11 und Bluetooth. Weitere Standards für drahtlose Netze findet der interessierte Leser z. B. in [NMG_01] sowie [SIKO_01].

3.3 Sicherheit in WLANs

Wer heute über drahtlose Netzwerke schreibt, kommt um das Thema Sicherheit nicht herum. Die Blauäugigkeit, mit der einzelne Unternehmen noch vor wenigen Jahren ihre WLANs betrieben haben, ist weitestgehend verflogen. Im Internet finden auch unbedarfte Anwender schnell Programme, die Angriffe auf drahtlose Netzwerke durchführen und dem Abhörer die übertragenen Daten offenlegen. Die Sicherheit von WLANs spielt auch im Bereich Streaming-Media eine große Rolle. Während die Verschlüsselung von Streaming-Inhalten, wie z. B. einer Vorlesung, augenscheinlich noch keinen Sinn ergibt, erweist sich schon allein die Integration von Streaming-Media insbesondere per Multicasting in ein verschlüsseltes Infrastruktur-Netzwerk als problematisch.

Beim „sicheren“ Multicasting müssen alle Stationen zwei WEP-Schlüssel verwalten. Zum einen ihren eigenen privaten und zum anderen den gemeinsamen Schlüssel, der für Multicasts benutzt wird. Im Folgenden wird ein Überblick über die existierenden Verschlüsselungsmethoden sowie deren Integration in ein Streaming-Media-Umfeld dargestellt. Einfache Sicherheitsmechanismen wie das Setzen einer festen ESSID (Identifikation einer Zelle) oder MAC-Adressen-Authentifizierung finden dabei keine Berücksichtigung, da sie leicht zu fälschen und damit zu umgehen sind.

3.3.1 WEP Verschlüsselung

Ursprünglich wurde für den WLAN-Standard 802.11 die WEP(Wired Equivalent Privacy)-Verschlüsselung definiert. Dabei symbolisiert der Name bereits die Absicht des Verfahrens. Es sollte eine dem Kabel vergleichbare Sicherheit erreicht werden. Allerdings ist dies physikalisch in drahtlosen Netzwerken kaum machbar, während ein Kabel nur schwer abhörbar ist, insbesondere dann, wenn keine Spuren hinterlassen werden sollen (z. B. Veränderung des Widerstands der Leitung o. ä.). In einem drahtlosen Netzwerk ist ein Abhören theoretisch an jeder Station möglich. Alle Stationen teilen sich das Übertragungsmedium Luft, daher hören auch alle Stationen alle Nachrichten in ihrem Umkreis, auch wenn diese nicht an sie gerichtet sind. Sofern

eine abhörende Station also den Sender einer Nachricht empfangen kann, kann auch eine Entschlüsselung des übertragenen Inhalts erfolgen.

Die einfachste Verschlüsselung stellt somit die Kodierung der übertragenen Information selbst dar. Diese Kodierung erfolgt damit auf dem Layer 2, der Sicherungsschicht. Genau hier setzt das WEP-Verfahren an. Dabei wird ein RC4-Codec verwendet (siehe Abbildung 3-9), der die übertragene Information mit einem bis zu 128 Bit (effektiv 104 Bit, zuzüglich 24 Bit für den Initialisierungsvektor IV) langen Schlüssel chiffriert. Trotz der relativ einfachen Verschlüsselung über eine exklusive Operation ist der WEP-Algorithmus dank dem IV recht effektiv. Vor allem kommt der Geschwindigkeitsvorteil deutlich zum Tragen, eine XOR(exklusiv Oder)-Funktion lässt sich hardwaretechnisch ebenso billig wie schnell und effizient realisieren.

Der beim WEP verwendete Schlüssel ist der Empfangsstation bekannt und identisch mit dem Schlüssel des Senders. Durch die erneute Anwendung des Schlüssels und dem im Paket gespeicherten IV auf die übertragene Information kann der Empfänger die Daten schließlich wieder dechiffrieren.

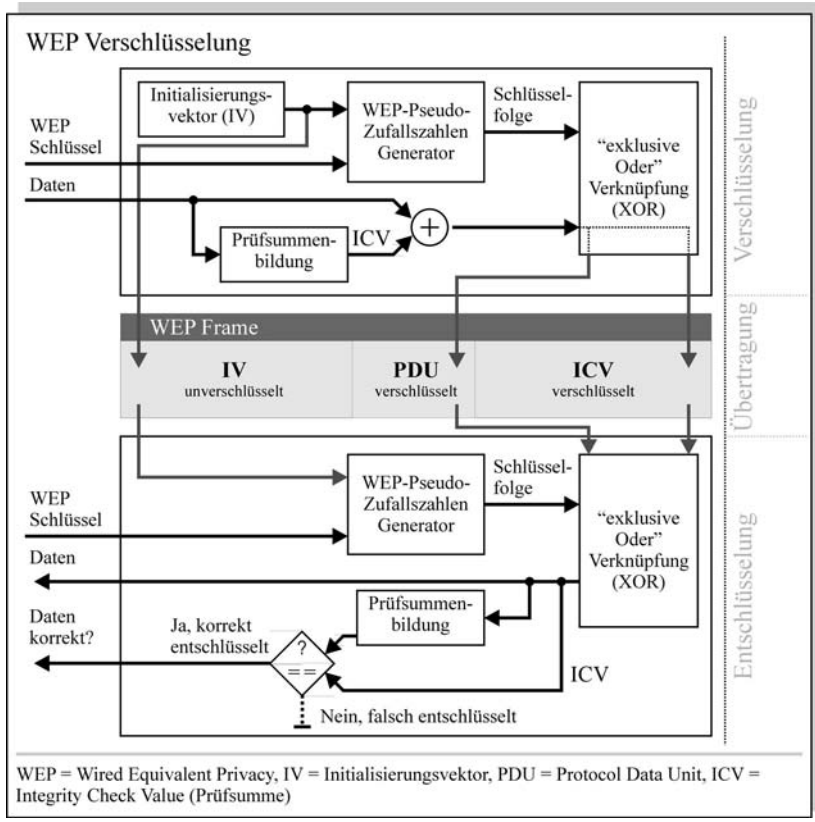

Abb. 3-9: WEP von der Verschlüsselung über die Übertragung bis zur Entschlüsselung

Abbildung 3-9 zeigt, wie beim WEP die zu übertragenden Daten verschlüsselt werden. Zunächst wird vom Sender ein zufälliger IV (Initialisierungsvektor) gebildet, der zusammen mit dem WEP-Schlüssel eine Pseudozufallszahlenfolge ergibt. Diese Folge wird auch als Schlüsselfolge bezeichnet. Die am Eingang anliegenden Daten werden zyklisch mit einer Prüfsumme versehen und werden schließlich mit der Schlüsselfolge über eine XOR-Funktion verknüpft. Dabei werden Daten und Checksummen getrennt in das zu übertragende Frame gefüllt. Neben diesen verschlüsselten Daten geht in das Frame der unverschlüsselte IV ein.

Beim Empfänger wird zunächst erneut eine Pseudozufallszahlenfolge mit dem gleichen IV (aus dem Frame) und dem gleichen WEP-Schlüssel (shared key – gemeinsamer Schlüssel) generiert. Die Schlüsselfolge des Empfängers sollte daher identisch mit der des Senders sein. Über eine erneute XOR-Verknüpfung der Schlüsselfolge mit den empfangenen Daten sollte der Empfänger die Daten korrekt entschlüsseln. Eine falsche Schlüsselfolge oder Verfälschungen während der Übertragung würden zu einer fehlerhaften Datenfolge führen. Indem der Empfänger über den vermeintlich entschlüsselten Datenstrom erneut zyklisch eine Prüfsumme (ICV) bildet und diese mit der ICV des Frames vergleicht, kann er schließlich vergleichen, ob die Daten, die er ermittelt hat, auch korrekt sind, sprich den Eingangsdaten entsprechen. Näheres zur WEP-Verschlüsselung findet sich in [NMG_01].

Diese Verschlüsselungsvariante weist mehrere Schwachstellen auf, die bereits seit 1998 bekannt sind.

Schwachstellen von WEP

Der beim WEP verwendete IV stellt eine große Schwachstelle dar. Dem IV steht nur ein relativ kleiner Adressraum zur Verfügung, was eine wiederholte Verwendung der gleichen IVen in einem relativ kurzen Zeitraum zur Folge hat. Kennt ein Angreifer den folgenden IV, kann er statistische Attacken (Auswertung der Verteilung der restlichen Codeelemente im Schlüssel) auf die Verschlüsselung vornehmen. Kennt der Angreifer den IV, kennt er de facto bereits einen Teil des Schlüssels.

Eine weitere Methode stellt die „Known Plaintext"-Attacke dar, bei der der Angreifer versucht, bekannten Inhalt von Paketen im Netz verschlüsselt wiederzufinden. Dabei kann der Angreifer z. B. von außen eine Station innerhalb des WLAN pingen. Schließlich kann er die Pakete im WLAN auswerten (sniffen) und durch den bekannten Inhalt seines Ping Rückschlüsse auf den verwendeten Verschlüsselungs-Key gewinnen.

Ferner kann der Angreifer eine „Denial of Service"(Ausfall des Access Point)-Attacke gegen den Access Point starten, da die An- und Abmeldung der Stationen an diesem ohne gesicherte Authentifizierung o. ä. erfolgt. Dabei sendet er solange in Folge Anmeldungen (Associates bzw. Disassociates) an den Access Point, bis dieser den Dienst einstellt („Denial of Service").

Mit einer Wörterbuch-Attacke („Dictionary Attack") kann der Angreifer, sofern der Schlüssel aus einem Passwort generiert wurde, versuchen, den WEP-Schlüssel zu erlangen. Gelingt ihm dies nicht per Wörterbuch, so kann er versuchen, den Schlüssel über rohe Gewalt („brute force") zu knacken.

Dabei iteriert er alle möglichen Schlüssel und IVen solange durch, bis er den korrekten Schlüssel ermittelt hat.

Eine weitere Schwachstelle von WEP, die nicht auf die Sicherheit abzielt, ist der Performance-Verlust. Auch wenn die XOR-Operation sehr schnell über eine Nachricht angewendet ist, benötigt WEP neben der zusätzlichen Bandbreite auch mehr Rechenleistung beim Access Point. Bei mehreren Stationen, die per WEP an den Access Point angebunden sind, kann dies ein erheblicher Faktor sein.

Lösungen der Schwachstellen

Verschiedene Hersteller (z. B. Avaya mit WEP+) und auch das IEEE haben Erweiterungen für WEP spezifiziert. Diese Erweiterungen bieten allerdings ebenfalls keinen vollständigen Schutz. Beim WEP2 wird der IV auf 128 Bit verlängert und es kommen teilweise 802.1X sowie die Vergabe von dynamischen Schlüsseln zum Einsatz. Dabei kann auch dieses Verfahren keine vollständige Sicherheit bieten. Allerdings stellt die Vergabe von dynamischen Schlüsseln beim WEP eine sehr gute und sichere Lösung dar. In der Fachliteratur findet man diesen Ansatz auch unter dem Namen „Dynamische Schlüsselverwaltung“. Dabei werden einer Station während ihrer Anmeldung am drahtlosen Netzwerk dynamisch die verwendeten Schlüssel übergeben. Im Idealfall werden die Schlüssel dabei nach einem Public-/Private-Key-Verfahren verschlüsselt. Dadurch wird sichergestellt, dass sie nur der Empfänger entschlüsseln und anwenden kann.

Wird der dynamische Schlüsselaustausch in einem bestimmten Zeitintervall automatisch wiederholt, so ist es für einen Angreifer, der die oben genannten Schwachstellen von WEP ausnutzen will, kaum möglich, Zugang zu den Informationen zu bekommen. So kann z. B. ein Intervall von 5 Minuten gewählt werden, nach dem komplett neue Schlüssel ausgehandelt werden. In einem Zeitraum von 5 Minuten ist es einem Angreifer selbst per „brute force“-Attacke nicht möglich, die Daten zu dechiffrieren. Um Zugang zum WLAN zu erhalten, müsste der Angreifer alle möglichen IVen mitspeichern und bei einer erneuten Verwendung eines IVs versuchen, den Schlüssel zu knacken. Dies dauert in der Praxis einige Tage, da die meisten APs derzeit ein Hash-Verfahren für die Vergabe der Schlüssel verwenden, das eine Vergabe des gleichen IVs innerhalb kurzer Zeiträume wirkungsvoll unterbindet.

Der Overhead, der durch den Schlüsselaustausch entsteht, wird dabei weitgehend durch die hohe Sicherheit gerechtfertigt. WEP lässt sich also durchaus effektiv für die Verschlüsselung der übertragenen Daten anwenden, sofern es mit anderen Verfahren, die seine Schwächen eliminieren, verwendet wird. Die noch nicht fertig spezifizierte Erweiterung des IEEE 802.11i

[TGI] könnte die derzeit herstellerabhängige Implementierung der dynamischen WEP-Verschlüsselung resp. der Schlüsselverwaltung in einen Standard überführen. Allerdings wird derzeit geprüft, ob in 802.11i zusätzlich z. B. der RC4 gegen einen AES(Advanced Encryption Standard)[AES]-Algorithmus ausgetauscht werden könnte.

3.3.2 Autorisierung nach 802.1X

Bereits bei den Lösungen der WEP-Schwachstellen in 3.3.1 wurde ein Verfahren zum dynamischen Verwalten von Schlüsseln beschrieben. Heutige Implementierungen dieses Verfahren nutzen in der Regel Erweiterungen des 802.1X-Standards [802.1X]. 802.1X hat grundsätzlich allerdings nichts mit dynamischer Schlüsselverwaltung zu tun. Das IEEE definierte 802.1X als Standard für „Port-based Network Access Control“. Übertragen kann man sich eine solche Authentifizierung von Ports im Ethernet (802) so vorstellen:

Die Link-LED einer Netzwerkkarte oder eines Hubs bzw. Switches leuchtet erst dann auf, wenn nicht nur eine physikalische Verbindung des Ports mit dem Netzwerk besteht, sondern dieser auch authentifiziert ist.

Es besteht somit praktisch keine physikalische Verbindung zum Netz vor der Authentifizierung. 802.1X kann in 802.11-WLANs (sowie in allen weiteren 802-LANs) zum Einsatz kommen, wobei sich der Begriff Port virtuell auf einen Slot eines Access Points bezieht. Sind mehrere Stationen an einem Access Point angemeldet, so benutzt jede Station einen ihr zugewiesenen Slot. Dieser Slot wird bei der Anmeldung der Station (Assoziierung) vom Access Point intern vergeben. Er stellt die Basis für die Verbindung zum Client dar. Genau diese physikalische Basis der Verbindung lässt sich beim 802.1X in WLANs selektiv aktivieren bzw. deaktivieren je nach Berechtigung des Clients.

Abbildung 3-10 zeigt den Ablauf einer Authentifizierung und damit eines Portzugriffs beim 802.1X. Beim 802.1X wird zwischen einem Supplicant, einem Authenticator und einem Authentication Server unterschieden. Der Supplicant stellt einen z. B. mobilen Client dar, der Zugriff auf ein Netz erlangen möchte. Der Authenticator leitet die Anfrage des Supplicant in das Netz weiter. Er stellt dem Supplicant zwei virtuelle Ports zur Verfügung. Zum einen den uncontrolled Port, über den ein Client (als Supplicant) frei Daten versenden kann, sowie einen controlled Port, über den ein Client erst nach erfolgreicher Authentifizierung senden darf. Der uncontrolled Port lässt dabei nur EAP-Pakete hindurch. Somit ist über ihn lediglich die Authentifizierung selbst möglich. Der controlled Port ist nicht auf die Versendung von

EAP-Paketen beschränkt und bietet dem Client einen unbegrenzten Zugriff auf das Netz.

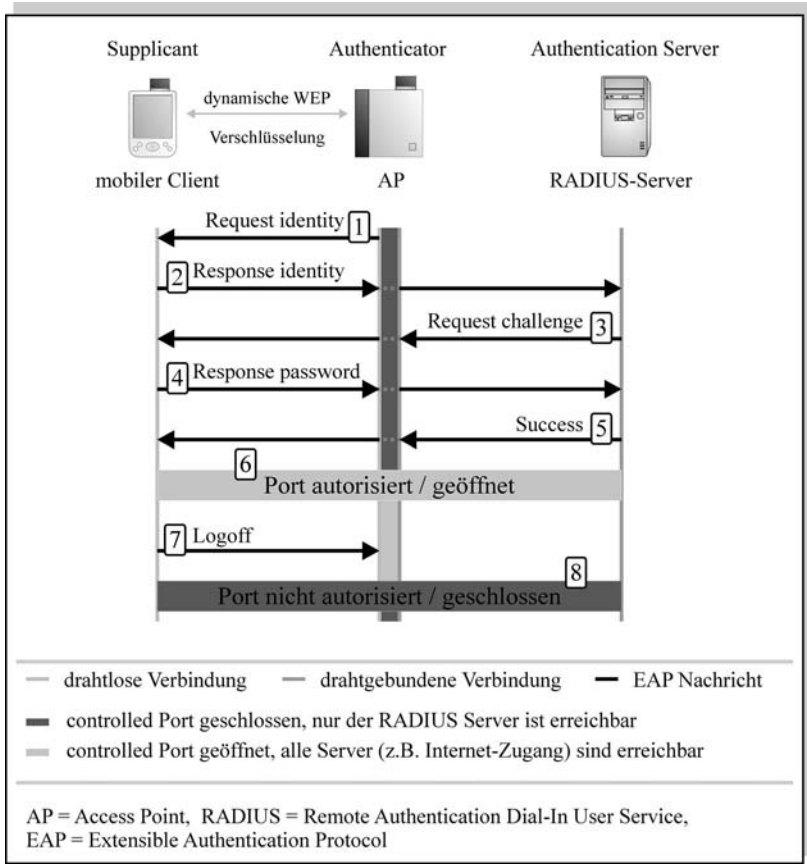

Abb. 3-10: Ablauf der Authentifizierung eines Ports beim 802.1X

In Abbilbung 3-10 bezieht sich der Begriff Port auf den controlled Port. Hellgrau bedeutet daher in diesem Fall, dass der controlled Port geöffnet ist, und der Client damit vollen Zugriff auf das Netz erhält. Dunkelgrau beschreibt den umgekehrten Fall. Im gezeigten Ablauf werden folgende Schritte unterschieden:

1. Der Access Point, an den sich der Client anmeldet, schickt als Authenticator an den Client als Supplicant einen Request nach dessen Identität.

