Streaming: Agenda

- "Definition"
- Übersicht

- Protokolle: RTP, RTCP, RTSP, RSVP
- Microsoft ASF und WMA
- RealAudio, Apple Quicktime, LiquidAudio

- Napster
- Gnutella

später:
- SDMI, Verschlüsselung, Wasserzeichen
- Mediensuche

Streaming: Literatur

trotz Bedeutung des Themas:
bisher keine Lehrbücher, Tutorials
=> statt dessen nur Weblinks / Hersteller-Whitepaper / ...

Internet RFCs, insbesondere RFC1889 für RTP und RTCP

www.microsoft.com/windowsmedia
www.realaudio.com
www.apple.com/quicktime
www.napster.com
gnutella.wego.com
www.liquidaudio.com

Streaming: Definition

"Streaming Media / Audio" :=

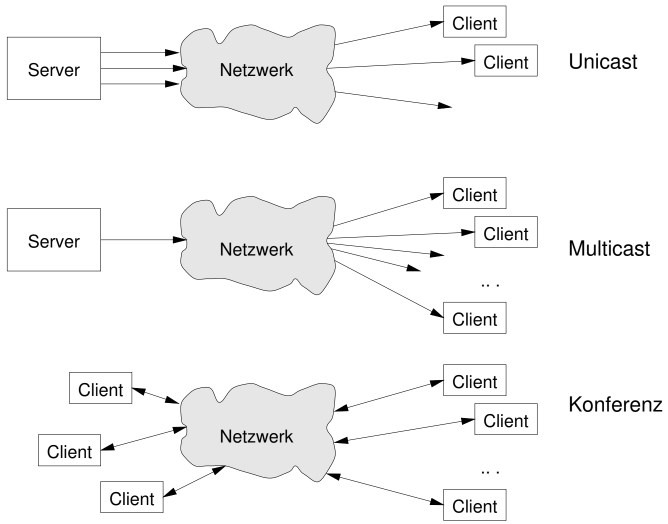
- Echtzeit-Übertragung von Medien, hier: Audiodaten
- von einem Server zu einem / vielen Clients

- ohne vorheriges Laden der gesamten Daten
- Beschränkung auf Bitrate des Kanals, mit Schwankungen
- erfordert Fehlertoleranzmechanismen
- Tradeoff Qualität / Robustheit / Bitrate

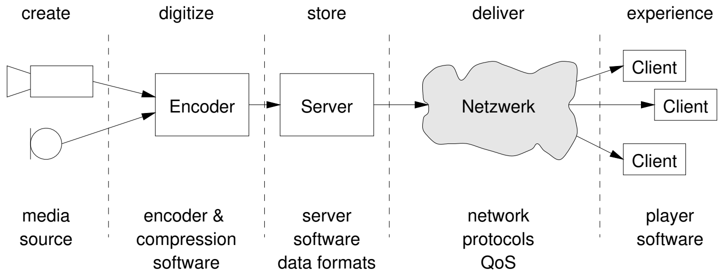
- Audio auch standalone
- aber meistens Video / Animationen mit sychronem Audio

- großes Marktpotential (z.B. Internet-Radio, Video on demand, ...)
- Microsoft WMA, RealAudio, Apple Quicktime, Liquid Audio, ...
- Mediensuche / Tauschbörsen (Napster) / Copyright-Fragen

Streaming: Szenarien



Streaming: System



funktionierendes System erfordert alle Komponenten:

- alle Anbieter liefern "integrierte Lösung"
- Encoder + Server + Management + Player (+ Verschlüsselung)
- Microsoft, RealNetworks, Apple, ...
- Live-Übertragung erfordert auch Echtzeit-Encoder

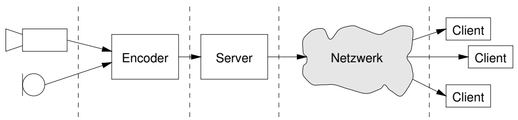
Streaming: Bandbreiten

"streaming" ist stark von verfügbarer Bandbreite abhängig:

Modem	28 .. 56	Kb/s	
ISDN	64	Kb/s	
DSL	128 .. 768	Kb/s	(bis ca. 4 Mb/s)
Intranet	10 .. 100	Mb/s	
MPEG-2 (DVD)	4 .. 9	Mb/s	
MPEG-1, CDDA	1.5	Mb/s	
MPEG-4 (DivX)	1.5	Mb/s	
MP3 Audio	64 .. 320	Kb/s	
RealAudio 8	10 .. 160	Kb/s	

=> selbst MP3 nicht per Modem streaming-fähig
=> Bildtelefonie (H.263 QCIF 15fps) erfordert mindestens ISDN

Streaming: Vor- und Nachteile



Vorteile:

- keine Wartezeit während der Dateiübertragung
- sofortiges Reinhören ("tune in")
- Live-Übertragungen
- kein Speicherplatz auf dem Client benötigt

Nachteile:

- durch verfügbare Bandbreite begrenzt
- deshalb oft noch schlechte Qualität
- spezielle Server und Software erforderlich

Streaming: QoS

"Quality of Service" ?!

