Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung
Fachgebiet Telematik/Rechnernetze

Hauptseminar
Sommersemester 2004
zum Thema:

# RTP/RTCP

## Vorstellung einer Protokollsuite für Multimedia-Streaming

| | |
|---|---|
| von | Lars Schünzel |
| Matrikelnummer | 25046 |
| Studiengang | Informatik |
| Betreuer | Dipl.-Inf. Thorsten Strufe |

# Inhaltsverzeichnis

# 1. Einleitung

## 1.1. Gegenstand der Arbeit

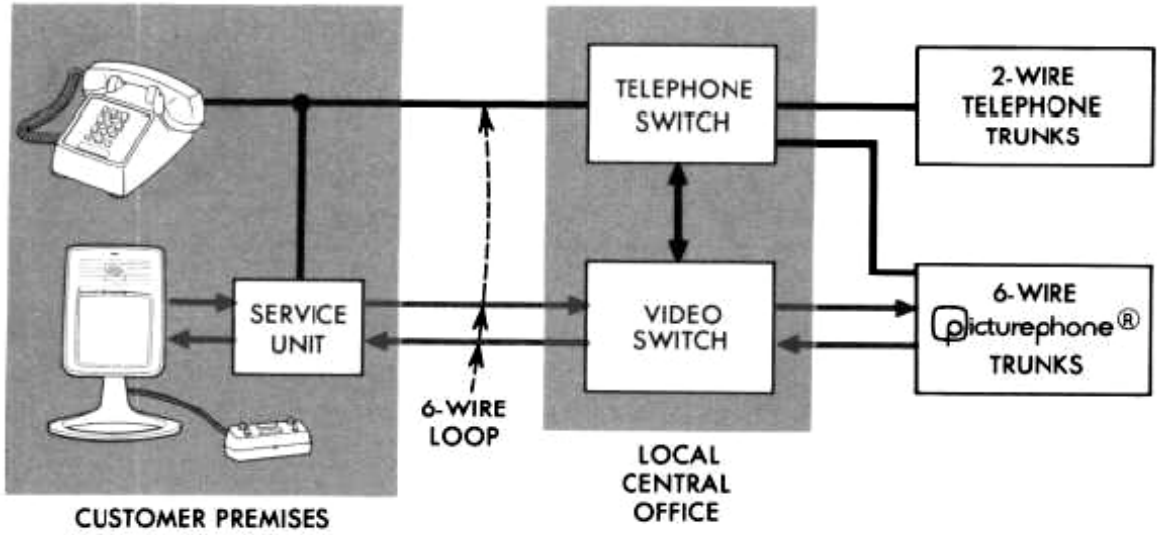
*Ziel dieser Arbeit ist es, die für Multimedia-Streaming geeignete RTP/RTCP Protokollsuite vorzustellen. Hauptaugenmerk soll dabei auf Besonderheiten im Design des Protokolls und auf Eigenschaften liegen, die in vergleichbaren Systemen nicht zu finden sind.*

## 1.2. Motivation: Wie Alles begann

In der Zeitschrift „Record“ der Bell Laboratories in der Ausgabe Mai/Juni 1969 kündigt Bell Systems einen neuen Telekomunikationsdienst an: Picturephone. „Picturephone Service - A New Way of Communicating“ titelt Julius P. Molnar, Executive Vice President, Bell Telephone Laboratories seinen einführenden Artikel über die ersten praktischen Gehversuche der Video-Telefonie in der Sonderausgabe der „Record“.

*„Rarely does an individual or an organization have an opportunity to create something of broad utility that will enrich the daily lives of everybody. Alexander Graham Bell with his invention of the telephone in 1876, and the various people who subsequently developed it for general use, perceived such an opportunity and exploited it for the great benefit of society. Today there stands before us an opportunity of equal magnitude- PICTUREPHONE® service.“ [10]*

Hinter dem Picturephone steckt eine einfache und logische Erweiterung des bestehenden Telefonnetzes. Mit Hilfe von zwei zusätzlichen Aderpaaren kann eine




Videokonferenz  mit 30 Bildern pro Sekunde übertragen werden.