2. Der Client beantwortet den Request des Authenticators mit seiner Identität in Form einer EAP-Nachricht. Dies kann z. B. mittels Zertifikat oder aber auch per Username bzw. Domäne und Hostname geschehen. Der Authenticator leitet die Identität des Client über den uncontrolled Port zur Überprüfung an den Authentication Server, den ein RADIUS-Server implementiert, weiter.
3. Der Authentication Server sendet einen Request nach einer Challenge (einer Bewerbung um den Port) über den uncontrolled Port des Authenticators an den Supplicant zurück.
4. Der Client sendet daraufhin ein Passwort oder ein Zertifikat, dass ihn zum Zugriff auf das Netz zulassen soll, über den uncontrolled Port des Authenticators an den Authentication Server.
5. Der Authentication Server prüft nun seinerseits die Übereinstimmung des Passworts oder des Zertifikats mit der in Schritt 2 eingegangenen Identität. In diesem Fall ist das Zertifikat bzw. das Passwort des Supplicant gültig. Daher sendet der Authentication Server eine Success Response an den Authenticator, der seinerseits daraufhin den controlled Port für den Supplicant freigibt und die Success Response an ihn weiterleitet.
6. Der Port ist nun für den Supplicant vollständig geöffnet. Er kann über ihn das gesamte Netz erreichen. Ab hier leuchtet die Link-LED der Netzwerkkarte des Supplicants und signalisiert die physikalische Verfügbarkeit der Verbindung.
7. Der Supplicant möchte die Verbindung beenden und sendet die EAP-Nachricht Logoff an den Authenticator.
8. Daraufhin schließt der Authenticator den controlled Port und trennt damit die Verbindung zum Client, dessen Link-LED daraufhin erlischt. Nun muss ein eventueller Client sich zunächst erneut um den Zugriff auf diesen Port bewerben, bevor er eine Verbindung zum Netz herstellen kann.

Für die Kommunikation zwischen Supplicant und Authenticator werden die EAP-Pakete in so genannte EAPOL(EAP over LAN)-Frames verpackt, die direkt im LAN auf MAC-Ebene verschickt werden. Somit wird zwischen Client und AP die Sicherheit auf dem Layer 2 garantiert. Authenticator und Authentication Server kommunizieren über RADIUS-Pakete miteinander, die in UDP-Paketen verpackt sind.

Implementierungen

Während der Standard 802.1X die Authentifizierung klar vorschreibt, ist die Art und Weise, in der diese ablaufen soll, frei implementierbar. Daher existieren verschiedene Ansätze.

- EAP-MD5 (Extensible Authentification Protocol – Message-Digest 5)

  Dabei werden die Authentifizierungsdaten als nach MD5 (Message-Digest Algorithm) codierte Strings versendet und vom Authentification Server in der Regel als „clear text"-Strings verglichen.

  Diese Methode bietet im Vergleich zu den nachfolgenden eine geringere Sicherheit, lässt sich jedoch sehr viel einfacher in bestehende RADIUS-Server integrieren. Die genaue Spezifikation des MD5-Algorithmus findet sich in [RFC_1321].

- EAP-TLS (Extensible Authentification Protocol – Transport Layer Security)

  Hierbei werden die Authentifizierungsdaten als TLS-Zertifikate (z. B. nach X.509) übertragen. TLS (Transport Layer Security) ist der Nachfolger vom SSL (Secure Socket Layer), der auch für Homebanking-Portale oder gesicherte Webseiten im Internet eingesetzt wird. Prinzipiell handelt es sich beim TLS um ein einfaches „plain text"-Protokoll, das nach Zertifikaten der beiden Gegenstellen fragt und diese nach dem Public/Private-Key-Verfahren (ähnlich PGP) vergleicht und verifiziert. Dabei ist anders als beim MD5 eine Authentifikation des Servers gegenüber dem Client möglich, was „Man-in-the-Middle"-Attacken unterbindet. Bei diesem Verfahren bekommt jeder Client ein eigenes Zertifikat, was seine Identität eindeutig macht. Dieses Zertifikat ist dem Server bekannt. Der Transport der Daten nach dem EAP-TLS ist besonders bei großen Schlüssellängen sehr sicher. Derzeit werden überwiegend DES-Schlüssel verwendet. In Zukunft werden AES-Schlüssel zu einer noch größeren Sicherheit beitragen. EAP-TLS ist in [RFC_2716] ausführlich beschrieben.

- EAP-TTLS (Extensible Authentification Protocol – Tunneled TLS)

  Die Implementierung einer Infrastruktur nach dem EAP-TLS gestaltet sich in der Praxis als schwierig. Dabei muss für jeden Client ein separates Zertifikat erzeugt und eine Zertifizierungsautorität mit allen Zertifikaten und eigenem Stammzertifikat gebildet werden. Aufgrund dieses hohen, vorrangig administrativen Aufwands wurde das EAP-TTLS auf der der Basis des EAP-TLS entwickelt. Dabei dient das TLS nur für die

Gewährleistung der Authentizität des Servers (durch sein öffentliches Zertifikat als Public Key), der die Verschlüsselung der Verbindung initiiert. Über diese verschlüsselte Verbindung werden schließlich unsichere Authentifizierungsprotokolle getunnelt. Somit kann innerhalb einer TLS-Verbindung z. B. die Authentifizierung per Klartext-Passwort und Benutzername erfolgen, ohne dass dies ein Sicherheitsrisiko darstellt. Durch diesen Vorteil kann EAP-TTLS in einem bestehenden WLAN nach wie vor auf Benutzername und Passwort als Authentifizierung aufsetzen, ohne die Erstellung von separaten Zertifikaten zu erzwingen.

Weitere Verfahren sind z. B. EAP-AKA [AKA], EAP-GSS [GSS], PEAP [PEAP] und LEAP [LEAP] von der Firma Cisco. Alle diese Verfahren unterstützen die Anmeldung per Benutzername und Passwort getunnelt über eine sichere Verbindung sowie Authentifizierung des Servers gegenüber dem Client per Zertifiktat. PEAP ist z. B. Bestandteil von Windows XP Service Pack 1 sowie dem Microsoft .NET Server.

Schwächen und Lösungen

Allein der Einsatz von 802.1X in einem WLAN kann noch nicht alle Sicherheitsrisiken abdecken. Erst die Kombination von 802.1X z. B. mit WEP macht die Authentifizierung sicher. 802.1X beinhaltet zwei bekannte Schwachstellen, die in [SA1X] beschrieben sind. Zum einen kann ein Angreifer beim 802.1X eine bereits authentifizierte WLAN-Verbindung „entführen“. Dabei sendet er an den Client eine sog. Disassociate(Abmelde)-Nachricht unter Vorgabe der Identität des Access Points. Der Client terminiert daraufhin seine Verbindung und der Angreifer kann diese unter der Vorgabe der Identität des Clients übernehmen. Dieses Verfahren funktioniert nur, wenn für die Verbindung keine Verschlüsselung (z. B. nach WEP) verwendet wird. Ansonsten könnte der Angreifer weder die Disassociate-Nachricht an den Client versenden (da diese vom Client nur korrekt verschlüsselt verarbeitet werden kann), noch nach der Übernahme der Verbindung mit dem Access Point kommunizieren.

Die zweite Schwäche des 802.1X stellt die Möglichkeit eines sog. Man-In-The-Middle-Angriffs dar. In diesem Fall gibt eine Station zwischen Client und Access Point vor, selbst der Access Point zu sein. Dabei leitet sie die Anfragen des Clients an den Access Point weiter und erhält damit selbst eine Autorisierung. Dieser Angriff ist nur dann möglich, wenn sich der Access Point nicht gegenüber dem Client authentifiziert. Wird somit z. B. das Protokoll EAP-TLS oder EAP-TTLS verwendet, ist dieser Angriff nicht mehr möglich, da der Client hier ein Zertifikat vom Server erhält, das als Public Key für die Kommunikation genutzt wird. Der Angreifer besitzt in diesem

Fall nicht den notwendigen Private Key, um die Nachrichten des Clients zu entschlüsseln.

Um 802.1X für die Verteilung von dynamischen WEP-Schlüsseln zu verwenden, wird in der Regel das MPPE(Microsoft Point-to-Point Encryption)-Protokoll [MPPE] als Erweitung der RADIUS-Antwort verwendet. Dabei wird z. B. beim EAP-TLS bzw. EAP-TTLS der Session Key des TLS [TLS] auf die Länge des WEP-Schlüssels, z. B. 104 Bit, gekürzt und direkt verwendet. Damit kann ein dem Client und Server bekannter Schlüssel definiert werden, ohne dass dieser Schlüssel öffentlich über das Netz verschickt, manipuliert oder ausspioniert werden kann. Der Session-Key einer TLS-Sitzung wird von Client und Server beim Start der Sitzung aus dem erhaltenen Schlüsselmaterial generiert.

3.3.3 Verfahren oberhalb von Layer 2

Vielerorts werden für sichere drahtlose Netzwerke VPNs (Virtual Private Networks) eingesetzt.

VPNs stellen einen sicheren Transporttunnel innerhalb eines ungesicherten, öffentlichen Netzwerks her. Dieser Tunnel wird in der Regel durch das Verfahren IPsec (IP security) [IPSEC] geschützt. Abbildung 3-11 zeigt die Nutzung von VPN-Tunneln in einem WLAN.

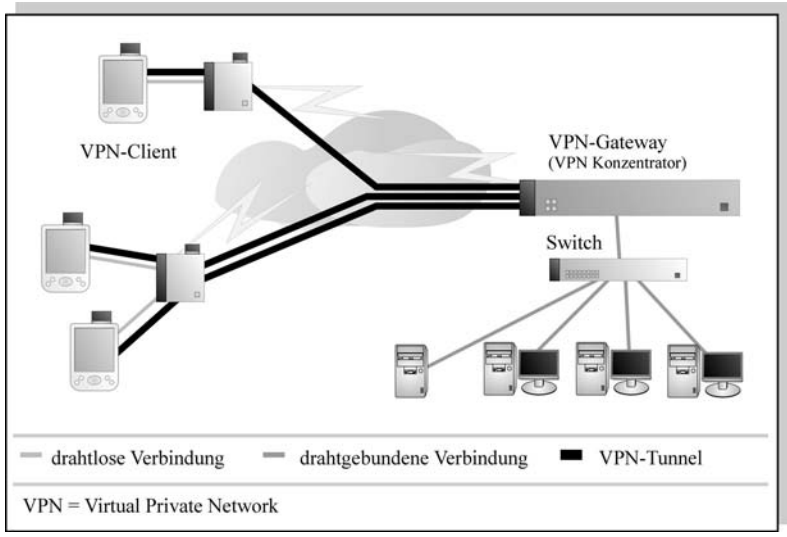

Abb. 3-11: Verschlüsselte Übertragung im WLAN mittels VPN (IPsec)

Beim IPsec werden alle transportierten IP-Pakete verschlüsselt (in IPsec-Pakete verpackt). Dadurch kann eine hohe Sicherheit der Daten erreicht werden. Insbesondere kommen beim IPsec sehr große Schlüssellängen und sichere Schlüsselaustauschverfahren [IKE] zum Einsatz. Auch wenn VPNs als WLAN-Infrastruktur eine nahezu absolute Sicherheit bieten, haben sie auch einige Nachteile:

- Sehr hohe Prozessorlast sowohl beim Client (Sender) als auch beim IPsec-Gateway (Empfänger)
- Großer, zusätzlicher Overhead in den Paketen durch die IPsec-Header
- Tunnel-Struktur bietet kein effizientes Multicasting an mehrere IPsec-Teilnehmer
- Anmeldungen per Zertifikat führen zu hohem Verwaltungsaufwand

Der Nachteil der hohen Prozessorlast kommt durch die aufwändige Ver- und Entschlüsselung der Pakete zu Stande. Gerade für kleine mobile Endgeräte kann dies eine zu große Belastung sein.

Der große zusätzliche Overhead entsteht insbesondere durch die zusätzlichen Header, die aufgrund des IPsec vor das IP-Paket gestellt werden. Nicht zuletzt bewirkt jedoch auch die Verschlüsselung selbst mitunter eine geringe Erhöhung des Datenaufkommens.

Aufgrund der Tatsache, dass jeder Tunnel einen expliziten Unicast-Kanal zum Empfänger darstellt (schließlich handelt es sich bei einem Tunnel per Definition um eine Punkt-zu-Punkt-Verbindung), wird ein Multicasting zu mehreren IPsec-Empfängern unmöglich. Die einzelnen Tunnel besitzen alle eigene Schlüssel, was ein Entschlüsseln eines Paketes durch mehrere Teilnehmer unmöglich macht. Die Pakete müssen separat verschlüsselt und übertragen werden.

Ähnlich wie das EAP-TLS beim 802.1X-Verfahren, sorgt die Verwaltung von einzelnen Zertifikaten der VPN-Clients für einen großen Aufwand. Daher bietet z. B. die Firma Cisco mit ihren VPN-Clients die Möglichkeit, gegen einen bestehenden RADIUS-Server mittels Benutzername und Passwort zu authentifizieren. Diese Erweiterung des IPsec ist jedoch nicht Bestandteil des ursprünglichen Standards. Näheres zu VPNs siehe [BADA_01].

Um die Nachteile der VPN-Verschlüsselung zu beheben, kann auf 802.1X und dynamische WEP-Schlüssel zurückgegriffen werden. Sollte dies nicht möglich sein, können grundsätzlich auch andere Verfahren oberhalb von Layer 2 eingesetzt werden. Für Homebanking über das Internet oder andere

gesicherte Web-Seiten wird das Verfahren SSL bzw. sein Nachfolger eingesetzt. Diese Verfahren lassen sich auch für andere Protokolle neben HTTP (für die Übertragung von Web-Seiten) verwenden. Insbesondere besteht beim SSL die Möglichkeit, unsichere Protokolle ähnlich wie beim IPsec über SSL zu tunneln. SSL und TLS stellen sehr einfache Klartextprotokolle dar, die über gegenseitige Zertifikate Server und Client authentifizieren und so eine gesicherte Übertragung ermöglichen. Dabei verwendet der Client den Public Key (in Form des Zertifikats) des Servers als Schlüssel für seine Pakete. Die Daten können schließlich nur vom Server selbst mit dem Private Key wieder entschlüsselt werden. Auch andere Protokolle wie SSH (Secure Shell) [SSH] bieten die Verschlüsselung der transportierten Nutzdaten (hier einer Terminalverbindung). SSH bietet sogar ebenfalls wie SSL einen gesicherten Tunnel für die Übertragung von unsicheren Daten über eine SSH-Sitzung an.

Daher stellt sich die Frage, ob man für sich persönlich ein sicheres drahtloses Netzwerk als ein solches definiert, bei dem alle übertragenen Informationen verschlüsselt werden. Stellt z. B. die Möglichkeit des Mitlesens einer aufgerufenen Web-Seite im Netz durch einen dritten weniger ein Problem dar, so können SSL-, TLS-, SSH- usw. Verbindungen für eine völlig ausreichende Sicherheit sorgen. Dabei könnte ein Angreifer zwar z. B. die übertragene Web-Seite zu einem Client mitlesen, nicht aber eine gesicherte Homebanking-Verbindung.

Weitere Informationen zum Thema TLS finden sich in [TLS]. SSL ist in [SSL] ausführlicher beschrieben.

3.3.4 Multicast-Schlüssel in WLANs

Bei der dynamischen WEP-Verschlüsselung, wie sie auch im Abschnitt 3.3.1 beschrieben wurde, bekommt jede Station (bzw. jede Sitzung) ihren eigenen WEP-Schlüssel. Damit wird ein Senden von Paketen an mehrere Stationen gleichzeitig (Broadcast bzw. Multicast) unmöglich. Daher wird bei der dynamischen WEP-Verschlüsselung neben dem Unicast-Schlüssel der einzelnen Stationen zusätzlich ein Broad- bzw. Multicast-Schlüssel vergeben. Mit diesem Schlüssel können die Stationen schließlich Pakete, die an alle oder mehrere Stationen gerichtet sind, entschlüsseln. Die Dechiffrierung der ausschließlich an sie gerichteten Pakete erfolgt dabei weiterhin über ihren eigenen Schlüssel.

Dieses Verfahren besitzt neben der Problemlösung des Multicasting in dynamisch verschlüsselten WLANs einen weiteren Vorteil. Da der Unicast Key als zweiter WEP-Key einer Station eingetragen wird, können auch Stationen,

die die dynamische Schlüsselvergabe nicht unterstützen, über den gemeinsamen WEP-Schlüssel (der dem Broad- und Multicast-Schlüssel entspricht) am WLAN teilnehmen. Allerdings werden in einem sicheren WLAN in der Regel auch die Broad- und Multicast-Schlüssel automatisch nach einem definierten Intervall gewechselt, was die Nutzung durch Stationen, die keine dynamische Schlüsselverwaltung unterstützen, unterbindet. Gerade für sichere Multicast-Streams ist ein solches regelmäßiges Ändern der Broad- und Multicast-Schlüssel notwendig.

3.4 Sicherheit bei Bluetooth

Auch wenn beim Bluetooth, bedingt durch die kurze Reichweite, augenscheinlich keine hohe Sicherheit erforderlich ist, definiert der Standard diese recht ausführlich. Bedenkt man, dass mittels Bluetooth auch der Zugriff auf Unternehmensnetzwerke oder z. B. kostspielige Handy-Verbindungen möglich wird, gewinnt die Sicherheit auch einen höheren Stellenwert. Bluetooth definiert im Wesentlichen drei Sicherheitsprofile:

- Security Mode 1: keine Verschlüsselung / Authentifizierung
- Security Mode 2: Verschlüsselung / Authentifizierung auf der Service-Ebene
- Security Mode 3: Verschlüsselung / Authentifizierung auf der Link-Ebene

Der einfachste, Security Mode 1, bietet keinerlei Sicherheit, lässt sich dafür allerdings ebenso leicht implementieren wie anwenden. Er kommt z. B. für Bluetooth-Erweiterungen wie Kopfhörer, Fernbedienungen o. ä. zum Einsatz.

Security Mode 2 sichert den Zugriff auf der Service-Ebene ab. Ein Bluetooth-Gerät kann mehrere Services anbieten. Z. B. kann ein Handy zum einen für den Austausch von Visitenkarten und zum anderen für die Verbindung über einen Mobilfunkanbieter als Modem genutzt werden. In diesem Beispiel wäre die Nutzung des Security Mode 2 sinnvoll. Dadurch könnte der Austausch von Visitenkarten mit dem Handy ohne Verschlüsselung und Authentifizierung des Nutzers erfolgen, während die kostspielige Nutzung einer Internetverbindung über das Handy nur verschlüsselt und nach vorheriger Autorisierung erfolgen kann.

Beim Security Mode 3 ist generell keine Verbindung zum Gerät ohne Verschlüsselung und Authentifizierung mehr möglich. Dies bietet z. B. für einen Bluetooth-Zugangspunkt zum Unternehmensnetzwerk eine hohe Sicherheit.

In Mode 2 und 3 kommt dabei eine der WEP-Verschlüsselung sehr ähnliche, allerdings in puncto Sicherheit des Verfahrens erweiterte Lösung zum Einsatz. Eine genaue Beschreibung des Verschlüsselungsverfahrens kann in [BTSIG] nachgelesen werden. Die Schlüssellänge beträgt bei Bluetooth 128 Bit. Wobei die Schlüssel aus der 48-Bit-Bluetooth-Adresse, einer automatisch vergebenen Schlüsselfolge, der Master Clock (Zeitgeber) und einer 128-Bit-Zufallszahlenfolge gebildet werden. Beim Bluetooth werden diese Werte über eine XOR-Operation (exklusives Oder) miteinander verknüpft, wobei ein lineares feedback des Schlüssels über eine zusätzliche XOR-Operation in die Schlüsselfolge mit einfließt. Die genaue Beschreibung des Algorithmus kann in [BTSIG] nachgelesen werden.

4. Streaming-Media in drahtlosen Netzwerken

Das folgende Kapitel beschäftigt sich mit der eigentlichen Kernaussage dieser Diplomarbeit. Dabei wird die Realisierung und Anwendung von Streaming-Media in drahtlosen Netzwerken auch unter der Verwendung von Multicasting beschrieben.

4.1 Realisierung von Streaming-Media in drahtlosen Netzwerken

Für die Realisierung sowie zufriedenstellende Anwendung von Streaming-Media in Übertragungsnetzwerken allgemein müssen die in den folgenden Abschnitten genannten Grundeigenschaften gewährleistet sein.

Neben diesen Grundeigenschaften bieten einige Streaming-Media-Server in der Praxis weitere Eigenschaften, die speziell für die Unterstützung von drahtlosen Teilnehmern implementiert wurden. Ein Beispiel hierfür ist die FEC (Forward Error Correction). Beim Versand mit FEC-Unterstützung wird grob gesagt einer Sequenz von Paketen ein Paket aus einer vorherigen Sequenz beigefügt. Durch diese erneute Übertragung des Pakets, die eine Redundanz darstellt, kann bei Paketverlusten innerhalb der vorherigen Sequenz eine Korrektur vorgenommen werden. Da drahtlose Netze eine sehr viel höhere Paketverlustrate aufweisen als z. B. drahtgebundene, kann auf diese Art und Weise eine sehr viel bessere Wiedergabe erreicht werden. Vor allem aufgrund der Tatsache, dass bei der Übertragung von Streaming-Media i. d. R. Protokolle wie UDP eingesetzt werden, die ein fehlerhaftes Paket zu Gunsten der Übertragungsgeschwindigkeit nicht erneut anfordern. Die FEC lässt sich bei den Streaming-Media-Servern, die diese unterstützen, meist

frei konfigurieren. Genauer gesagt lässt sich der Anteil, den die wiederholten Pakete in der aktuellen Übertragungssequenz erhalten sollen, frei einstellen.

Bei der Realisierung von Streaming-Media in drahtlosen Netzwerken sollte aufgrund der Übertragungseigenschaften, wie sie in Abschnitt 4.2 beschrieben werden, gegebenenfalls ein weitaus größer dimensionierter Jitter-Puffer (Abschnitt 1.1.1) für den Media-Player definiert werden.

4.1.1 Anforderungen an die verfügbare Bitrate

Sofern Unicasts (Abschnitt 2.1.1) für den Transport der Streaming-Media-Daten verwendet werden sollen, kann die Formel 4-1 für die Berechnung der erforderlichen Bitrate verwendet werden:

*Formel 4-1: Minimum der verfügbaren Bitrate = s * m * 1,1 Bit/s*

Dabei steht s für die Bitrate (Bit/s) des übertragenen einzelnen Streams und m für die Anzahl der Betrachter bzw. Streaming-Media Empfänger. Der Faktor 1,1 bietet einen Puffer, der Steuerkommandos des Streaming-Protokolls (Abschnitt 1.1.3) während der Wiedergabe (z. B. indem der Benutzer den Stream anhält oder zurückspult) sowie den Verkehr eines Kontrollprotokolls wie z. B. das RTCP (Abschnitt 1.6.5) beinhaltet. Dieser Faktor deckt keinesfalls alle Möglichkeiten ab, die zusätzliche Bandbreite während der Stream-Übertragung benötigen. Z. B. kann durch ein VPN zwischen Sender und Empfänger zusätzlicher Overhead und damit Bitratenbedarf durch zusätzliche Header entstehen. Der Faktor bildet daher lediglich einen Grundwert, der an die jeweilige Anwendung und Netzwerk-Struktur angepasst werden muss.

Für einen Stream mit 300 KBit/s, der an 50 Teilnehmer verteilt wird, wäre nach der Formel 4-1 eine Bitrate von: 300 KBit/s * 50 * 1,1 = 16500 KBit/s = 16,5 MBit/s notwendig.

Es ist zu beachten, dass diese Bitrate an jeder Stelle der Route durch das Übertragungsnetz sowie am Sender selbst zur Verfügung stehen muss (wie in Abschnitt 1.1.1 beschrieben).

Bei der Verwendung von Multicasts für die Übertragung fällt der Faktor m aus der Formel 4-1 heraus.

4.1.2 QoS-Anforderungen (Verzögerung, Abweichung, Fehlerrate)

Die Übertragung von Video und Audio stellt extrem hohe Anforderungen in punkto maximaler Verzögerung sowie maximaler Laufzeitschwankung bzw. Verzögerungsabweichung (Jitter) der Pakete. Diese Anforderungen wurden

bereits im Abschnitt 1.1.2 sowie 1.2.1 und 1.2.2 herausgestellt. In drahtlosen Netzwerken treten sehr viel größere Verzögerungen und Verzögerungsabweichungen auf, als dies in drahtgebundenen Netzwerken der Fall ist. Dies resultiert aus der Übertragung per Funk, die als Signal absorbiert, reflektiert und damit verfälscht werden kann. Außerdem steigt die Verzögerung mit zunehmender Distanz zum Sender stärker als in drahtgebundenen Netzwerken.

Die QoS-Anforderungen werden in erster Linie durch die Echtzeitfähigkeit des drahtlosen Netzwerks bzw. seiner Übertragungstechnik bestimmt. Diese Eigenschaften sind im Abschnitt 4.2 ausführlich beschrieben.

4.1.3 Multicasting und effiziente Verteilung

Wird Streaming-Media in drahtlosen Netzwerken an eine große Zahl von gleichzeitigen Teilnehmern (z. B. mehr als 20) als Live-Stream (Abschnitt 1.1) gesendet, so kann die Übertragung per Multicast eine wichtige Anforderung werden. Bei der Übertragung per Multicast bleibt die erforderliche Bitrate konstant (Abschnitt 2.1.5) und ist genau so groß wie die Rate eines Streams an einen einzelnen Teilnehmer. Für die Übertragung an eine große Zahl von Teilnehmern kann Multicasting in einem drahtlosen Netzwerk sogar die einzige Realisierungsmöglichkeit sein. Drahtlose Netzwerke besitzen in der Regel eine relativ geringe effektive Bandbreite (Abschnitt 3.2.1), in der sich z. B. beim 802.11b nur etwa ein paar Dutzend Streams per Unicast ausliefern lassen:

Formel 4-2: 11 MBit/s / 2 / 300 KBit/s = ~18 Teilnehmer per Unicast

Bei einer Bitrate von 300 KBit/s lässt sich ein Video mit einer guten Qualität in einer Auflösung von etwa 320 x 240 flüssig übertragen (Abschnitt 1.4.3). Die maximale Bitrate von 802.11b muss in der Praxis sehr viel geringer angenommen werden als die theoretische von 11 MBit/s. Daraus resultiert der Faktor 2, der die Bitrate in der Formel 4-2 dividiert. Dieser Wert stellt einen Erfahrungswert dar, der zum einen die Bitrate an das arithmetische Mittel der Bitraten der 802.11b-Modulationstechniken annähert ((11 MBit/s + 5,5 MBit/s + 2 MBit/s + 1 MBit/s) / 4 = 4,875) und zum anderen z. B. in [CT_01] als Praxis-Messwert erfasst wurde. Wird ein sehr stark räumlich verteiltes drahtloses Netzwerk betrieben, so kann dieser Wert auch > 2 sein. In drahtlosen Netzwerken, die nur einen Raum abdecken, kann er im Idealfall auch geringfügig < 2 sein.

Bei der Übertragung des Streams per Multicast verliert die Formel 4-2 ihre Wirkung. Aufgrund der Tatsache, dass im 802.11b-Netzwerk die Adressierung der IEEE-802-Familie genutzt wird, gilt hier die im Abschnitt 2.1.3

beschriebene Umsetzung der IP-Multicast-Adressen auf entsprechende MAC-Adressen. Somit wird das Multicasting in diesem Netz unterstützt. Die Übertragung des in Formel 4-2 verwendeten Streams mit einer Bitrate von 300 KBit/s benötigt somit unabhängig von der Anzahl der Teilnehmer in jedem Fall und für jede Anzahl von Teilnehmern nur 300 KBit/s. Durch dieses Verfahren lassen sich somit nicht nur Bandbreite für die Übertragung einsparen und weitere Anwendungen oder sogar mehrere unterschiedliche Streams parallel an alle Teilnehmer realisieren, sondern vor allem das Nadelöhr der geringen Bandbreite in drahtlosen Netzwerken effizient umgehen.

Eine ausführliche Beschreibung der Multicasting im 802.11b WLAN findet sich im Abschnitt 4.4.

Am Rande sei bemerkt, dass sogar Entwürfe bzw. Protokolle für ein Routing von Multicast-Nachrichten in drahtlosen Netzwerken existieren. So unterstützt z. B. das ABAM-Protokoll [ABAM] ein Routing von Multicast-Nachrichten in drahtlosen Ad-Hoc-Netzwerken, wie sie im Abschnitt 3.1.2 beschrieben werden. Solche Ansätze zählen zu den Multicast-Routing-Protokollen, wie sie im Abschnitt 2.2 beschrieben werden. Auch beim ABAM wird versucht, einen Multicast-Baum aus allen mobilen Empfängern zu bilden. Multicast-Routing-Protokolle sind, wie bereits im Abschnitt 2.2 herausgestellt, kein zentrales Thema der vorliegenden Arbeit. Außerdem existieren z. B. für das ABAM bis dato keine gebräuchlichen Implementierungen.

4.2 Echtzeitfähigkeit von drahtlosen Netzwerken

Bereits im Abschnitt 4.1.2 wurde die Relevanz der QoS-Eigenschaften eines drahtlosen Netzwerks für die Übertragung von Streaming-Media herausgestellt. Die wichtigen QoS-Parameter für Video sind im Abschnitt 1.2.2 dargestellt, für Audio wurden die Eigenschaften im Abschnitt 1.2.1 aufgezählt. QoS-Eigenschaften werden in erster Linie durch die Echtzeitfähigkeiten des drahtlosen Netzwerks beschrieben.

Die Echtzeitfähigkeiten des Netzwerks beziehen sich dabei vorrangig auf die QoS-Eigenschaften Verzögerung sowie Verzögerungsabweichung (Jitter). Die maximale Bitrate ist genau wie die Fehlerrate ebenfalls wichtig (siehe auch Abschnitt 4.1.3), wirkt sich jedoch sehr viel weniger störend aus, als die Verzögerungen bzw. Jitter. Bei einer geringen Bitrate lässt sich etwa die Auflösung eines übertragenen Videos anpassen. Eine erhöhte Fehlerrate kann z. B. durch Prädiktionsalgorithmen Fehler aus einem Einzelbild (auch bei hohen Fehlerraten) derart ausgleichen, dass der Betrachter dies nur schwer bemerkt. Selbst wenn bei einer Bildrate von z. B. 25 Bildern pro

Sekunde ein komplettes Bild ausfällt, wird diese Störung mit einer Dauer von 1/25 Sekunde kaum bemerkt bzw. akzeptiert.

Anders gestaltet sich dies bei Verzögerungen. Treten mitten im Stream Lükken z. B. mit einer Dauer von mehr als ein paar Sekunden auf, ist die Qualität für den Betrachter sehr schnell inakzeptabel. Noch schlimmer als die reine Verzögerung ist eine Verzögerungsabweichung innerhalb der Wiedergabe. Die Wiedergabe ist in diesem Fall nicht länger flüssig, sondern umgangssprachlich „ruckelig“.

Abbildung 4-1 zeigt den Zusammenhang zwischen Verzögerung und Verzögerungsabweichung von Paketen. Wie bereits im Abschnitt 1.1 erläutert, werden Verzögerungsabweichungen (Jitter) beim Streaming in der Regel durch Jitter-Puffer beim Empfänger gemindert.

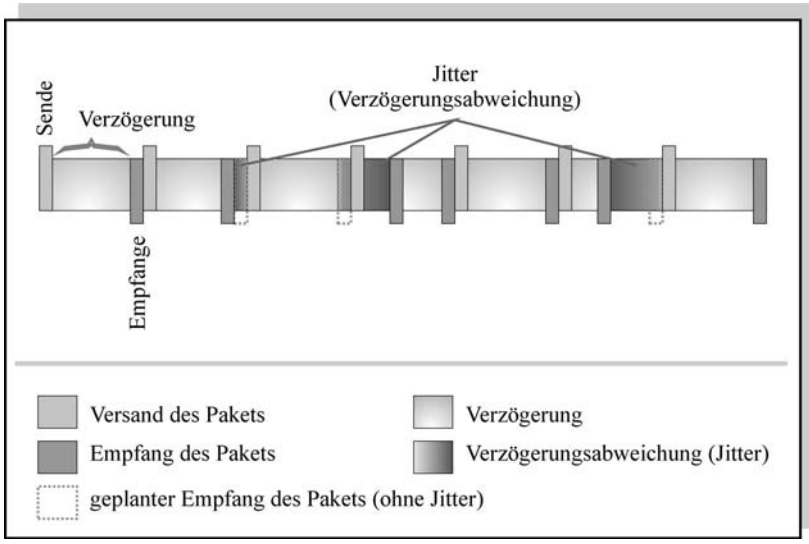

Abb. 4-1: Verzögerung und Verzögerungsabweichung zwischen Versand und Empfang von Paketen

Die Fähigkeiten der in dieser Arbeit verwendeten Übertragungsverfahren, Verzögerungen und Jitter auf der physikalischen Übertragungsebene zu vermeiden, soll in den folgenden Abschnitten erläutert werden. Im Abschnitt 4.1.1 werden die Echtzeitfähigkeiten von 802.11 (WLAN) herausgestellt, während im Abschnitt 4.1.2 die Verzögerungs- und Jitter-Eigenschaften von Bluetooth dargestellt sind.

Für alle drahtlosen Netzwerke gilt, dass alle QoS-Parameter abhängig von der Entfernung zwischen Sender und Empfänger sowie der Signalqualität sind. Funkwellen können zwischen Sender und Empfänger z. B. reflektiert oder absorbiert werden, wie bereits im Abschnitt 3 erwähnt. Dabei treten Bitfehler auf, die Fehlerrate steigt, und es kommt zur erneuten Übertragung des Frames (Schicht 2) oder sogar Paketes. Dadurch erhöht sich die Verzögerung bei der Übertragung des Paketes. Die Schwankung der Verzögerung durch erneute Übertragung einzelner Frames bzw. ganzer Pakete hat dadurch höhere Jitter (Verzögerungsabweichungen) zur Folge. Dieser Effekt der „Bitverfälschung“ während der Übertragung ist bei drahtlosen Netzen weitaus gravierender als z. B. bei drahtgebundenen Netzen.