Bell ging davon aus, dass sich bis Mitte der 80er Jahre ungefähr 3 Millionen Picturephones in Büros und Haushalten befinden würden. Die geschätzten Einnahmen beliefen sich auf 5 Milliarden US-Dollar pro Jahr.
Die anfänglichen Reaktionen fielen sehr positiv aus. Die Euphorie verebbte jedoch ziemlich schnell im Angesicht der hohen Kosten. Auch bereitete es vielen Leuten Unbehagen von Anderen zu Hause gesehen zu werden. AT&T gab schließlich 1973 seine Pläne auf, das Picturephone Mod II zu vermarkten.

**Abbildung:** Prinzipschaltbild Picturephone

## 2. vat, vic, wb & sdr

Anfang der 90er Jahre entwickelt die Network Research Group des Lawrence Berkeley National Laboratory [11] für Sun Workstations mehrere Programme um eine vollständige Videokonferenz zu ermöglichen. Ursprünglich war die Programmsuite allein für die Anwendung auf dem MBone gedacht, sie wurde aber bald auch für Unicast-Szenarien angepasst.
Die Suite besteht jeweils aus einem Programm für die Ton und die Vieoübertragung, einem „White Board“ und einem „Session Directory“.
Das White Board ist ein virtuelles Zeichenbrett, auf dem alle Konferenzteilnehmer Notizen austauschen können. Das Session Directory verwaltet alle Datenströme mehrerer Konferenzen und bietet sie übersichtlich an. Mit dieser Programmsuite, die mittlererweilen für fast jede Plattform erhältlich ist, lassen sich einfach Konferenzen, Vorlesungen und Seminare realisieren.
Aus den ersten Versuchen mit „vat, the audio conferencing tool“ entstand Version 0 des RTP-Protokolls.