In den folgenden Betrachtungen wird jeweils davon ausgegangen, dass sich die Stationen (Sender und Empfänger) in unmittelbarer Nähe befinden (ca. 2 - 10 Meter) und dass die Stärke des Signals beim Empfänger ausreichend ist.

4.2.1 Echtzeitfähigkeit von 802.11-WLAN

Charakteristische Verzögerungen eines 802.11-WLAN in Abhängigkeit der Größe der Pakete zeigt die Tabelle 4-1 aus [NMG_01]:

Paketgröße (Bytes)	**100**	**250**	**500**	**750**	**1000**
Minimum	1,63	3,22	3,16	4,35	4,69
Maximum	5,00	8,33	7,81	11,02	14,51
Mittelwert	2,84	3,65	5,00	6,31	7,65

Tabelle 4-1: Verzögerung in Abhängigkeit von der Paketgröße in Millisekunden

Dabei wurde dieser Test in [NMG_01] durch ein simples Ping-Pong-Protokoll mit UDP-Paketen realisiert. Solche UDP-Pakete kommen auch häufig beim Versenden von Streaming-Media zum Einsatz, da im Gegensatz zum TCP hier keine gesicherte Verbindung benutzt wird [BADA_01]. Z. B. im Falle einer Video-Übertragung ist die erneute Übertragung eines zuvor fehlerhaft empfangenen Bildes auch wenig sinnvoll, da das Bild zu einem späteren Zeitpunkt nicht länger in die Reihenfolge des Videos passt.

Aus Tabelle 4-1 ist erkennbar, dass die Verzögerung pro Paket mit zunehmender Paketgröße wächst. Während dieser Effekt auch in drahtgebundenen Netzen existiert, ist die relativ große Differenz zwischen Maximum und Mittelwert von entscheidender Bedeutung. Die in Tabelle 4-1 aufgeführten Verzögerungszeiten sind grundsätzlich sowohl für eine Video- als auch für eine Audio-Übertragung akzeptabel. Nach [BADA_01] sowie Vorgaben der ITU (International Telecommunications Union) sind Verzögerungen im Audio-Bereich unter 150 ms nicht wahrnehmbar sowie zwischen 150 und 400 ms tolerierbar. Bei der Übertragung von Video gelten andere maximal tolerierbare Verzögerungen. Davon ausgehend, dass ein „flüssiges“ Video 25 Bilder pro Sekunde liefern soll, müsste für das Übertragungsnetz, sofern sich das Differenzbild in einem einzigen Paket übertragen lässt, eine maximale Verzögerung von:

1/25 Sekunde = 40 ms

gelten. In der Regel lassen sich jedoch auch Verzögerungen bis zu 50 ms tolerieren (20 Bilder pro Sekunde). Die Verzögerung bezieht sich dabei auf den gesamten Übertragungsweg bis zur Wiedergabe inkl. Wartezeit z. B. in Jitter-Puffern oder Ko- bzw. Dekodierung des Paketes.

Bei genauer Betrachtung stellt die Verzögerung einen weniger relevanten Echtzeit Parameter dar, da der Betrachter in der Regel eine Verzögerung von mehreren Sekunden, vor der Wiedergabe des Streams akzeptiert. Selbst bei der Wiedergabe mehrerer synchroner Streams als Streaming-Media wirkt sich eine Verzögerung in den in Tabelle 4-1 gezeigten Größen nicht negativ aus (Video und Audio werden z. B. bei ca. 80 ms Versatz, sprich Verzögerung, als „lippensynchron“ akzeptiert). Die Verzögerung bekommt eine größere Relevanz, wenn der Betrachter nicht passiv den Stream empfängt, sondern interaktiv darauf antwortet. So kann z. B. bei einer Telefonverbindung auch eine relativ geringe Verzögerung inakzeptabel werden, da sie sich effektiv zwischen den beiden Teilnehmern im Dialog verdoppelt.

Weitaus problematischer sind die Verzögerungsabweichungen (Jitter). Aus Tabelle 4-1 lässt sich erkennen, dass z. B. für Pakete mit einer Größe von 1000 Bytes ein maximaler Versatz (Differenz zwischen Minimum und Maximum) von ca. 10 ms möglich ist. Diese Hundertstel Sekunde kann, wenn man bedenkt, dass für ein Einzelbild mitunter mehrere Pakete benötigt werden, zu merklichen Störungen der Wiedergabe führen. Allerdings verwenden nahezu alle Media-Player für die Minderung dieses Effekts einen sog. Jitter-Puffer. In ihm werden alle Pakete eingereiht und vor der Wiedergabe in die korrekte Reihenfolge gebracht. Der Nachteil dieser Methode ist die erhöhte

Wartezeit auf den Wiedergabebeginn. Der Vorteil ist die Glättung von Verzögerungsabweichungen.

Für Multicast-Nachrichten kommt eine weitere QoS-Anforderung hinzu. Da diese Pakete in der Regel nicht wiederholt oder in irgendeiner Form gesichert werden, ist die Verlustrate solcher Pakete auf dem Übertragungsnetz entscheidend. In [NMG_01] wird eine Verlustrate für Multicast-Frames im WLAN von 0,03 % angegeben. Im dort aufgeführten Beispiel gingen von 13.355.215 Nachrichten an drei Rechner insgesamt (für alle Rechner) 4.695 Nachrichten verloren. Diese Rate ist ebenfalls völlig akzeptabel, da bei einer Rate von 25 Bildern pro Sekunde auch der Ausfall eines gesamten Bildes vom Auge kaum wahrgenommen würde. Angenommen, die aus dem Beispiel von [NMG_01] gesendeten 13.355.215 Nachrichten enthielten alle jeweils ein Bild (bzw. ein Differenz-Bild – siehe Abschnitt 1.4.2), so würde erst eine Verlustrate von ca. 6 % einen Verlust von durchschnittlich einem Bild pro Sekunde bedeuten.

Eine große Anforderung wird an 802.11-WLANs bei der Verteilung von Streaming-Media in Echtzeit an das Zugriffsverfahren gestellt. In der Regel dürfen sich sendewillige Stationen im WLAN direkt um den Zugriff auf das Medium bewerben und meist sofort senden. Genauer gesagt wird jedoch vor dem Senden eine zufällige Backoff-Zeit abgewartet. Dadurch ist die Wartezeit nicht deterministisch. Bei der standardmäßigen DCF (siehe Abschnitt 3.2.1) können somit zufällige Verzögerungen auftreten. Für die Einhaltung der QoS-Anforderungen ist in diesem Zusammenhang das PCF(Point Coordination Function)-Zugriffsverfahren aus Abschnitt 3.2.1 sinnvoll. Bei diesem Verfahren tritt ein zentraler Verwalter, der PC (Point Coordinator), für den Zugriff auf das Netzwerk ein. Dieser PC fragt die sendewilligen Stationen im Netz reihum ab und erteilt ihnen den Zugriff. Durch die Implementierung dieser PCF kann z. B. gewährleistet werden, dass Streaming-Media-Daten, die eine Station (auch aus dem LAN) in das WLAN sendet, bevorzugt verteilt werden.

4.2.2 Echtzeitfähigkeit von Bluetooth

Aufgrund der Tatsache, dass Bluetooth ein separates Profile für die Unterstützung von Telefonverbindungen besitzt, bietet es ein gesondertes Zugriffsverfahren, das Echtzeit-Übertragungen ermöglicht. Die beiden beim Bluetooth verwendeten Zugriffsverfahren wurden bereits im Abschnitt 3.2.2 beschrieben. Der SCO (Synchronous Connection-Oriented Link), der für zeitkritische Verbindungen, insbesondere Telefonverbindungen, definiert wurde, vergibt dabei Zeitschlitze für die jeweiligen Clients. Anders als die PCF des 802.11-WLAN werden diese Zeitschlitze jedoch fest zugewiesen.

Die Clients (auch Slaves genannt) erhalten somit reihum einen fest definierten Zeitschlitz. Für die Vergabe der Zeitschlitze wird das TDMA/TDD(Time Division Multiple Access / Time Division Duplex)-Verfahren verwendet. Dabei wird der Zeitraum, in dem Übertragungen möglich sind, in mehrere Zeitschlitze eingeteilt. Da ein Bluetooth-Piconet genau 8 Stationen (1 Master und 7 Slaves bzw. Clients) unterstützt, werden 8 Zeitschlitze definiert und der jeweiligen Station fest zugewiesen.

Ansonsten ähnelt das Verfahren stark der im Abschnitt 3.2.1 beschriebenen PCF des 802.11-WLAN. Auch bei der SCO existiert ein Koordinationspunkt in Form des Masters im Piconet (Bluetooth-Netzwerk-Profil), der die Zeitschlitze verwaltet.

Ein Zeitschlitz nimmt beim Bluetooth einen Zeitraum von 625 µs ein. Maximal muss eine sendewillige Station im Piconet unter Verwendung der SCO somit 7 Zeitschlitze abwarten, (da insgesamt 8 Stationen und Zeitschlitze definiert sind) was einer Wartezeit von

*7 * 625 µs = 4375 µs = 4,375 ms*

entspricht. Diese Wartezeit kann somit als maximale Verzögerung angenommen werden. Bereits im Abschnitt 4.1.1 wurde erwähnt, dass nach [BADA_01] sowie Vorgaben der ITU eine Verzögerungszeit von weniger als 150 ms nicht bemerkbar ist sowie Zeiten bis 400 ms vom Zuhörer toleriert werden. Die Verzögerungszeit unter Verwendung der SCO bei Bluetooth ist somit völlig akzeptabel. Hinzu kommt, dass, wie ebenfalls im Abschnitt 4.1.1 erwähnt, die Verzögerung an sich weniger kritisch für die Echtzeitfähigkeit ist. Viel relevanter in diesem Zusammenhang sind Jitter (Verzögerungsabweichungen). Da allerdings bei der SCO feste Zeitschlitze zugeteilt und zentral verwaltet werden, liegen die Jitter zwangsläufig nahe Null. Es existieren keine Jitter, da jede Station zu einer fest definierten Zeit (Zeitschlitz) Zugriff auf das Medium erhält.

Obwohl die Echtzeitfähigkeiten von Bluetooth in diesem Abschnitt als hervorragend dargestellt werden, sei erneut daraufhin gewiesen, dass diese Fähigkeiten nur für einen Verbund von maximal 7 Stationen sowie eines Masters gewährleistet werden. Werden mehr als 7 Clients in einem Piconet mit einem Master betrieben, so nehmen diese passiv an der Kommunikation teil. Solche Stationen müssen explizit vom Master aktiviert werden. Die Verzögerung ihrer Übertragungen kann somit im schlimmsten Fall ins Unendliche steigen.

4.3 Streaming-Media in drahtgebundenen Netzwerken

Eine Übertragung von Streaming-Media-Inhalten in drahtgebundenen Netzwerken gestaltet sich sehr viel einfacher als in den in dieser Arbeit behandelten drahtlosen Netzwerken. Der Unterschied zwischen der Übertragung von Streaming-Media in drahtlosen Netzwerken und drahtgebundenen Netzwerken besteht in erster Linie in den unterschiedlichen Quality of Service(QoS)-Eigenschaften der beiden Übertragungsmedien. Der größte Unterschied liegt dabei in der Fehlerrate. Während in drahtlosen Netzwerken bedingt durch Interferenzen, Absorption sowie Reflektion der Funkwellen häufig Bitfehler entstehen, ist die Fehlerrate in drahtgebundenen Netzwerken nahezu Null. Selbst wenn die Fehlerrate in einem drahtgebundenen Netz anwachsen sollte, so lässt sich z. B. eine Bruchstelle o. ä. leicht finden und beseitigen. Bei drahtlosen Netzen kommen solche Fehlerraten meist durch ungünstige Umgebung, Witterung (z. B. auch nasses Laub an Bäumen im Herbst) und Vegetation sowie unzählige andere Faktoren zustande. Eine Behebung der Fehlerrate lässt sich daher nicht wie im drahtgebundenen Netz z. B. „einfach“ auf eine Kabelbruchstelle zurückverfolgen.

Die extrem geringe Fehlerrate in drahtgebundenen Netzwerken führt durch die damit verbundene geringe Anzahl von erneuten Übertragungen auch zu verhältnismäßig kleinen Verzögerungszeiten und vor allem Verzögerungsabweichungen. Da die Kodierung des Signals auf einem Kabel sowieso weniger robust (im Vergleich z. B. zum Barker-Code im Abschnitt 3.2.1) durchgeführt werden muss, und die Übertragung des Signals annähernd in Lichtgeschwindigkeit erfolgen kann, sind die Verzögerungszeiten ohnehin weitaus geringer als bei der Übertragung von Funkwellen.

Über ein Kabel lassen sich durch mehrere Adern zusätzlich mehrere Übertragungskanäle parallel nutzen. Ein Vorgang, für den in drahtlosen Netzwerken pro Kanal eine separate Antenne benötigt würde bzw. ein separater Empfänger. Durch diese Bündelung von Kanälen wird z. B. das Gigabit-Ethernet in drahtgebundenen Netzwerken realisert, das vier Aderpaare eines Kat.5(Kategorie 5)-Kabels benötigt.

Gigabit-Ethernet deutet als Schlagwort bereits auf den Vorteil in Bezug auf den QoS-Parameter „Bitrate“ bei drahtgebundenen Netzwerken hin. In drahtgebundenen Netzwerken existieren sehr viel höhere Bitraten als in den drahtlosen Netzwerken. Die höchste verfügbare Bitrate in drahtlosen Netzwerken beträgt 54 MBit/s, während im drahtgebundenen Ethernet bis zu 1 Gigabit/s geboten werden. 10-Gigabit-Ethernet befindet sich als Backbone auf Glasfaserbasis ebenfalls auf dem Vormarsch. Selbst bei einer 100-MBit/s-Anbindung der Arbeitsplätze und einem 100-MBit/s-Backbone

lässt sich ein 300-KBit/s-Stream nach der Formel 4.1 an die folgende Anzahl von Teilnehmern verteilen:

*Minimum der verfügbaren Bitrate = s * m * 1,1 Bit/s*

$$m = \frac{Maximum\ der\ verfügbaren\ Bitrate}{s \cdot 1{,}1 Bit/s} = \frac{100 MBit/S}{300 KBit/s \cdot 1{,}1 Bit/s} \approx 300 Teilnehmer$$

Bei einer Bitrate von 300 KBit/s lässt sich ein Video mit einer guten Qualität in einer Auflösung von etwa 320 x 240 flüssig übertragen (Abschnitt 1.4.3). In einem WLAN lassen sich, wie im Abschnitt 4.1.3 gezeigt, weitaus weniger Teilnehmer versorgen.

Bei einer Anzahl von 303 möglichen Teilnehmern verliert manches Multicast-Szenario seine Berechtigung. Trotzdem lassen sich auch Multicasts sehr einfach in drahtgebundenen Netzwerken realisieren. Vorraussetzung dafür ist lediglich, dass die verwendeten Bridges bzw. Switches, die die einzelnen Segmente (auch Collision Domains) miteinander verbinden, Multicasts weiterleiten. Nahezu alle Komponenten im Ethernet-Bereich unterstützen dies mittlerweile. In Netzwerken, die aus mehreren Segmenten zusammengesetzt werden, existieren unter Umständen auch separate IP-Subnetze. Diese Subnetze und vor allem die zugehörigen Router müssen ein Multicast-Routing, wie im Abschnitt 2.2 gezeigt, unterstützen.

Abschließend sei bemerkt, dass drahtgebundene Netze aufgrund ihrer guten QoS-Eigenschaften und vor allem durch die hohe Bitrate ideal als Zubringer für drahtlose Netzwerke dienen können. Dabei kann ein einziges drahtgebundenes Segment mehrere drahtlose Netzwerke (Zellen) anbinden. Drahtgebundene Netzwerke bieten sich außerdem ideal für die Realisierung von Content Delivery Networks an, bei denen sich sog. Edge-Server am Rande des Netzes und damit in unmittelbarer Nähe zum Nutzer etablieren. Diese Edge-Server beziehen den auf ihnen gelagerten Inhalt schließlich von einem zentralen z. B. Streaming-Server.

4.4 Multicast-Techniken im WLAN

Aufgrund der Tatsache, dass der WLAN-Standard 802.11 zur 802-Ethernet-Familie gehört, kann im WLAN direkt auf die von der IEEE vorgegebene Unterstützung von Multicasts zurückgegriffen werden. Die Basis dafür bildet die Umsetzung von IP-Multicast-Adressen auf entsprechende MAC-Adressen (wie im Abschnitt 2.1.3 beschrieben).

Zusätzlich müssen die APs (Access Points), die zwischen WLAN und LAN als Bridge fungieren, Multicasts unterstützen bzw. vom einen in das andere Netz weiterleiten. Auch die eigentlichen WLAN-Adapter an den mobilen Stationen müssen Multicast-Adressen implementieren. Durch die einheitlichen Vorgaben des 802.11 bzw. der 802-Familie ist diese Unterstützung bei nahezu allen gängigen WLAN-APs sowie Adaptern gegeben.

Die einzige Besonderheit bei der Verwendung von Multicasting im 802.11-WLAN kommt bei der Verwendung von dynamischen WEP-Schlüsseln zur Absicherung des Netzwerks zum Tragen. Da in einem solchen Szenario jede Station einen eigenen, dynamisch zugewiesenen WEP-Schlüssel erhält, wird die Verschlüsselung eines Multicast-Paketes mit einem gemeinsamen Schlüssel unmöglich. Ein statischer WEP-Schlüssel im WLAN, der auf jeder Station inkl. dem AP identisch ist, hingegen stellt kein Problem dar. Mit ihm können auch Multicasts verschlüsselt werden, da jede Station den für die Dechiffrierung notwendigen Schlüssel besitzt. Für die Realisierung von Multicasting in WLANs mit dynamischem WEP wird in den gängigen Produkten ein zweiter WEP-Schlüssel, der sog. Broad- und Multicast-Schlüssel, definiert. Dieser lässt sich z. B. im Zusammenspiel mit 802.1X automatisch an die Stationen verteilen. Dieser Aspekt wurde auch im Abschnitt 3.3.2 erwähnt.

4.5 Multicast-Techniken bei Bluetooth

Rein physikalisch wären bei Bluetooth die Bedingungen für die Unterstützung von Multicasts die gleichen wie bei 802.11-WLAN. Insbesondere der in der Entwicklung befindliche 802.15-WPAN[802.15]-Standard des IEEE, der Kurzstreckennetzwerke mittels Bluetooth definiert, sollte durch die 802-Familie Multicast-Adressen unterstützen. In der Bluetooth-Spezifikation 1.1 [BTSIG] ist allerdings bis dato keine Unterstützung für Multicasting berücksichtigt. Es werden weder gesonderte Adressen reserviert noch Transportmechanismen für Multicasts vorgesehen.

Ein viel größeres Manko, das die Unterstützung von Multicasting insbesondere wie im Abschnitt 2.1.2 beschrieben verhindert, ist die Tatsache, dass bei Bluetooth keine IP-Pakete und damit IP-Adressen auf dem L2CAP-Layer verwendet werden. Eine Umsetzung zwischen logischer IP-Adresse auf Bluetooth-Hardware-Adresse ist somit nicht möglich. IP-Pakete werden im derzeit spezifizierten LAP (LAN Access Profile) von Bluetooth ausschließlich eingebetet in PPP(Point to Point Protocol)-Paketen übertragen. Man spricht dabei auch von einer Verkapselung der IP-Pakete in PPP-Pakete. Eine Point-to-Point-Verbindung stellt jedoch gerade einen Unicast da, bei dem von einem Punkt zu genau einem anderen gesendet wird. In diesem Fall

würden Multicast-Pakete in separaten Punkt-zu-Punkt-Verbindungen an die Empfänger übertragen. Der Vorteil des Multicasting würde sich somit erübrigen.

Abbildung 4-2 zeigt das Problem der Nutzung von PPP zur Übertragung von IP-Paketen. Während auf dem gemeinsam genutzten Bluetooth-Übertragungsnetz theoretisch Multicasts übertragen werden könnten, müssen die IP-Pakete aufgrund der Tatsache, dass diese über eine PPP-Verbindung versendet werden, separat zu jedem Empfänger transferiert werden.

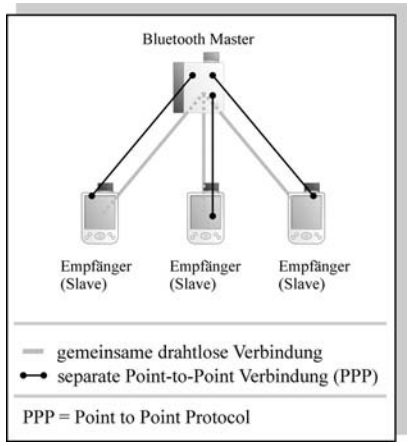

Abb. 4-2: Nutzung von PPP zur IP-Paket-Übertragung bei Bluetooth

Das PPP lässt sich nicht als Point-to-Multipoint-Protokoll verwenden. Daher ließe sich das Problem nur durch eine Realisierung der Übertragung von IP-Paketen auf dem L2CAP-Layer (in dem direkt Bluetooth-Adressen verwendet werden) realisieren. Dieses Verfahren befindet sich derzeit unter anderem in [802.15] in Entwicklung.

In diesem Fall würde allerdings immer noch die Umsetzung von IP-Multicast-Adressen auf Bluetooth-Multicast-Adressen fehlen, die auch in [802.15] sowie anderen Erweiterungen des Bluetooth-Standards [BTSIG] nicht vorgesehen sind.

Aufgrund der geringen maximalen Teilnehmerzahl in Bluetooth-Piconets (maximal 7 Clients) stellt sich ohnehin die Frage, ob der zusätzliche Aufwand eines Multicasting direkt ad absurdum führt. Denkbar wäre z. B. das Versenden von Broadcasts mit Streaming-Inhalten direkt an die Clients. Pro-

blematisch würde dies jedoch im Scatter-Net (einem Verbund mehrerer Piconets – wobei der Master des einen Piconet Slave im jeweils anderen wird). Hier würden sich Broadcasts unkontrolliert vermehren.

Gerade für Bluetooth wäre jedoch die Lösung der Übertragung mittels Multicasting bzw. Broadcasting elementar, da die verfügbare Bitrate sehr gering ist.

4.6 Anwendungsszenarien

Streaming-Media und drahtlose Netzwerke besitzen beide für sich ein ernormes Wachstum. Immer mehr Unternehmen, Heimanwender und Behörden setzen WLANs für die Realisierung ihrer Netzwerke oder als Ergänzung ein. Streaming-Media-Inhalte besitzen im Internet ebenfalls ein großes Wachstum, z. B. in den im Abschnitt 1.1 sowie 1.9 genannten Bereichen.

Die Konvergenz dieser beiden Entwicklungen liefert die möglichen Szenarios für Streaming-Media in drahtlosen Netzwerken und damit die Anwendungsmöglichkeiten dieser Diplomarbeit. Der Bereich Streaming-Media wird als einer der großen Wachstumsmärkte der Zukunft gehandelt. Dabei stützen sich die Prognosen auf die Ablösung von Fernseher, Radio, Telefon sowie der Konvergenz aller zu einem kompakten Gerät, das alle diese Funktionen realisiert. Teilbereiche dieser Vision sind schon heute sinnvoll einsetzbar.

Streaming-Media im WLAN in Wartehallen, Schulungsräumen usw.

Abbildung 4-3 zeigt einen möglichen Einsatz von Streaming-Media über ein WLAN. Dabei wird eine Wartehalle, z. B. am Flughafen, mit Streaming-Media (wie einer Nachrichten-Sendung, einem Börsenbericht oder einer Flughafeninformation) beliefert. Dieses Szenario kann anstelle einer Wartehalle auch ein Café, einen Schulungsraum o. ä. darstellen.

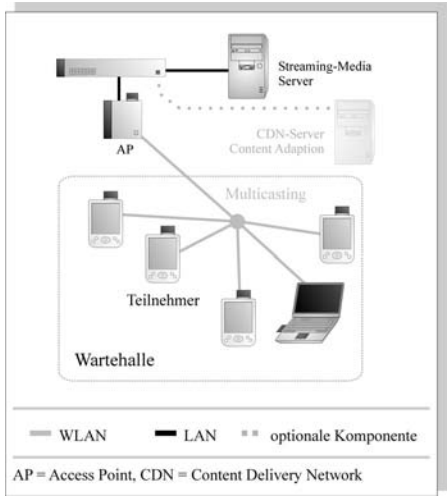

Abb. 4-3: Beispiel für Streaming-Media im WLAN: Wartehalle, Café, Schulungsraum…

Da in dem in Abbildung 4-3 gezeigten Szenario unter Umständen sehr viele Teilnehmer im drahtlosen Netzwerk existieren könnten, sollte die Verteilung des Streaming-Media wenn möglich per Multicasting erfolgen. Multicasting setzt implizit einen Live-Stream voraus. Sollen die Teilnehmer jedoch „on demand" aus einer Auswahl von verschiedenen Streaming-Media-Inhalten wählen können, so kann die Übertragung nur per Unicast erfolgen. In diesem Fall reduziert sich die Anzahl der möglichen Teilnehmer in einer Zelle auf die im Abschnitt 4.1.3 ermittelte Zahl.

Als optionale Komponente kann in diesem Szenario neben dem Streaming-Media-Server ein weiterer Server eingesetzt werden. Dieser Server könnte z. B. eine Content-Adaption (Anpassung des Streaming-Inhalts) durchführen. So könnten z. B. in einer Wartehalle sehr unterschiedliche Teilnehmer mit dem Netz verbunden sein. Z. B. könnte ein Teilnehmer einen PDA mit einer vergleichsweise geringen Bildschirmauflösung verwenden, während ein anderer sein Notebook mit einer hohen Auflösung verwendet. Mitunter könnten die Teilnehmer auch eine unterschiedliche Nationalität haben und unterschiedliche Sprachen sprechen. Die Content-Adaption, z. B. nach dem Protokoll ICAP [ICAP], könnte in einem solchen Fall, entsprechend nach den Informationen der Clients (z. B. Nationalität und Auflösung des

Displays), einen passenden Stream auswählen, der den Inhalt ideal auf den jeweiligen Teilnehmer anpasst. Die Content-Adaptionsfähigkeit lässt sich auch in einem Edge-Server eines CDNs (Content Delivery Network) unterbringen. Ein CDN verteilt den Content auf mehrere Server, insbesondere Edge-Server, die am Rande des Netzes und damit in unmittelbarer Nähe zum Teilnehmer für eine gute und flüssige Übertragung sorgen. Ein solches CDN ermöglicht auch die Verteilung von Streaming-Media an unterschiedliche WLANs, wie in Abbildung 4-3 gezeigt.

Beispiel: Fernseh- bzw. Rundfunk-Sender

Das in Abbildung 4-4 gezeigte Szenario zeigt die Verteilung von Streaming-Media im WLAN am Beispiel eines Rundfunk- oder Fernseh-Senders. Dabei wird die laufende Sendung z. B. über einen Studio-PC direkt auf einen Streaming-Media-Server übertragen. Dieser Server beliefert die einzelnen Abteilungen des Senders mit einem Live-Stream. Viele Sender drängen derzeit auf den Verzicht einer separaten Koax-Verkabelung für diesen Zweck hin zur einheitlichen Übertragung über TP-Kabel bzw. Glasfaser. Dadurch lassen sich auch entfernte Funkhäuser oder Aussenstellen über WAN-Verbindungen mit dem Stream versorgen. Gerade in der Redaktion existiert jedoch zusätzlich ein weiterer Trend. Die Redakteure arbeiten zunehmend frei und ggf. zu Hause an einzelnen Beiträgen, die sie direkt auf dem Streaming-Media-Server oder im Archiv ablegen. Um diese Clients flexibel im Netz zu integrieren, bietet sich gerade ein WLAN an. Durch den Einsatz von Multicasting, wie in 4.2.1 beschrieben, lassen sich dadurch auch größere Abteilungen mit der laufenden Sendung versorgen.

Geradezu nebenbei kann der Streaming-Media-Server auch einen Live-Stream ins Internet schicken und damit dem Wachstum des im Abschnitt 1.9 genannten Internet-TV bzw. Internet-Radio Rechnung tragen.

Auch das Archiv kann auf diese Weise flexibel angebunden werden. Dadurch können die Redakteure, aber auch z. B. die Marketingabteilung, etwa auf Werbespots, Trailer, Jingles o. ä. zugreifen und diese einsetzen.

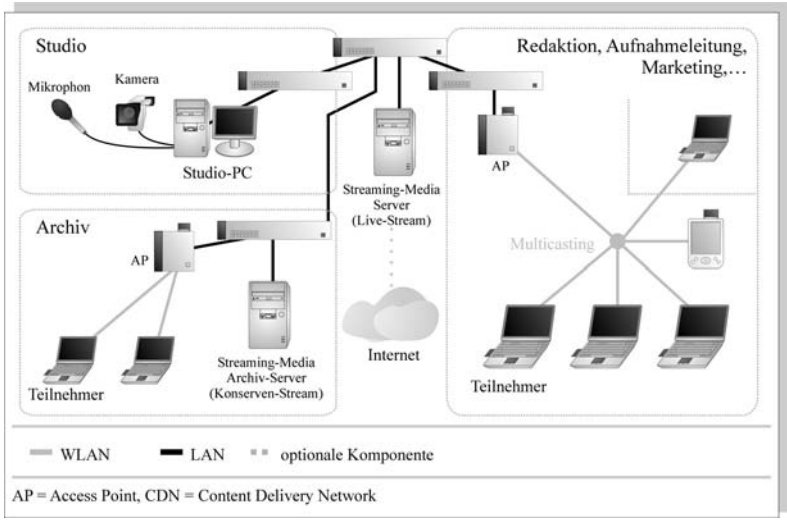

Abb. 4-4: Beispiel für Streaming-Media im WLAN: Rundfunk- oder Fernseh-Sender usw.

Das in Abbildung 4-5 gezeigte Szenario ist auch als „Distributed Classroom“ bekannt. Eine der vielen Universitäten, die an einem solchen Szenario arbeitet, ist die University of Washington innerhalb des E-Learning-Projekts Conference XP. Auf der Web-Seite [COXP] des Projekts finden sich zahlreiche Informationen zu einem solchen Szenario. In einem „Distributed Classroom“ können Studenten und Dozenten gleichermaßen mittels WLAN und Streaming-Media kooperieren.

In Abbildung 4-5 wird der Vortrag des Dozenten direkt von dessen Notebook aufgenommen und z. B. mit einem Video und seiner Sprache an den Streaming-Media-Server gesendet. Der Dozent kann dadurch den Vortrag zu Hause vorbereiten und direkt im Hörsaal wiedergeben. Der Streaming-Media-Server verteilt die Informationen daraufhin entweder direkt an die Studenten im Hörsaal oder z. B. in die Bibliothek oder die Mensa. Die Studenten können sich flexibel in das WLAN integrieren. Anstelle der Bibliothek kann auch ein Hörsaal einer entfernten Hochschule mit dem Stream beliefert werden und die Studenten dort mit Informationen versorgt werden. Auch der Dozent selbst kann seinen Vortrag an unterschiedlichen Orten halten. Die zusätzliche, optionale Archivierung des Streams auf dem Streaming-Media-Server macht z. B. das „Senden“ des

Vortrags zu einem späteren Zeitpunkt oder auf Abruf für die Studenten möglich. Dabei kann hier auch eine Authentifizierung der Studenten realisiert werden (wie im Abschnitt 3.3.2 beschrieben).

Beispiel: Distributed Classroom – E-Learning

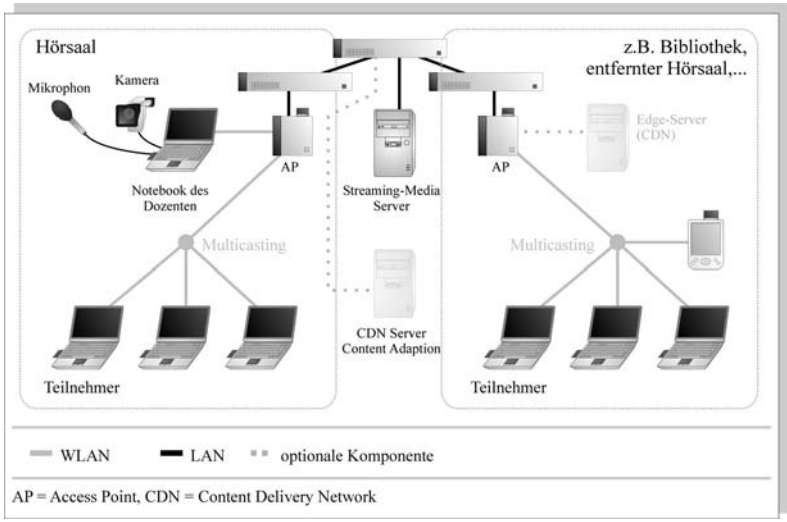

Abb. 4-5: Beispiel für Streaming-Media im WLAN: Distributed Classroom

Wächst der skizzierte „Distributed Classroom“ über mehrere Orte hinweg, so kann z. B., wie in Abbildung 4-5 anhand der grauen CDN-Server demonstriert, ein CDN (Content Delivery Network) realisiert werden, das den Vortrag des Dozenten auf mehrere Server verteilt. Am Rande des CDN und damit in unmittelbarer Nähe zum Teilnehmer könnten die Edge-Server für eine schnelle und effiziente Wiedergabe des Streams sorgen.

Das in Abbildung 4-5 gezeigte Szenario bietet eine ausgezeichnete E-Learning-Plattform.

Die in diesem Abschnitt aufgezeigten Szenarien stellen nur einige ausgewählte Beispiele dar. In der Praxis existieren viele Variationen und Kombinationen dieser Szenarien.

5. LCDNs in verteilten drahtlosen Netzwerken

Für die Verteilung von Streaming-Media an eine große Anzahl von Empfängern wurde bereits in den vorherigen Kapiteln häufig von CDNs gesprochen. Basis für dieses Kapitel ist daher die im Abschnitt 4.6 gezeigte Abbildung 4-5. Die Realisierung der in ihr gezeigten optionalen Komponenten, die ein CDN bilden, wird in diesem Kapitel fixiert. Im Zusammenhang mit dem Thema dieser Arbeit haben CDNs dabei die Aufgabe, in großen Szenarien mit mehreren drahtlosen Netzwerken (Zellen) die einzelnen Zellen mit Inhalten zu beliefern. Genauer gesagt wird durch CDNs die Last, die durch den Transport der Inhalte an verschiedene drahtlose Netzwerke entsteht, gesenkt. Innerhalb der Zellen wird der Inhalt schließlich per Multicasting, wie im Abschnitt 4.4 beschrieben, verteilt.