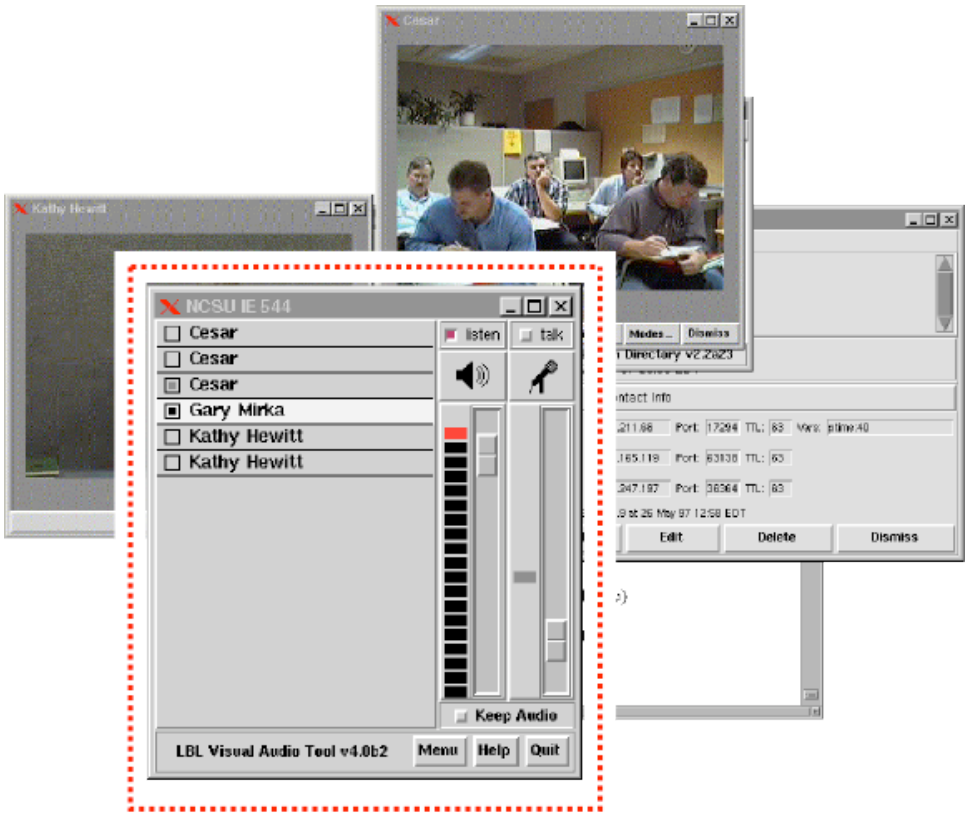



**Abbildung:** vat, vic, wb, sdr (Im Fordergrund: vat)

# 3. Einführung in RTP/RTCP

RTP, das „Realtime Transfer Protocol“, ist ein Internet Standard Protokoll und wurde von der Audio Video Transport Working Group der IETF entwickelt. Es wurde für die Übertragung von Echtzeitdaten, u. a. Bild und Ton, entworfen. Seine Hauptanwendungsbereiche sind „media-on-demand“ und Internet Telefonie.
RTP regelt die Datenübertragung der Echtzeitdaten, d. h. es enthält Timing-Informationen, Möglichkeiten Datenverlust zu detektieren und Inhalt des Stroms zu beschreiben.
In Verbindung mit RTCP, dem „Realtime Control Protocol“ sind Konferenzen beliebiger Größe möglich. RTCP enthält unter anderem eine Möglichkeit zur Identifikation aller beteiligten Systeme, Statistiken über die Verbindungsqualität und bietet Mittel zur Synchronisation mehrerer Datenströme z.B. Bild und Ton an.
Obwohl RTP zuerst für UDP/IP implementiert wurde ist das Protokoll transportunabhängig. Denkbar wäre eine Anwendung auf CLNP, IPX o. ä.
1996 wurde RTP erstmals als RFC1889 veröffentlicht. Seit Mai 2004 wurde das Dokument durch RFC3550 abgelöst.
RTP wird für Voice over IP genutzt. Dies ist fest geschrieben als Standard der International Telecommunication Union (ITU-T H.225.0).

# 4. Spezielle Forderungen von Multimedia-Daten

Wenn man ein Protokoll zur Übertragung von Multimedia-Daten betrachten und seine Stärken und Schwächen aufzeigen möchte, muss man sich erst einige Gedanken zu den speziellen Forderungen machen, die aus den möglichen Anwendungsfällen erwachsen.
Ein denkbares Szenario ist z.B. eine Konferenzschaltung mit Bild und Ton zwischen mehreren Teilnehmern eines Softwareprojektes. Im Allgemeinen werden sich nicht alle zugeschalteten Systeme im gleichen Netz befinden. Die Übertragung könnte über mehrere Zwischensysteme erfolgen. Einige Mitarbeiten könnten auf Grund beschränkter Bandbreite, z.B. weil sie über ein Funknetz kommunizieren, nur den Tonkanal empfangen.
Eine Vielzahl von Problemen entstehen, die bei einer reinen Datenübertragung kaum eine Rolle spielen.
Zu nennen wären hier die Echtzeitanforderung: In Telefonieanwendungen und bei Konferenzen sind spürbare Verzögerungen unerwünscht - Konferenzteilnehmer fallen sich ins Wort oder der Ton ist nicht Lippensynchron. Eng damit verbunden ist der Jitter. Durch die unterschiedlichen Laufzeiten der Datenpakete im Netzwerk entstehen beim Empfänger variierende Verzögerungszeiten, die durch Pufferung ausgeglichen werden müssen.
Endsysteme können in Netzen unterschiedlicher Bandbreite sitzen oder es ist nur ein Prozentsatz der maximal möglichen Datenrate für die Übertragung der Multimedia-Daten reserviert. Ist der Transportkanal fehlerbehaftet, so ist zu überlegen wie Übertragungsfehler oder Paketverluste ausgeglichen werden sollen. Ein bei anderen Protokollen übliches erneutes Senden macht nur dann Sinn, wenn das Paket den Empfänger noch erreicht bevor es veraltet ist. Einzelne fehlerhafte oder ganz verloren gegangene Pakete können in der Anwendung z.B. durch Interpolation einfach rekonstruiert werden.
Sucht ein Protokoll weite Verbreitung und will offen für zukünftige Entwicklung sein, so darf es sich nicht auf eine Plattform festlegen. Es sollte zur Kodierung der eigentlichen Nutzdaten auf die Vielzahl der existierenden standardisierten Datenformate zurückgreifen können.

# 5. Application Level Framing & Integrated Layer Processing

In der Zeitschrift „Computer Communications Review“ der IEEE in der Ausgabe 20 vom September 1990 stellen David D. Clark und David L. Tennenhouse (Laboratory of Computer Science, M.I.T.) 1990 im Artikel „Architectural Considerations for a New Generation of Protocols“ [7] die in RTP/RTCP angewendeten Architekturprinzipien "Application Level Framing" und "Integrated Layer Processing" vor.