Durch die Nutzung von CDNs kann auf diese Art und Weise der Spagat zwischen effizienter Verteilung von Streaming-Media an mehrere Teilnehmer und Konserven-Streams erreicht werden. Multicasts können, wie im Abschnitt 4.1.3 gezeigt, nur effizient für Live-Streams eingesetzt werden, Konserven-Streams lassen sich nicht als Multicast versenden. Somit entsteht bei Konserven-Streams eine extrem hohe Bandbreitenanforderung. Während diese erhöhte Bandbreitenanforderung im Backbone von drahtgebundenen Netzwerken z. B. mit einer Bandbreite 1 GBit/s (siehe Abschnitt 4.3) erreicht werden kann, stellt sie für drahtlose Netzwerke mit geringer Bandbreite (Abschnitt 4.1.3) eine große Hürde dar. Liefert der Origin-Server jedoch die Daten zunächst an einen Edge-Server am drahtlosen Netzwerk, so kann dieser den Stream, der dort höchstwahrscheinlich an mehrere Teilnehmer gleichzeitig gesendet werden soll, effizient als Multicast an die Teilneh-

mer versenden. Daher können dank CDNs in dem aufgezeigten Szenario (z. B. in Abbildung 4-5) folgende Eigenschaften erreicht werden:

- Reduzierung der erforderlichen Bandbreite am Origin-Server (auch und gerade bei Unicasts)
- Effiziente Verteilung von Streaming-Media in drahtlosen Netzwerken (auch für Unicasts)
- Ggf. Verzicht auf aufwändige Multicast-Routing-Strukturen im Backbone
- Verminderung der erforderlichen Bandbreite im Backbone

Der Titel dieses Kapitels führt in diesem Zusammenhang einen neuen Begriff ein. Sog. LCDNs (Local CDNs), die gewissermaßen lokal im Intranet bzw. im LAN- und MAN(Metropolitan Area Network)-Bereich eingesetzt werden können. Der Begriff CDN bezieht sich dabei in der Informatik eher auf ein WAN, vor allem aber auf das Internet. CDNs wurden vor einigen Jahren als Verteilung der Web-Inhalte auf mehrere Server und für die Beschleunigung des Versands der Inhalte an den Nutzer eingerichtet. Weltweit existiert eine große Anzahl von Content-Delivery-Anbietern (CDN-Betreibern), die rund um den Globus tausende von Edge-Servern bei lokalen ISPs sowie an wichtigen Knotenpunkten des Internet installiert haben.

LCDNs stellen eine Projektion dieser weltweiten CDN-Struktur auf den LAN- bzw. MAN-Bereich dar.

Grundsätzlich entstehen LCDNs aus dem gleichen Bedürfnis wie CDNs, nämlich der Tatsache, dass insbesondere Streaming-Media-Inhalte ohne sie nicht ausreichend schnell an drahtlose Teilnehmer gesendet werden können.

Abschnitt 5.1 zeigt die Strukturen eines CDN, die auch für die in dieser Arbeit aufgezeigten LCDNs gelten. Abschnitt 5.2 bietet eine Beschreibung des Themenbereichs Content-Adaption, der vor allem durch den Zuwachs von dynamischen Inhalten auf den Edge-Servern entstanden ist und zunehmend gefordert wird.

Im Abschnitt 5.3 wird schließlich das im Abschitt 4.6 gezeigte Szenario aus Abbildung 4-5 um eine LCDN-Struktur erweitert.

5.1 CDN-Strukturen

Ein CDN (Content Delivery Network) stellt ein logisches Netz innerhalb eines Übertragungsnetzwerks dar. Das Übertragungsnetzwerk ist für ein CDN in der Regel das Internet. Innerhalb des Internet werden für ein CDN

weltweit mehrere Edge-Server installiert. Edge-Server befinden sich idealerweise bei lokalen ISPs (Internet Service Provider) oder an wichtigen Knotenpunkten.

Ein CDN bildet ein logisches Netzwerk aus zwei Server-Klassen. Die eine Klasse bilden sog. Edge-Server, die am Rande des CDN und damit i. d. R. auch am Rande des Übertragungsnetzes (z. B. dem Internet) liegen. Ein Edge-Server befindet sich somit in unmittelbarer Nähe des Clients bzw. des Endanwenders. Dadurch ergeben sich für den Nutzer deutlich höhere Durchsatzraten, kleinere Verzögerungen sowie weniger Verzögerungsabweichungen. Für den Anbieter des Inhalts sowie dessen Server ergibt sich analog eine geringere Auslastung bzw. Vermeidung von Überlastungen. Der Anbieter stellt die zweite Klasse von Servern eines CDN: die sog. Origin-Server. Ein Origin-Server stellt den Inhalt für die Teilnehmer zur Verfügung und verteilt ihn an die Edge-Server. In der Vergangenheit waren Edge-Server von ihrer Art her verteilte Caches. Heute hat sich dieses Bild gewandelt, da vermehrt auch dynamische Inhalte vor Ort am Edge-Server (personalisiert) assembliert werden und damit die erhöhte Last der dynamischen Seitenerzeugung am Origin-Server reduziert wird. Abbildung 5-1 zeigt die klassische Struktur eines CDN. Das CDN bildet dabei ein logisches Netz im Internet.

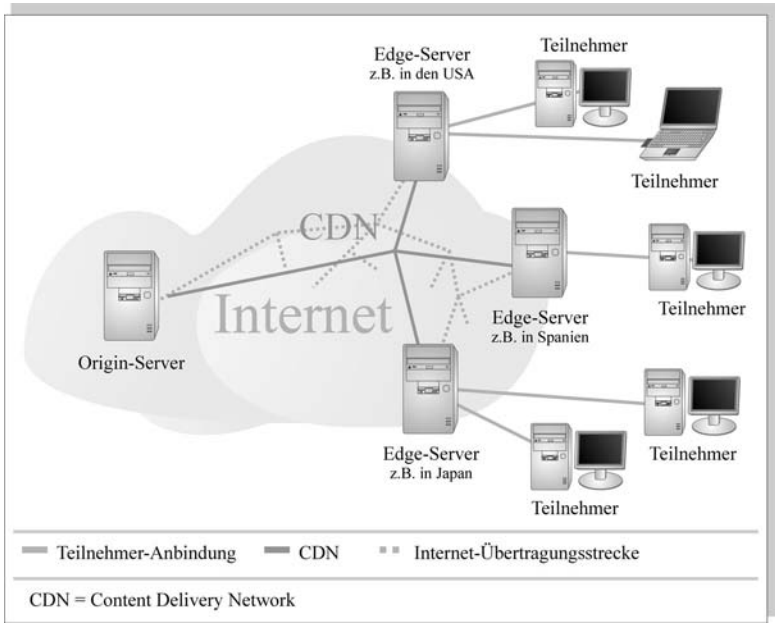

Abb. 5-1: Logisches CDN innerhalb des Internet

Die Edge-Server in Abbildung 5-1 könnten z. B. in unterschiedlichen Ländern oder Kontinenten platziert sein. Oft verwenden Content-Distributoren jedoch je nach Last mehrere Edge-Server in einem Land bzw. im näheren Umkreis. Ein weiterer Vorteil für den Teilnehmer an dieser Struktur ist, dass z. B. bei einer fehlerhaften Übertragung die erneute Anforderung der Daten nicht zunächst über viele Router (und deren Warteschlangen) an den Origin-Server weitergeleitet werden muss, sondern das Paket einfach erneut vom Edge-Server angefordert werden kann. Man spricht in diesem Zusammenhang von einer erhöhten Interaktionsgeschwindigkeit für den Teilnehmer. Dabei bezieht sich die Interaktionsgeschwindigkeit auf den Zeitraum zwischen Anforderung des Teilnehmers und der daraufhin gesendeten Antwort.

Eine detaillierte Beschreibung eines CDN findet sich z. B. bei dem Content-Distributor [AKAM].

An dieser Stelle sei erwähnt, dass ein CDN nicht zwangsläufig gesonderte Edge-Server benötigt. Der Begriff der Edge-Server wurde mehr und mehr geprägt, seitdem am Rande des CDN auch dynamische Inhalte bereitgestellt werden sollten. Ein CDN kann im eigentlichen Sinne auch aus einer Ansammlung von verteilten Cache-Servern bestehen. So bieten manche Hersteller von Streaming-Media-Software die Möglichkeit an, eigene Cache-Server für die Verteilung der Daten anzugeben. Diese Cache-Server dienen dann lediglich für die Verteilung der Last während der Content-Abfrage. Sie müssen nicht zwangsläufig näher am Teilnehmer platziert sein als der Origin-Server. Auch die Bildung eines solchen Netzes ohne explizite Edge-Server kann jedoch als CDN bezeichnet werden. Derartige CDNs können z. B. Zweigstellen und Vertretungen eines Unternehmens verbinden oder, wie in diesem Kapitel gezeigt, LCDNs bilden. Die Abbildung 5-2 zeigt ein solches LCDN.

5.1.1 Server

In Abschnitt 5.1 wurden bereits zwei Klassen von Servern in einem CDN vorgestellt. Während der Origin-Server einen Streaming-Media- bzw. Web-Server, wie z. B. in Abbildung 1-16 oder 1-15 gezeigt, darstellt, beinhaltet der Edge-Server gesonderte Funktionskomponenten. Vor einigen Jahren stellten Edge-Server komplette Kopien des Origin-Servers und seiner Netzwerkstruktur (Firewalls, Datenbank-Server usw.) dar. Solche kompletten Kopien der Server-Strukturen sind nicht nur kostspielig, sondern auch schwer zu administrieren. Daher sind heutige Edge-Server von der Grundfunktionalität her ein Cache-Server. Auf dem Cache werden Kopien einzelner Inhalte des Origin-Servers abgelegt und an die Teilnehmer verteilt.

In diesem Zusammenhang spricht man auch von einem Reverse-Proxy-Server. Ein Proxy-Server ermöglicht einem Benutzer den Zugriff auf mehrere Web-Seiten. Ein Reverse-Proxy ermöglicht mehreren Benutzern den Zugriff auf eine Web-Seite, dem Inhalt des Origin-Servers. Gängige Proxy-Server wie der Squid [SQUI] unterstützen die Konfiguration als Reverse-Proxy. Auch Web-Server wie z. B. der Apache [APAC] bieten eine Reverse-Proxy-Funktion.

Von den Edge-Servern wird jedoch heute mehr verlangt als das pure Caching von statischen Inhalten. Eine weitaus größere Last als die Verteilung von statischen Inhalten erzeugen auf den Origin-Servern die dynamischen Inhalte (z. B. personalisierte Inhalte). Um die komplette Web-Seite dynamisch zu erzeugen, wäre jedoch an jedem Edge-Server erneut die komplette Server-Struktur des Origin-Servers notwendig. Um dieses Problem zu lösen, bedienen sich Edge-Server einer sog. Content-Adaption (Anpassung des Inhalts). Dabei kann z. B. mit bestimmten Beschreibungssprachen innerhalb einer Web-Seite markiert werden, welche Teile statisch sind und daher im Cache gespeichert werden dürfen. Eine solche Beschreibungssprache sind z. B. die ESI (Edge Side Includes) [ESI]. Die Anpassung des ausgelieferten Inhalts (Content Adaption) ist z. B. mit dem Protokoll ICAP möglich. ESI und ICAP werden in den Abschnitten 5.2.2 und 5.2.1 beschrieben.

Vermehrt bieten Content-Distributoren an ihren Edge-Servern auch die Unterstützung von verteilten Web-Anwendungen. So können bei vielen Anbietern auch Java-2-Enterprise-Edition[J2EE]-Komponenten, wie Enterprise Java Beans, oder .NET [DOTN] Komponenten verwendet werden.

5.1.2 Anwendungsszenario LCDN

Wie bereits im Abschnitt 5.1 erwähnt, verteilt ein CDN im klassischen Fall den Content im Internet über mehre Länder und Kontinente hinweg. CDNs können jedoch auch im lokalen Umfeld, z. B. im LAN eines Unternehmens, eine wichtige Funktion erfüllen. In diesem Zusammenhang wurde bereits im Abschnitt 5 von sog. LCDNs (Lokalen CDNs) gesprochen. Abbildung 5-2 zeigt die Struktur eines solchen LCDN am Beispiel eines Unternehmens.

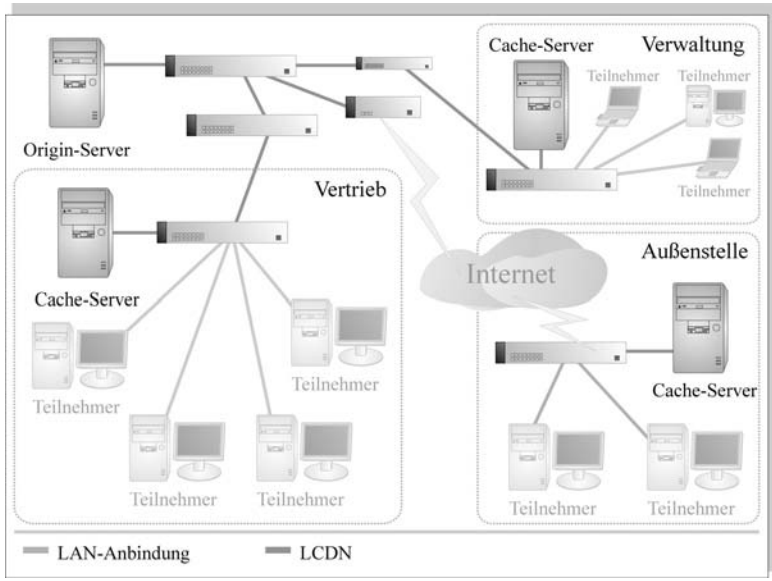

Abb. 5-2: Beispiel eines LCDN (innerhalb eines Unternehmens)

Dabei wird der Content (z. B. Streaming-Media) von einem Origin-Server in der Hauptniederlassung des Unternehmens bereitgestellt. Da z. B. im Vertrieb, der Verwaltung sowie einer Außenstelle vermehrt auf Teile dieses Content zugegriffen wird (z. B. im Vertrieb auf Werbe-Trailer von gefertigten Maschinen), soll mit diesen Stellen ein LCDN realisiert werden. Dabei müssen sich die Server, die Endpunkte des LCDN sind, in den Abteilungen nicht zwangsläufig am Rande des Übertragungsnetzes befinden. Daher wurde in Abbildung 5-2 anstelle von Edge-Server die neutrale Bezeichnung Cache-Server verwendet. Die Teilnehmer können von diesen Cache-Servern den gewünschten Content sehr viel schneller abrufen, da diese i. R. näher platziert sind als der Origin-Server. Ein weiterer Vorteil ist, dass die Auslastung des Origin-Servers, vor allem aber die Auslastung des Backbone (Switches bzw. Router für die Anbindung der Abteilungen) stark reduziert wird. Überlast-Situationen z. B. beim Bereitsstellen eines neuen Werbe-Trailers durch das Marketing werden so effizient vermieden.

Ein weiterer Vorteil offenbart sich, wenn für die Verteilung von Streaming-Media Multicasting genutzt werden soll (wie im Abschnitt 2.1.5 beschrieben). Um eine solche Verteilung in dem in Abbildung 5-2 gezeigten Szenario vom Origin-Server aus zu ermöglichen, müssten alle Router und

Switches (z. B. aufgrund von VLANs – siehe [BADA_01]) auf dem Weg zum Teilnehmer ein Multicast-Routing, wie im Abschnitt 2.2 beschrieben, unterstützen. Durch die Verwendung von Cache-Servern lässt sich von diesen aus ein Multicast an die Teilnehmer durchführen, während die Inhalte als Unicasts vom Origin-Server an die Cache-Server geschickt werden. Ein aufwändiges Multicast-Routing im Backbone kann somit vermieden werden.

Eine besondere Beachtung benötigt in Abbildung 5-2 die Außenstelle. Sie ist über eine WAN-Verbindung (über das Internet) z. B. per VPN (siehe Abschnitt 3.3.3) mit der Hauptniederlassung verbunden. WAN-Verbindungen besitzen in der Regel eine sehr viel geringere Bandbreite als z. B. LAN-Verbindungen. Durch die Verteilung des Inhalts vom Origin-Server an den Cache-Server der Außenstelle wird so für die dort angebundenen Teilnehmer ein Abruf des Inhalts überhaupt erst möglich. Mehrere Teilnehmer würden in der Außenstelle bei der Verteilung per Unicast eine viel größere Bandbreite fordern, als die WAN-Verbindung ermöglichen könnte (siehe auch Abschnitt 4.1.3). Außerdem können auf diese Weise z. B. Verbindungskosten der WAN-Verbindung gespart werden. Für einen reibungslosen Betrieb könnte der Cache-Server der Außenstelle z. B. über Nacht die aktuellen Inhalte des Origin-Servers abrufen und speichern.

5.2 Content-Adaption

Edge-Server eines CDN müssen neben dem reinen Caching der Inhalte auch eine Adaption realisieren. So müssen z. B. dynamische Web-Seiten erzeugt werden. Außerdem muss der Content auf den entsprechenden Teilnehmer angepasst werden. Zum einen können z. B. durch personalisierte Inhalte auf einer Web-Seite persönliche Links, Favoriten o. ä. eingefügt werden. Solche Personalisierungen lassen sich nicht nur am Origin-Server, sondern auch am Edge-Server durchführen. Außerdem können z. B. Inhalte für mobile Endgeräte angepasst werden (Verringerung der Auflösung eines übertragenen Videos o. ä.). Alle diese Vorgänge werden unter Content-Adaption zusammengefasst und beschreiben die Anpassung des Contents des Origin-Servers an Anforderungen der Teilnehmer durch einen Edge-Server.