## 5.1. Das ISO/OSI Referenzmodell

Um einen Einblick in das Konzept zu erhalten, muss man sich kurz einige Eigenschaften traditioneller Schichten Schichtenmodelle ins Gedächtnis rufen: Das ISO/OSI Referenzmodell oder auch der TCP/IP Protokollstack gliedern die Kommunikation in streng unterteilte Funktionsschichten. Jede Schicht nutzt nur Funktionen der darunter liegenden Schicht um Dienste für die darüberliegende zu erbringen. Von Schicht zu Schicht werden die zu transportierenden Daten gekapselt und können daher nur von der jeweiligen Funktionsschicht interpretiert werden. Zwischen- und Transitsysteme wie Router oder Switches müssen nur die unteren Schichten implementieren um ihre Funktion erfüllen zu können.

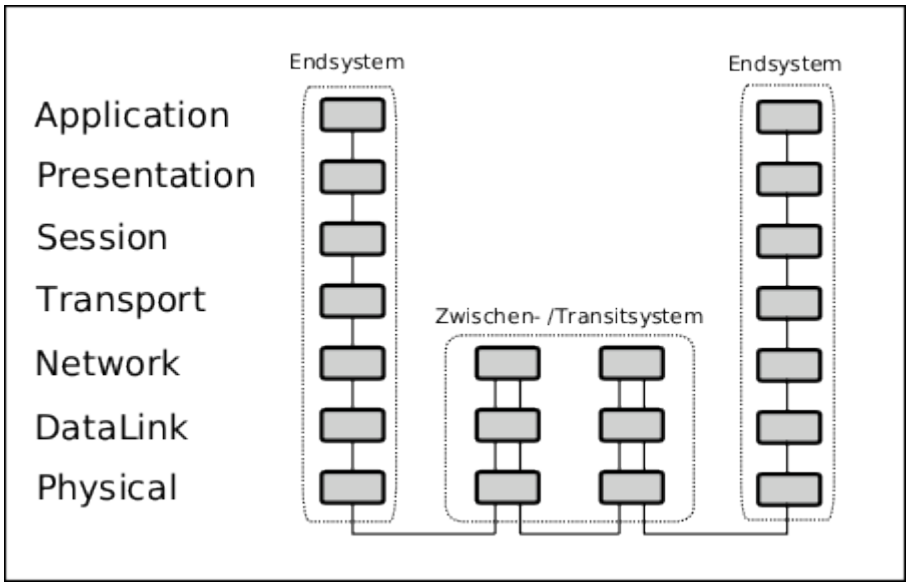



**Abbildung:** ISO/OSI Referenzmodell

## 5.2. Anforderungen und Aufgaben eines Protokolls

Bei Betrachtung zukünftiger Netzwerke können drei Tendenzen festgestellt werde: Die Übertragungsgeschwindigkeit nimmt sowohl im Netz selbst als auch von und zu den Endsystemen stark zu. Bestehende Protokolle stellen immer häufiger den Flaschenhals der Übertragung dar.
Zweitens muss ein modernes Protokoll in Bezug auf die kleinste Übertragungseinheit kompatibel mit neuen Netzwerktechnologien sein. So enthält zum Beispiel eine ATM-Zelle des auf Backbones angewendeten Asynchronen Transfermodus 48 Byte Nutzdaten. [9]
Drittens werden immer mehr Anwendungen in Endsystemen vereint. Moderne PCs übernehmen vielfältige Serverdienstleistungen und sind sowohl für alltägliche Büroarbeiten geeignet als auch als multimediale Zentrale für Kommunikation und Unterhaltung im privaten Bereich.

Viele der heute verwendeten Protokoll-Suiten stützen sich auf das ISO Referenz Modell oder ähnliche in Schichten strukturierte Modelle.
Es ist wichtig zwischen Architektur und Implementierung eines Protokolls zu trennen. Die Architektur beschreibt die Einteilung in funktionale Module, die Semantik der einzelnen Module und die Syntax des Protokolls.
Auf keinen Fall sollte die Architektur des Protokolls die Implementierungsmöglichkeiten einschränken. Der Entwickler einer Anwendung sollte größtmögliche Freiheit bei der Wahl einer konkreten Designstrategie haben. Ein Zwischensystem benötigt weniger Funktionalität als ein Endsystem. Entsprechend können die sinnvollen Optimierungen zur Durchsatzsteigerungen sehr unterschiedlich ausfallen.