In der Abbildung 4-3 im Abschnitt 4.6 wurde beispielsweise eine Anpassung des übertragenen Streams an mobile Endgeräte vorgeschlagen. Dadurch kann auch auf PDAs o. ä. eine flüssige Wiedergabe erreicht werden, die z. B. ohnehin kein hochauflösendes Video auf ihrem Display anzeigen könnten. Neben einer solchen Anpassung kann aber auch z. B. die Landessprache des Contents angepasst werden.

Für die Implementierung einer Content-Adaption existieren zwei Möglichkeiten:

- Die Adaptierung bzw. Änderung der Anfrage des Teilnehmers bzw. der Antwort des Edge-Servers auf eine Anfrage
- Die Integration von variablen Bereichen innerhalb einer Web-Seite, die bei der Auslieferung durch den Edge-Server von diesem modifiziert werden können

Die erste Möglichkeit wird z. B. durch das Protokoll ICAP implementiert. Sie wird im Abschnitt 5.2.1 beschrieben.

Die Integration variabler Bereiche in eine Web-Seite bedarf einer Ergänzung der Seitenbescheibungssprache HTML. Eine solche Erweitung stellen z. B. die ESI (Edge Side Includes) dar, die im Abschnitt 5.2.2 erläutert werden.

5.2.1 Das Protokoll ICAP

Eine genaue Beschreibung des ICAP-Protokolls würde den Rahmen dieses Abschnitts und vor allem die Bedeutung für dieses Kapitel sprengen. Daher werden in diesem Abschnitt nur die Grundzüge des Protokolls ICAP erklärt.

Das Protokoll ICAP [ICAP] ist als IETF-Internet-Draft spezifiziert. ICAP bildet ein auf Klartextnachrichten ausgerichtetes Protokoll, ähnlich dem HTTP [HTTP]. Wie auch beim HTTP wird zwischen Requests und Responses unterschieden. Ein Request wird dabei vom ICAP-Client an den ICAP Server gesendet, der diesen mit einer Response beantwortet. Der ICAP-Client ist in der Regel Teil eines Reverse Proxy und somit Teil eines Edge-Servers. Eine ICAP-Implementierung für Squid befindet sich z. B. unter [ICSQ]. Der ICAP-Server ist eine separate Funktionskomponente, die jedoch auch auf dem gleichen Rechner wie der Reverse Proxy realisiert werden kann. ICAP-Client und -Server befinden sich somit zwischen Teilnehmer und Origin-Server und sind in der Lage, sowohl Anfragen an den Origin-Server als auch dessen Antworten darauf zu modifizieren.

Für diesen Zweck bietet ICAP zwei Methoden:

- REQMOD
- RESPMOD

Die REQMOD(Request Modification)-Methode modifiziert den Request eines Teilnehmers auf dem Weg zum Origin- bzw. Edge-Server. Ruft ein Teilnehmer beispielsweise das Video start.mov auf, so kann z. B. anhand seines Request-Headers die Version seines Internet-Browsers und das ver-

wendete Betriebssystem erkannt werden. Wird bei dieser Überprüfung, die durch den ICAP-Server durchgeführt wird, z. B. festgestellt, dass es sich um einen PDA handelt, so könnte der ICAP-Server den Request z. B. modifizieren, indem start-lowres.mov (als Video mit geringerer Auflösung) vom Edge- bzw. Origin-Server abgefragt wird.

Die RESPMOD(Response Modification)-Methode modifiziert die Antwort des Edge- bzw. Origin-Servers auf den Request eines Teilnehmers. Dabei wird die Antwort des Edge- bzw. Origin-Servers an den ICAP-Client im Reverse Proxy geleitet. Der ICAP-Client schickt die Antwort an den ICAP-Server. Dieser kann nun in die Antwort z. B. ein personalisiertes Menü, den Namen des Teilnehmers oder das aktuelle Datum o. ä. einfügen. Die RESPMOD-Methode kann unter Umständen einen hohen Datentransfer verursachen, sofern ICAP-Server und ICAP-Client nicht auf dem gleichen Rechner installiert sind. In diesem Fall müssen alle Antworten, die modifiziert werden, komplett vom ICAP-Client an den -Server sowie zurückgesendet werden.

Im Zusammenhang mit dem Protokoll ICAP werden auch häufig OPES (Open Pluggable Edge Services) verwendet [OPES]. Diese Services beziehen sich auf die Modifikation des direkten Datenflusses zwischen Teilnehmer und Origin-Server. Sog. OPES-Prozessoren, die die Datenströme modifizieren, können sich dabei sowohl beim Empfänger (bzw. dessen ISP) sowie dem Anbieter selbst befinden. OPES können anhand von Regeln die versendeten HTTP-Nachrichten ändern. Eine ausführliche Beschreibung der OPES findet sich in [OPES].

ICAP und OPES werden teilweise in direkte Konkurrenz zu dem von Microsoft im .NET-Framework proklamierten Protokoll SOAP (Simple Object Access Protocol) [SOAP] sowie UDDI (Universal Description Discovery and Integration) [UDDI] gesetzt. Für einen Vergleich der beiden Verfahren sei auf die Lektüre der entsprechenden Verweise verwiesen.

5.2.2 Beschreibungssprache ESI

Ein gänzlich anderer Ansatz als die Modifikation des Inhalts durch einen separaten Server, wie z. B. im Abschnitt 5.2.1 beschrieben, ist die Einbettung bestimmter Modifikationsinformationen in den Inhalt selbst. Für diesen Zweck existieren einige Erweiterungen zu der Seitenbeschreibungssprache HTML. Ein Beispiel sind die ESI (Edge Side Includes) [ESI], die auch als Draft beim W3C (World Wide Web Consortium) [W3C] vorliegen. ESI beinhalten eine geringe Anzahl von XML-Tags [XML] ähnlich den Tags der

HTML. Sie wurden federführend von Akamai [AKAM], einem der größten Content-Distributoren, eingeführt.

Kurz beschrieben, können durch ESI bedingte dynamische Ergänzungen des übertragenen Inhalts durchgeführt werden. So kann z. B. ein bestimmter Teil einer Seite durch ESI als dynamisch markiert werden, so dass der Edge-Server diesen Teil des Contents direkt vom Origin-Server bezieht und nicht in seinem Cache ablegt. Dadurch können auch dynamische Seiten teilweise im Edge-Server assembliert werden, ohne diese komplett vom Origin-Server abzurufen. Wie bei den OPES können auch ESI durch einen gesonderten ESI-Prozessor verarbeitet werden, der daraus dynamisch einen Ausgabestrom erzeugt, der in den transportierten Inhalt vor der Auslieferung einfließt. Eine genaue Beschreibung der ESI findet sich unter [ESI].

5.3 Anwendungsszenario „LCDNs“ im WLAN-Verbund

Die Abbildung 5-3 zeigt die Erweiterung des in Abbildung 4-5 im Abschnitt 4.6 gezeigten „Distributed Classrooms“. Dabei wurden an den einzelnen Stellen, an denen eine Verteilung des Streams erfolgen soll, zusätzliche Cache-Server installiert. Diese Cache-Server stehen über ein LCDN in Verbindung mit dem Origin-Server, der weiterhin die Streaming-Media-Inhalte bereitstellt. Durch die gezeigte Struktur lässt sich z. B. auf die Unterstützung des Multicast-Routing (Abschnitt 2.2) im Backbone der Universitäten verzichten. Gerade auf der Verbindungsstrecke zum „entfernten Hörsaal“ kann dies eine wichtige Anforderung sein, da viele Teilstrecken im Internet kein solches Multicast-Routing unterstützen.

Ein weiterer Vorteil ist die effizientere Verteilung von Unicasts, die z. B. für Konserven-Streams unvermeidbar sind. So kann z. B. der Stream einer Vorlesung per Unicast von dem Origin-Server an alle Cache-Server übertragen werden, die dann per Multicast in ihrem Netz den Stream an die Teilnehmer verteilen. Insbesondere der Engpass „maximale Teilnehmeranzahl bei Unicasts im WLAN“, wie im Abschnitt 4.1.3 beschrieben, ließe sich damit entschärfen. Für eine weitere Reduzierung der Belastung des Backbone durch den Stream könnte die Vorlesung z. B. auf dem Origin-Server archiviert werden und zu einem späteren Zeitpunkt (z. B. in der Nacht) an den Cache-Server der Bibliothek übertragen werden, der dann zu einem späteren Zeitpunkt die Vorlesung, z. B. am darauf folgenden Tag, wiedergibt. Diese Option kann unter Umständen auch für den entfernten Hörsaal interessant werden, da dieser ggf. mit einer geringen Bandbreite mit dem Internet verbunden ist. In diesem Fall würde jedoch eine mögliche Interaktion der dort verbundenen Studenten und dem Dozenten im Hörsaal verhindert werden,

da die Wiedergabe nicht synchron erfolgen würde. Dies kann einen erheblichen Nachteil bedeuten.

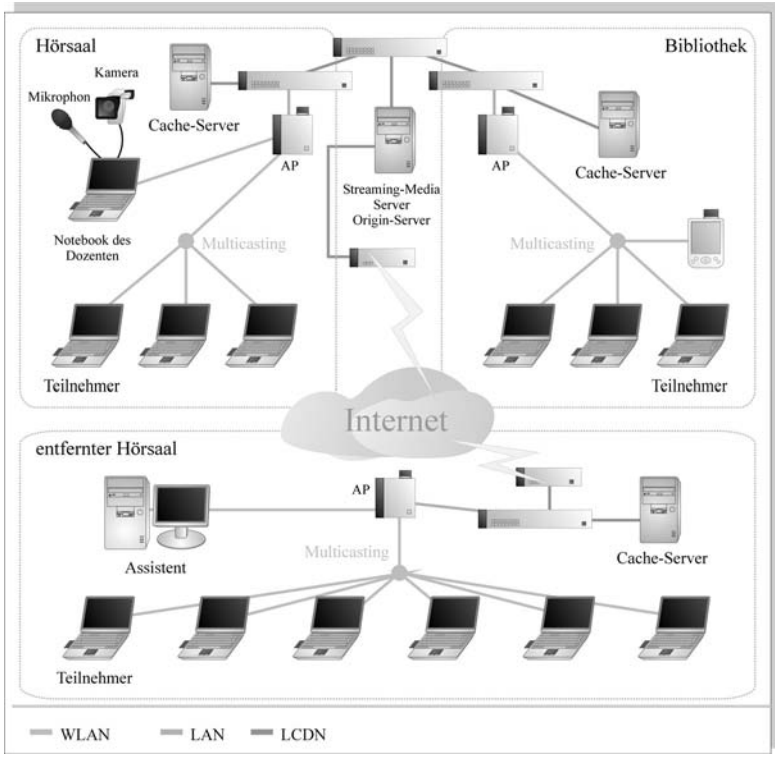

Abb. 5-3: Beispiel für ein LCDN im „Distributed Classroom“

Für eine ideale Anpassung an die jeweiligen Teilnehmer könnten die Cache-Server in Abbildung 5-3 auch, wie im Abschnitt 5.2 kurz beschrieben, eine Content-Adaption vornehmen. Dabei könnten z. B. Teilnehmer mit einem PDA automatisch einen Stream zugewiesen bekommen, der auf die Größe des Displays des PDA angepasst ist (siehe Abschnitt 5.2.1). Auf diese Weise könnten sogar Teilnehmer mit anderen Landessprachen individuell der Vorlesung in ihrer Sprache folgen.

Die Übertragung des Streams wird dabei für alle Teilnehmer bedingt durch die Cache-Server, die sich im gleichen Netz und somit in unmittelbarer Nähe zu ihnen befinden, deutlich beschleunigt. Auch die Kommunikation z. B. des Kontrollprotokolls (siehe Abschnitt 1.6) und der Transportkontrolle (siehe Abschnitt 1.6.5) würde zwischen Cache-Server und Teilnehmer erfolgen

und somit keine Last im Backbone verursachen. Die Beschleunigung der Wiedergabe des Streams wirkt sich dabei positiv auf die QoS-Eigenschaften (Abschnitt 4.2) Verzögerung, Verzögerungsabweichung sowie Bitrate aus.

Der entscheidende Vorteil der in Abbildung 5-3 gezeigten Streaming-Media-Struktur ist ihre Skalierbarkeit. Sie bietet auch für eine große Anzahl von Teilnehmern sowohl per Unicast als auch per Multicast genug Flexibilität für deren Versorgung. Es könnten, wie in einem Baukasten, Elemente wie in Abbildung 5-3 gezeigt (Hörsaal, entfernter Hörsaal, Bibliothek…) hinzugefügt werden. Allerdings sollte bei einer hohen Anzahl von Cache-Servern eine Lastverteilung (load balancing) durchgeführt werden, indem vor dem Origin-Server selbst einige Cache-Server eingerichtet werden, die die Anfragen verteilen und so einer Überlastung des Origin-Servers vorbeugen.

Die in Abbildung 5-3 gezeigte Struktur eines LCDN muss nicht zwangsweise über gesonderte Edge-Server erreicht werden. Viele Streaming-Media-Server bieten die Möglichkeit, andere Streaming-Media-Server als Cache zu verwenden. Somit wäre z. B. für den entfernten Hörsaal lediglich eine Partnerschaft (Cache-Verbund) mit dem Origin-Server notwendig, sofern dort ein separater Streaming-Media-Server existiert (z. B. für die Streams der eigenen Vorlesungen).

Für die Realisierung des in Abbildung 5-3 gezeigten Szenarios können z. B. die folgenden Komponenten verwendet werden:

Origin-Server

- Real Networks – Helix Universal Server [REAL]
- oder Microsoft Windows Media Services – Server (integriert im Microsoft .NET Server)

Cache-Server

- analog zum Origin-Server mit Referenzierung als Cache (somit ebenfalls Helix oder Windows Media)
- oder Squid [SQUID]
- ggf. ICAP Unterstützung wie z. B. in [ICSQ] verfügbar

Access Points

- Cisco 340, 350 oder 1200 Access Point (802.1X-Unterstützung, Multicast-Unterstützung)
- oder z. B. Avaya AP-2000 Access Point (802.1X-Unterstützung, Multicast-Unterstützung)

Teilnehmer

- RealOne Player [REAL]
- oder Windows Media Player (in Microsoft Windows enthalten)
- ggf. separater 802.1X-Client (ist z. B. in Microsoft Windows ab XP enthalten)

Nachtrag

Der Bereich der drahtlosen Netzwerke ist ständig im Wandel. Neue Technologien und vereinzelt bereits Geschäftsideen rund um die sog. Hot Spots, die an öffentlichen Plätzen einen freien, häufig kostenlosen Internet-Zugang bieten, sorgen nahezu täglich für Neuheiten auf diesem Gebiet. Dieser Abschnitt soll daher einige Ergänzungen zu den nachfolgenden Kapiteln bieten, die zum Zeitpunkt der Erstellung der Diplomarbeit noch nicht vorlagen. Die Ergänzungen betreffen in erster Linie das Kapitel 3 „Drahtlose Netzwerk-Technologien".

Neue drahtlose Übertragungstechniken

Seit Juni 2003 steht mit dem 802.11g-Standard eine neue Übertragungstechnik mit einer maximalen Bruttodatenrate von 54 MBit/s im 2,4-GHz-ISM(Industrial Scientific Medical)-Band (siehe Abschnitt 3.2.1) zur Verfügung. 802.11g nutzt dabei das im Abschnitt 3.2.1 für 802.11a im 5-GHz-Bereich beschriebene Orthogonal Frequency Division Multiplexing (OFDM) im 2,4-GHz-ISM-Band. Für den Idealfall sieht der 802.11g-Standard OFDM als ausschließliche Übertragungstechnik vor. Zusätzlich unterstützt 802.11g jedoch auch das Complementary Codes Keying (das im Abschnitt 3.2.1 beschrieben wird). Dadurch wird die Implementierung von 802.11g-Geräten möglich, die abwärtskompatibel zum 802.11b-Standard sind. Für diesen Fall definiert der 802.11g-Standard zwei optionale Verfahren, bei denen der Header bzw. die Präambel jedes übertragenen Paketes mittels CCK kodiert wird. Dies ermöglicht es, die vom

802.11b im Abschnitt 3.2.1 beschriebenen Zugriffsverfahren zu verwenden und damit 802.11b-Clients in eine 802.11g-Zelle zu integrieren.

Nach der Übertragung der Präambel bzw. des Headers wird beim CCK/OFDM-Verfahren die eigentliche Nutzlast des Paketes mittels OFDM versendet. Das zweite optionale Verfahren ist Packet Binary Convolutional Coding (PBCC), das die Nutzlast mit einem wesentlich komplexeren Verfahren nach der 8PSK (Eight Phase Shift Keying, vgl. QPSK Abschnitt 3.2.1) überträgt. Abhängig von den Empfangseigenschaften z. B. in Verbindung zur Entfernung zum Sender werden wie bei 802.11a und 802.11b gröbere Modulationen mit einer entsprechend niedrigeren Bruttodatenrate verwendet.

Die Abgrenzung von 802.11g zu 802.11a bzw. deren Akzeptanz wird die nahe Zukunft von drahtlosen Netzwerken bestimmen. Während 802.11g die Abwärtskompatibilität gewährleistet, wurden für 802.11a u. a. in den USA weitaus größere Frequenzbereiche mit entsprechend mehr Kanälen im 5-GHz-Bereich reserviert, die zudem nicht wie im ISM-Band mit anderen Anwendungen wie drahtlosen Videoübertragungsgeräten, Bluetooth usw. geteilt werden müssen. Für Europa wird hier der 802.11h-Standard erwartet, der eine adaptive Leistungsregelung sowie dynamische Frequenzwahl im 5-GHz-Bereich ermöglicht.