Die Hauptfunktionen eines Protokolls lassen sich aufteilen in die Datenverarbeitung und die Flusskontrolle.

Die Datenverarbeitung gliedert sich in 6 Schritte:

- Senden und Empfangen von Daten aus und in das Netzwerk
- Fehlererkennung
- Pufferung auf Senderseite um im Fehlerfall Daten erneut senden zu können
- Kopieren der Daten zwischen dem Netzwerkinterface und der Applikation
- Wandlung der Darstellung der Informationen zwischen verschiedenen Syntaxen

Einige wichtige Funktionen der Flusskontrolle sind:

- Vermeiden von Überlastung der Netzwerkverbindung
- Erkennen von Übertragungsproblemen
- Versenden von Empfangsbestätigungen
- Multiplexen
- Erstellen von Zeitstempeln
- Packetieren der Daten

Diese Funktionen werden in traditionellen Protokollen streng einzelnen Schichten zugewiesen.
Einfache Untersuchungen an bestehenden Implementierungen von Protokollstacks zeigen, dass im Vergleich mit handoptimiertem Code der größte Overhead bei der Darstellungswandlung zu finden ist. Um die Performanz des Protokolls zu erhöhen richtet sich das Augenmerk nun auf diese teuren Funktionen. Eine Lösung wäre es auf die Darstellungswandlung komplett zu verzichten und Daten in einem Raw-Format zu übertragen. Allerdings würde das auf Kosten der Generalität des Protokolls gehen.

Eine andere Herangehensweise ist es der Applikation die Möglichkeit zu geben so viele Operationen wie möglich auf schon empfangenen Daten auszuführen. Effiziente Technologien wie Pipelining können dann angewendet werden.
Um dies zu erreichen muss die Applikation selbst entscheiden können, wie sie auf unvollständige, fehlerhafte Daten oder Datenpakete, die in falscher Reihenfolge eintreffen reagiert.
Bleiben diese Aufgaben in tieferliegenden Schichten wie zum Beispiel TCP kann die Anwendung erst mit dem Bearbeiten der Daten beginnen wenn diese vollständig und in richtiger Reihenfolge vorliegen.

## 5.3. ALP & ILP als Lösungsansatz

Die Prinzipien „Applikation Level Framing“ und „Integrated Layer Processing“ versuchen nun diese Überlegungen zu fassen und umzusetzen:

Die Aufteilung des Datenstroms in Pakete erfolgt in der Applikation, nicht in einer separaten Schicht. Die kleinste interpretierbare Einheit wird die „Application Data Unit“. Sie ist gleichzeitig die kleinste Synchronisationseinheit, d.h. entweder eine ADU wird fehlerlos empfangen und kann an die Bearbeitungspipeline abgegeben werden oder sie wird verworfen. Die Applikation entscheidet dann je nach Anwendungsfall ob sie neu übertragen werden muss oder ob sie rekonstruiert werden kann.
Jede ADU muss Informationen enthalten um der Applikation zu ermöglichen jederzeit den Zustand der Kommunikation zu kennen. Viele Operationen können nach dem Pipelining-Prinzip durchgeführt werden. Aufgaben können nach Performance-Aspekten umgeordnet werden, Operationen können vorgezogen oder gepuffert werden. Zwischen- und Transitsysteme müssen wie im reinen Schichtenmodell nur die unteren Schichten implementieren.

# 6. Verwirklichung der Anforderungen

## 6.1. Trennung Protokoll / Profil / Payload

Um die verschiedenen möglichen Anwendungsfälle von Echtzeitapplikation abdecken zu können gliedert sich die RTP/RTCP Protokolle in drei Teile:
Das RFC 3550 (RTP: A Transport Protocol for Real-Time Applications) [3] enthält die Spezifikationen des eigentlichen Protokolls und aller beteiligten Systeme. Hier werden alle generischen Header, der Ablauf der Kommunikation und der Fehlerfall erklärt.
So genannte Profile beschreiben spezifische Richtlinien für konkrete Anwendungsfälle. Sie definieren in der Spezifikation offen gelassene Parameter, legen mögliche Erweiterungen des RTP-Headers fest und geben die Kodierungsmöglichkeiten für die zu übertragenden Daten an.
Das RFC 3551 (Profile for Audio and Video Conferences with Minimal Control) [4] verwirklicht zum Beispiel das eingangs erwähnte Szenario einer Konferenzschaltung.
Und schließlich existieren noch "Payload Format Specifications" für verschiedene Medienformate. Einige sind schon im RFC 3551 [4] enthalten, neue können aber problemlos zugefügt werden.

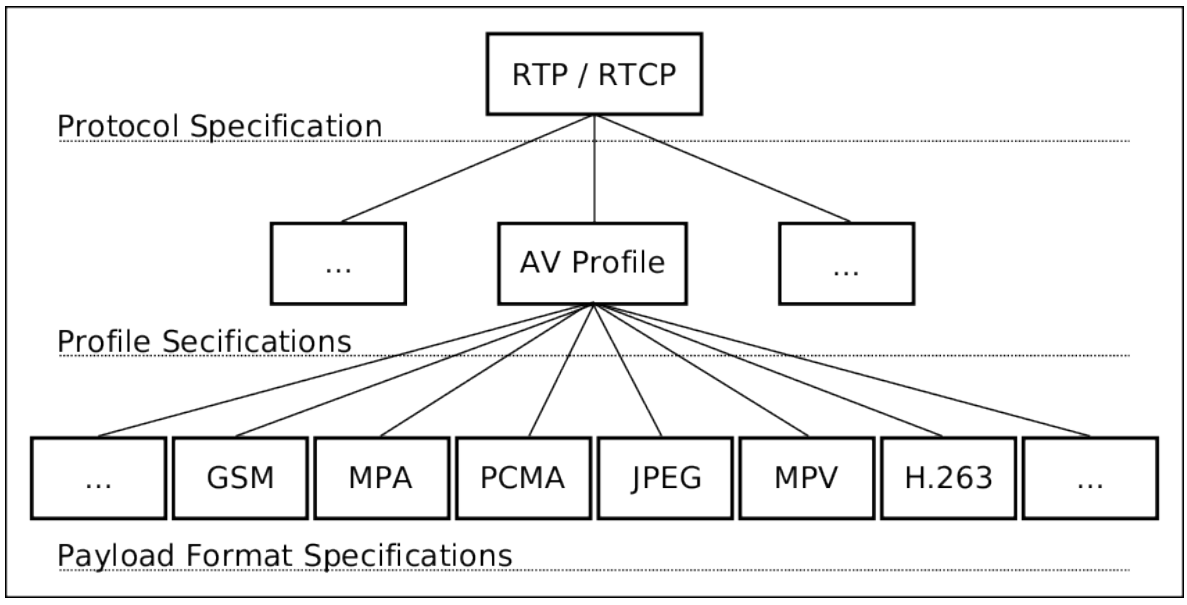



**Abbildung:** Protokoll / Profil / Payload

## 6.2. Die RTP-Sitzung

Unter einer *RTP-Sitzung* versteht man alle Teilnehmer einer RTP-Kommunikation eines Medientyps. Ein Teilnehmer ist entweder ein Endsystem, ein Monitor, ein Translator, ein Mixer oder eine Kombination daraus.
In einer Unicast-Umgebung hält jeder Teilnehmer ein Portpaar für den RTP-und den RTCP-Kanal zu jedem anderen Teilnehmer.
Im Multicast-Fall kann auch für alle Teilnehmer ein gemeinsames Poortpaar vereinbart werden.
Alle RTP-Sessions einer zusammengehörenden Teilnehmermenge nennt man *Multimedia-Session.* Das können bei einer Videokonferenz z.B. eine Video- und mehreren Audio-RTP-Sessions sein.
Jeder Erzeuger eines RTP-Paket-Stroms ist eine *Synchronization Source*. Jede SSRC ist eindeutig durch eine 32-Bit-Zahl identifizierbar und versieht ausgehende Pakete mit einem eigenen Zeitstempeln und Sequenz Nummern.
Eine *Contributing Source* (CSRC) ist ein Teilnehmer dessen Datenstrom mit mehreren anderen von einem Mixer zu einem zusammengefasst wurde.
Der *RTP payload* meint die eigentlichen Daten, die in einem RTP-Paket transportiert werden. z.B. ein Audio-Sample.