Am Horizont der drahtlosen Übertragungstechniken zeichnen sich mittlerweile Verfahren ab, die Bruttodatenraten über 100 MBit/s erreichen. Hier ist insbesondere der 802.11n-Standard zu nennen.

Auch der Bluetooth-Standard soll noch 2003 in der erweiterten Version 1.2 erscheinen. Dabei ist als zentraler Vorteil des neuen Standards das Adaptive Frequency Hopping (AFH) zu nennen. Während der aktuelle, im Abschnitt 3.2.2 beschriebene Standard mittels FHSS über das gesamte Frequenzband springt, wird beim AFH der am wenigsten belegte Frequenzbereich genutzt. Dadurch wird eine Koexistenz von Bluetooth mit anderen drahtlosen Übertragungstechniken im 2,4-GHz-ISM-Band hinsichtlich der möglichen Bruttodatenrate für beide Techniken effektiver. Neben der Verbesserung der Anwendbarkeit von Bluetooth-Geräten sieht der Standard außerdem das sog. Enhanced Voice Processing vor, das für eine Verbesserung der Quality-of-Service-Eigenschaften insbesondere bei der Sprachübertragung eingesetzt werden soll. Hierbei wird durch zusätzliche Fehlerkorrekturverfahren, wie dem im Abschnitt 4.1 beschriebenen Forward Error Correction, die Fehlerrate (siehe Abschnitt 1.1.2) verringert.

Am Rande ist zu bemerken, dass sich die im Abschnitt 3.2.3 noch genannte HomeRF-Arbeitsgruppe mittlerweile endgültig aufgelöst hat. Wie im

Abschnitt 3.2.3 bereits vermerkt wurde, hat die Relevanz von HomeRF als drahtlose Übertragungstechnik in den letzten Jahren rapide abgenommen.

Neue Techniken für die Sicherheit drahtloser Netzwerke

Als Nachfolger der im Abschnitt 3.3.1 beschriebenen WEP-Verschlüsselung, wurde der Wi-Fi-Protected-Access(WPA)-Standard eingeführt. Er implementiert gewissermaßen vorab einige Sicherheitsfunktionen des noch nicht ratifizierten 802.11i-Sicherheitsstandards für drahtlose Netzwerke. Letzterer bietet u. a. neue Verschlüsselungsverfahren wie den Advanced Encryption Standard (AES). WPA erhöht die Sicherheit von drahtlosen Übertragungen u. a. durch die Erweiterung des Initialisierungsvektors zur Behebung seiner im Abschnitt 3.3.1 beschriebenen Schwachstellen. Außerdem unterstützt es eine Überprüfung der Integrität eines empfangenen Paketes über zugehörige Hash-Werte, um die Manipulation von verschlüsselten Paketen durch Dritte zu verhindern. Für die Integration in bestehende IT-Sicherheitsstrukturen implementiert der WPA-Standard außerdem den bereits im Abschnitt 3.3.2 genannten Standard 802.1X.

Allerdings zeigt auch der WPA-Standard bereits einige Schwächen. Die Integritätsprüfung von Paketen beim WPA erzeugt eine hohe Rechenlast. Da Access Points meist nur mit einem vergleichsweise schwachen Prozessor ausgestattet sind, können Sie bei einer hohe Rate an eingehenden Paketen derartig überlastet werden, dass sie schließlich den Dienst verweigern. Man spricht in diesem Zusammenhang auch von einem Denial-of-Service. Die sicherste Technik für den Einsatz von drahtlosen Netzwerken dürfte somit auch in Zukunft ein im Abschnitt 3.3.3 skizziertes VPN sein. Da dieses jedoch aufgrund seiner Eigenschaft als Menge von Point-to-Point-Verbindungen (resp. Unicast-Verbindungen) keine effizienten Multicasts unterstützt, sollte für Streaming-Media nach wie vor auf die im Abschnitt 3.3.2 aufgezeigte Technik 802.1X zurückgegriffen werden.

A - Abbildungsverzeichnis

B - Tabellenverzeichnis

C - Literaturverzeichnis

[802.11] IEEE 802.11, 1999 Edition (ISO/IEC 8802-11: 1999)
http://standards.ieee.org/getieee802

[802.1X] IEEE 802.1X-2001
IEEE Standards for Local and Metropolitan Area Networks: Port-Based Network Access Control
http://standards.ieee.org/getieee802

[802.15] IEEE 802.15.1-2002
IEEE Standard for information technology – Telecommunication and information exchange between systems - LAN/MAN - Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Wireless Personal Area Networks™ (WPANs™)
http://standards.ieee.org

[ABAM] Associativity Based Multicast
http://www.online.kth.se/courses/common/adhoc/newcontent/7_6.html

[AES] Advanced Encryption Standard
http://csrc.nist.gov/encryption/aes

[AKA] J. Arkko, H. Haverinen
Internet Draft - EAP AKA Authentication, 2002
ftp://ftp.isi.edu/in-notes/internet-drafts/draft-arkko-pppext-eap-aka-05.txt

[AKAM] Akamai
http://www.akamai.de

[APAC] Apache Web Server / Foundation
http://www.apache.org

[BBSZ_00] J. Böhringer, P. Bühler, P. Schlaich, H.-J. Ziegler
Kompendium der Mediengestaltung für Digital- und Printmedien
Springer, 2000

[BADA_01] Anatol Badach, Erwin Hoffmann
Technik der IP-Netze
Hanser, 2001

[BT] Bluetooth
www.bluetooth.com

[BTSIG] Bluetooth Special Interest Group Specification
http://www.bluetooth.com/dev/specifications.asp

[COXP] Microsoft Research - conferencing experience project
http://www.conferencexp.net/community

[CT_01] Praxis Test WLAN
Fachzeitschrift c't 14/2002, Seite 84
Heise Verlag

[DOTN] Microsoft .NET Framework
http://www.microsoft.com/net

[ESI] Edge Side Includes
http://www.w3.org/TR/esi-lang
http://www.esi.org

[GGS] B. Aboba, D. Simon
Internet Draft - EAP GSS Authentication Protocol, 2002
ftp://ftp.isi.edu/in-notes/internet-drafts/draft-aboba-pppext-eapgss-12.txt

[HAKA_71] S. L. Hakami
Steiner's problem in praphs and its implications
Networks, Band 1, 1971

[HTTP] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee
Hypertext Transport Protocol - RFC 2616
ftp://ftp.isi.edu/in-notes/rfc2616.txt

[ICAP] Internet Content Adaption Protocol
http://www.i-cap.org

[ICSQ] Squid ICAP Client
http://icap-server.sourceforge.net/squid.html

[IKE] Internet Key Exchange - RFC 2409
http://www.ietf.org/rfc/rfc2409.txt

[IPSEC] IP Security Protocol
http://www.ietf.org/html.charters/ipsec-charter.html

[J2EE] Sun Microsystems, Inc.
Java 2 Enterprise Edition
http://java.sun.com/j2ee/

[KURO_02] James F. Kurose, Keith W. Ross
Computernetzwerke – Ein Top-Down-Ansatz mit Schwerpunkt Internet
Addison-Wesley, 2002

[LEAP] Cisco LEAP (Lightweight EAP)
http://lists.cistron.nl/pipermail/cistron-radius/2001-September/002042.html

[MBONE] Multicast Backbone on the Internet (MBone in Deutschland)
http://www.mbone.de

[MPPE] Microsoft Point-To-Point Encryption Protocol
http://www.ietf.org/internet-drafts/draft-ietf-pppext-mppe-keys-03.txt
http://www.ietf.org/rfc/rfc3078.txt

[MSASF] Microsoft Advanced Systems / Streaming Format
http://www.microsoft.com/windows/windowsmedia/WM7/format/asfspec11300e.asp

[NMG_01] Edgar Nett, Michael Mock, Martin Gergeleit
Das drahtlose Ethernet
Addison-Wesley, 2001

[OGGV] Ogg Vorbis Codec
http://www.vorbis.org

[OPES] Open Pluggable Edge Services
http://www.ietf-opes.org

[PEAP] G. Zorn, D. Simon, A. Palekar, S.Josefsson, H. Andersson
Internet Draft - Protected EAP Protocol (PEAP), 2002
ftp://ftp.isi.edu/in-notes/internet-drafts/draft-josefsson-pppext-eap-tls-eap-05.txt

[QUIC] Apple Quicktime
http://www.apple.com/quicktime

[REAL] Real Networks
http://www.realnetworks.com

[RFC_1075] D. Waitzman, C. Partridge, S.E. Deering
Distance Vector Multicast Routing Protocol
Status: experimental, 1988, ftp://ftp.isi.edu/in-notes/rfc1075.txt

[RFC_1321] R. Rivest
The MD5 Message-Digest Algorithm
Status: informational, 1992, ftp://ftp.isi.edu/in-notes/rfc1321.txt

[RFC_1584] J. Moy
Multicast Extensions to OSPF
Status: proposed standard, 1994, ftp://ftp.isi.edu/in-notes/rfc1584.txt

[RFC_1700] J. Reynolds, J. Postel
Assigned Numbers
Status: historic, 1994, ftp://ftp.isi.edu/in-notes/rfc1700.txt

[RFC_1738] T. Berners-Lee, L. Masinter, M. McCahill
Uniform Resource Locators (URL)
Status: proposed standard, 1994, ftp://ftp.isi.edu/in-notes/rfc1738.txt

[RFC_1889] Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson
RTP: A Transport Protocol for Real-Time Applications
Status: proposed standard, 1996, ftp://ftp.isi.edu/in-notes/rfc1889.txt

[RFC_2189] A. Ballardie
Core Based Trees (CBT version 2) Multicast Routing -- Protocol Specification –
Status: experimental, 1997, ftp://ftp.isi.edu/in-notes/rfc2189.txt

[RFC_2201] A. Ballardie
Core Based Trees (CBT) Multicast Routing Architecture
Status: experimental, 1997, ftp://ftp.isi.edu/in-notes/rfc2201.txt

[RFC_2236] W. Fenner
Internet Group Management Protocol, Version 2
Status: proposed standard, 1997, ftp://ftp.isi.edu/in-notes/rfc2236.txt

[RFC_2326] H. Schulzrinne, A. Rao, R. Lanphier
Real Time Streaming Protocol (RTSP)
Status: proposed standard, 1998, ftp://ftp.isi.edu/in-notes/rfc2326.txt

[RFC_2327] M. Handley, V. Jacobson
SDP: Session Description Protocol
Status: proposed standard, 1998, ftp://ftp.isi.edu/in-notes/rfc2326.txt

[RFC_2362] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, L. Wei
Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification
Status: experimental, 1998, ftp://ftp.isi.edu/in-notes/rfc2362.txt

[RFC_2716] B. Aboba, D. Simon
PPP EAP TLS Authentication Protocol
Status: experimental, 1999, ftp://ftp.isi.edu/in-notes/rfc2716.txt

[RFC_3232] J. Reynolds, Ed.
Assigned Numbers: RFC 1700 is Replaced by an On-line Database
Status: informational, 2002, ftp://ftp.isi.edu/in-notes/rfc3232.txt

[SA1X] Arunesh Mishra, William Arbaugh
An Initial Security Analysis of the 802.1X Standard
2002, http://www.cs.umd.edu/~waa/1x.pdf

[SIKO_01] Axel Sikorra
Wireless LAN Protokolle und Anwendungen
Addison-Wesley, 2001

[SMPTE] Society of Motion Picture and Television Engineers
http://www.smpte.org/smpte_store/standards

[SOAP] Simple Object Access Protocol
http://www.w3.org/TR/SOAP/

[SQUI] Squid Web Proxy Cache
http://www.squid.org

[SSH] Open Secure Shell
http://www.openssh.org

[SSL] Open Secure Socket Layer
http://www.openssl.org

[TANE_00] Andrew S. Tanenbaum
Computernetzwerke, 3. Auflage
Addison-Wesley, 2000

[TGI] Task Group I – 802.11i
http://grouper.ieee.org/groups/802/11/index.html

[TLS] Transport Layer Security
http://www.ietf.org/html.charters/tls-charter.html

[UDDI] Universal Description, Discovery and Integration
http://www.uddi.org

[W3C] World Wide Web Consortium
http://www.w3c.org

[WIZI_99] Ralph Wittmann, Martina Zitterbart
Multicast – Protokolle und Anwendungen
dpunkt.verlag, 1999

[WMA] Microsoft Windows Media Audio
http://www.microsoft.com/windows/windowsmedia

[WMV] Microsoft Windows Media Video
http://www.microsoft.com/windows/windowsmedia

[WWW_80] D. Wall
Mechanisms for Broadcast and Selective Broadcast
PhD Dissertation, Stanford University, 1980

W. Waung
Wireless Mobile Data Networking – The CDPD Approach
Wireless Data Forum, 1998
http://www2.wirelessdata.org/public/whatis/whatis.html

B. M. Waxman
Routing of multipoint connections
IEEE Journal on Selected Areas in Communications, Band 6, Nr. 9, 1988

[XML] Extensible Markup Language
http://www.w3.org/XML

In der Reihe GWDG-Berichte sind zuletzt erschienen:

Nähere Informationen finden Sie im Internet unter
`http://www.gwdg.de/forschung/publikationen/gwdg-berichte`

Nr. 40 *Plesser, Theo und Peter Wittenburg* (Hrsg.):
Forschung und wissenschaftliches Rechnen - Beiträge zum Heinz-Billing-Preis 1994
1995

Nr. 41 *Brinkmeier, Fritz* (Hrsg.):
Rechner, Netze, Spezialisten. Vom Maschinenzentrum zum Kompetenzzentrum - Vorträge des Kolloquiums zum 25jährigen Bestehen der GWDG
1996

Nr. 42 *Plesser, Theo und Peter Wittenburg* (Hrsg.):
Forschung und wissenschaftliches Rechnen - Beiträge zum Heinz-Billing-Preis 1995
1996

Nr. 43 *Wall, Dieter* (Hrsg.):
Kostenrechnung im wissenschaftlichen Rechenzentrum - Das Göttinger Modell
1996

Nr. 44 *Plesser, Theo und Peter Wittenburg* (Hrsg.):
Forschung und wissenschaftliches Rechnen - Beiträge zum Heinz-Billing-Preis 1996
1997

Nr. 45 *Koke, Hartmut und Engelbert Ziegler* (Hrsg.):
13. DV-Treffen der Max-Planck-Institute - 21.-22. November 1996 in Göttingen
1997

Nr. 46 **Jahresberichte 1994 bis 1996**
1997

Nr. 47 *Heuer, Konrad, Eberhard Mönkeberg und Ulrich Schwardmann:*
Server-Betrieb mit Standard-PC-Hardware unter freien UNIX-Betriebssystemen
1998

Nr. 48 *Haan, Oswald* (Hrsg.):
Göttinger Informatik Kolloquium - Vorträge aus den Jahren 1996/97
1998

Nr. 49 *Koke, Hartmut und Engelbert Ziegler* (Hrsg.):
IT-Infrastruktur im wissenschaftlichen Umfeld - 14. DV-Treffen der Max-Planck-Institute, 20. - 21. November 1997 in Göttingen
1998

Nr. 50 *Gerling, Rainer W.* (Hrsg.):
Datenschutz und neue Medien - Datenschutzschulung am 25./26. Mai 1998
1998

Nr. 51 *Plesser, Theo und Peter Wittenburg* (Hrsg.):
Forschung und wissenschaftliches Rechnen - Beiträge zum Heinz-Billing-Preis 1997
1998

Nr. 52 *Heinzel, Stefan und Theo Plesser* (Hrsg.):
Forschung und wissenschaftliches Rechnen - Beiträge zum Heinz-Billing-Preis 1998
1999

Nr. 53 *Kaspar, Friedbert und Hans-Ulrich Zimmermann* (Hrsg.):
Internet- und Intranet-Technologien in der wissenschaftlichen Datenverarbeitung - 15. DV-Treffen der Max-Planck-Institute, 18. - 20. November 1998 in Göttingen
1999

Nr. 54 *Hayd, Helmut und Theo Plesser* (Hrsg.):
Forschung und wissenschaftliches Rechnen - Beiträge zum Heinz-Billing-Preis 1999
2000

Nr. 55 *Kaspar, Friedbert und Hans-Ulrich Zimmermann* (Hrsg.):
Neue Technologien zur Nutzung von Netzdiensten - 16. DV-Treffen der Max-Planck-Institute, 17. - 19. November 1999 in Göttingen
2000

Nr. 56 *Plesser, Theo und Helmut Hayd* (Hrsg.):
Forschung und wissenschaftliches Rechnen - Beiträge zum Heinz-Billing-Preis 2000
2001

Nr. 57 *Hayd, Helmut und Rainer Kleinrensing* (Hrsg.):
17. und 18. DV-Treffen der Max-Planck-Institute
22. - 24. November 2000, 21. - 23. November 2001 in Göttingen
2002

Nr. 58 *Macho, Volker und Theo Plesser* (Hrsg.):
Forschung und wissenschaftliches Rechnen - Beiträge zum Heinz-Billing-Preis 2001
2003

Nr. 59 *Suchodoletz, Dirk von*:
Effizienter Betrieb großer Rechnerpools - Implementierung am Beispiel des Studierendennetzes an der Universität Göttingen
2003

Nr. 60 *Haan, Oswald (Hrsg.)*:
Erfahrungen mit den IBM-Parallelrechnersystemen RS/6000 SP und pSeries690
2003

Nr. 61 *Rieger, Sebastian*
Streaming-Media und Multicasting in drahtlosen Netzwerken - Untersuchung von Realisierungs- und Anwendungsmöglichkeiten
2003