## 6.3. Mögliche Netzelemente

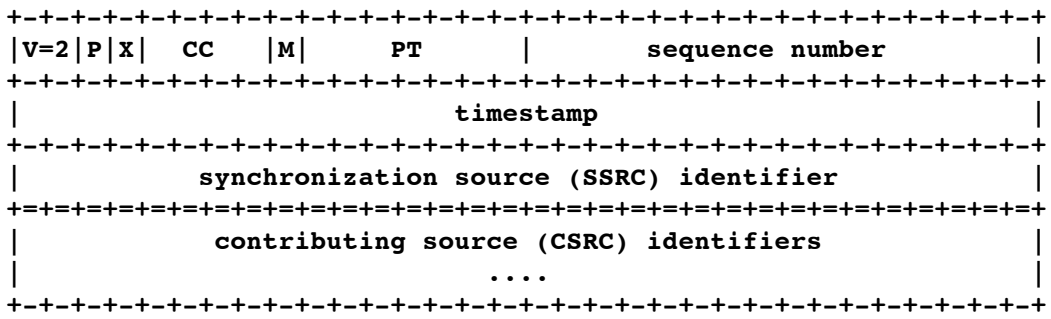
Ein mögliches Netzelement ist der *Monitor*. Ein Monitor ist eine Applikation, die Statusinformationen und Fehlerdiagnosen von anderen Teilnehmern aufsammelt und Statistiken führt. Normalerweise ist ein Monitor in jedes Sende- und Empfangsprogramm integriert. Denkbar ist auch ein unabhängig laufender Monitor in einer Firewall oder für einen ISP.

Ein Zwischensystem in der RTP-Kommunikation ist das Netzelement *Translator*. Seine Aufgaben sind vielfältig. Ein Translator kann z.B. Multi- und Unicast-Domänen verbinden. Mit seiner Hilfe können RTP-Verbindungen getunnelt werden z.B. in Gestalt eines SSH-Tunnels durch eine Application-Level-Firewall. Es ist sowohl ein Wechsel der Kodierung möglich als auch das Verbinden von Netzen mit unterschiedlichen Transportschichten.

Ein *Mixer* ist ein Zwischensystem, dass von mehreren Sendern RTP Pakete empfängt, evtl. das Datenformat ändert, sie zu einem zusammenfügt und diese weiterleitet. Der Mixer synchronisiert die eingehenden Ströme und erzeugt einen neuen Zeitstempel.

# 6.4. Der RTP Header

Viele der eingangs erwähnten Anforderungen spiegeln sich im RTP Header wieder. Untypisch für ein so universal einsetzbares Protokoll ist seine Einfachheit. Aber gerade durch diese einfache Struktur und die Verwendung von Feldern fester Länge sind die Hardwareanforderungen an ein Zwischensystem sehr gering.




In den ersten zwei Bits des Headers wird die Versionsnummer des RTP Protokolls kodiert. Die im RFC 3550 beschriebene Version ist „2“. Version „1“ ist die des ersten Drafts und Version „0“ kennzeichnet das in der Einleitung erwähnte Tool „vat“.

Das *padding-Bit (P)* zeigt an ob vor dem eigentlichen Payload noch Bytes eingefügt wurden um die Blockgröße des Heades anzupassen. Die ist für einige Verschlüsselungsalgorithmen nötig.

Ist das *extension-Bit (X)* gesetzt, handelt es sich um einen erweiterten Header. Dieser kann weitere profilspezifische Informationen enthalten.

Der *CSRC count (CC)* gibt die Anzahl der beitragenden Ströme an die im Feld CSRC aufgelistet werden.

Die Bedeutung des *marker-Bits (M)* kann ebenso in einem Profil definiert werden. Es kann z.B. verwendet werden um besondere Ereignisse anzuzeigen.

Der *payload type (PT)* spezifiziert das Format der transportierten Daten. RFC 3551 vergibt z.B. den payload type 3 für 8kHz Daten im GSM Format.

Die *sequence number* ist die fortlaufende Nummer des Paketes. Diese ist wichtig im Hinblick auf Paketverlust oder Verschlüsselung.

Der *timestamp* ist schließlich der (relative) Zeitstempel für das erste im Paket enthaltene Sample.

## 6.5. Der RTCP-Kanal

Das zu RTP gehörige Kontrollprotokoll RTCP basiert auf periodischem Senden von Paketen an alle Teilnehmer einer Sitzung. Benutzt die RTP Sitzung UDP wird für den RTCP-Kanal einfach ein weiterer Port reserviert.

Im Großen und Ganzen erfüllt RTCP vier Funktionen:
Die Hauptaufgabe ist die Qualität der Datenübertragung zu überwachen. Nur so können Fehler im Übertragungsweg diagnostiziert werden. Adaptive Komprimierungsverfahren können diese Informationen direkt benutzen um die vorhandene Bandbreite optimal zu nutzen.
Zweitens enthält ein RTCP eine eindeutige Kennung für jede RTP Quelle, den *„canonical name“ (CNAME)*. Dies ist nötig da die Identifikationsnummer eines Erzeugers (SSRC) im Laufe einer Sitzung wechseln kann wenn z.B. ein Programm neu gestartet wurde. Über den CNAME und die Zeitstempel im RTP-Paket ist es möglich mehrere Ströme (z.B. Ton und Bild) eines Teilnehmers zu synchronisieren. Um diese beiden Funktionen zu realisieren ist es notwendig, dass jeder Teilnehmer an jeden anderen RTCP Pakete verschickt. Somit ist Allen die Anzahl der Teilnehmer jederzeit bekannt. Daraus kann die Rate der zu sendenden RTCP Pakete berechnet werden.
Die vierte Funktion beinhaltet eine optionale, minimale Sitzungskontrolle. Es können einfache Name-Wert Paare übertragen werden um Informationen wie Name oder Email-Adresse eines Teilnehmers zur Verfügung zu stellen. Will eine Anwendung mehr Funktionen für die Sitzungskontrolle verwenden, kann ein eigenes Protokoll dafür verwendet werden. Zu nennen sind hier beispielhaft das „Session Description Protocol“ (SDP, RFC 2327) [5] und das „Session Initiation Protocol“ (SIP, RFC 3261) [6].

# 7. Quellenverzeichnis


Request for Comments

- [1] 1889 - RTP: A Transport Protocol for Real-Time Applications
- [2] 1890 - RTP Profile for Audio and Video Conferences with Minimal Control
- [3] 3550 - RTP: A Transport Protocol for Real-Time Applications
- [4] 3551 - RTP Profile for Audio and Video Conferences with Minimal Control
- [5] 2327 - SDP: Session Description Protocol
- [6] 3261 - SIP: Session Initiation Protocol

[7] „Architectural Considerations for a New Generation of Protocols“
David D. Clark, David L. Tennenhouse (M.I.T.) in SIGCOMM Sep. 1990

[8] „Praxisbericht“
Olaf Beier,
http://heim.ifi.uio.no/~griff/theses/Olaf.Beier/Praxisbericht.pdf

[9] RTP: About RTP and the Audio-Video Transport Working Group
Henning Schulzrinne, http://www.cs.columbia.edu/~hgs/rtp

[9] „Lehr- und Übungsbuch Telematik”
Gerhard Krüger, Dietrich Reschke, Fachbuchverlag Leibzig, 2. Auflage

[10] „Bell System Memorial - WESTERN ELECTRIC TELEPHONES“
http://www.bellsystemmemorial.com/telephones-picturephone.html

[11] „LBNL's Network Research Group“
http://www-nrg.ee.lbl.gov/

# 8. Abkürzungen

| | |
|---|---|
| AT&T | American Telephone & Telegraph Corporation |
| CLNP | Connectionless Network Protocol |
| IEEE | Institute of Electrical and Electronics Engineers |
| IETF | Internet Engineering Task Force |
| IP | Internet Protocol |
| IPX | Internetwork Packet eXchange |
| ITU | International Telecommunication Union |
| MBone | Multicast Backbone |
| MIT | Massachusetts Institute of Technology |
| RTCP | Realtime Control Protocol |
| RTP | Realtime Transfer Protocol |
| UDP | User Datagram Protocol. |