- Internet ist "best effort" Netzwerk
- selbst innerhalb von Firmen-Intranets keine Garantien
- ATM hat sich nicht durchgesetzt

=> keine Bandbreite reservierbar
=> Fehlertoleranzmechanismen notwendig:

- forward error correction
- Dienstgütemanagement
- adaptive Verfahren

○ Forschungsschwerpunkt bei TKRN (Prof. Wolfinger)
○ insbesondere QoS bei Videokommunikation

Streaming: Marktbedeutung

- digitale Techniken kurz vor dem Durchbruch
- mit DSL-Modem ist Video-on-demand möglich

=> enormes Marktvolumen

- aktiver Kampf um Marktanteile

=> proprietäre, nicht dokumentierte Protokolle
=> Standards sind zu langsam (?!)

- Marktführer: Microsoft, RealNetworks, Apple
- aber Revolutionen jederzeit möglich, siehe Napster
- oft keinerlei Dokumentation verfügbar
- Testberichte / Vergleichstests fragwürdig
  siehe "NSTL" Tests: MS vs. RealAudio

Streaming: Internet

Multimedia-Übertragung im Internet ("IP"):

- hohe Bandbreiten erforderlich (s.o.)
- für Unicast / Multicast- Übertragungen
- nur Paketvermittlung, keine Punkt-zu-Punkt-Verbindungen
- Echtzeitanforderungen: z.B. Latenz < 250ms für Telephonie

=> "klassische" Protokolle (ftp) für Streaming ungeeignet
=> neue Protokolle notwendig

- verlorene Daten (dropped frames) oft tolerierbar
- verlorene Pakete meistens bei überlastetem Netzwerk
- wiederholte Übertragung verschlimmert das Problem

=> auf IP/UDP aufsetzen, nicht auf TCP

Streaming: Internet-Protokolle

neue Protokolle für Multimedia-Daten:

- Pakete klassifizieren und priorisieren
- Reservierung von Resources / Bandbreite / Routes
- Anpassung an Netzauslastung

- RSVP — resource reservation protocol — RFC2205..2209
- RTP — real time transport protocol — RFC1889, 1890
- RTCP — real time control protocol
- RTSP — real time streaming protocol

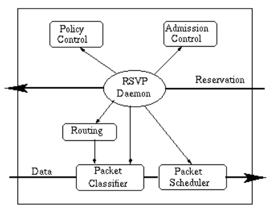
- proprietäre Protokolle (Aufsetzen auf IP/UDP)

RSVP

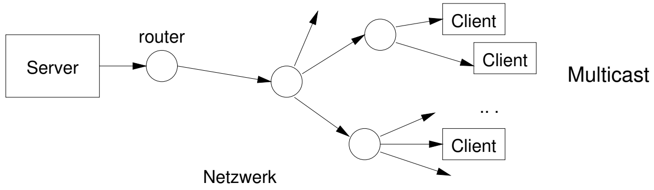
"resource ReSerVation Protocol" : RFC 2205..RFC2209 (approved 1997)

- Internet "network control protocol"
- Empfänger beantragt garantierte Bandbreite (QoS)
- für eine Punkt-zu-Punkt Verbindung
- bei allen beteiligten Routern
- realisiert über verändertes (priorisiertes) Packet-Scheduling

- RSVP überträgt keine Nutzdaten
- rsvpd Dämon-Prozeß auf jedem Router
- "admission control" Resourcen übrig ?
- "policy control" Zugriffsrechte ?
- jeder Router hat "Veto-Recht"



RSVP: Stream Merging

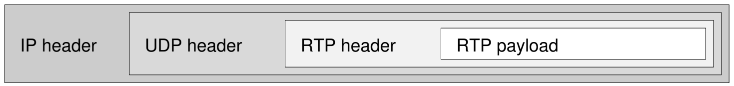


Merging:

- Multicast: Verteilung der Daten an viele Empfänger
- Reservierungen mehrerer/aller Clients kombinieren
- erfordert keine zusätzliche "Upstream"-Bandbreite (!)

=> Übertragung skaliert auch für sehr viele Clients

RTP



"real time transport protocol", RFC1889 / RFC1890

• setzt auf UDP auf	aber auch RTP auf IP direkt möglich
• timestamps	für Streaming / Echtzeitwiedergabe
• sequence numbers	weil UDP keine Reihenfolge garantiert
• payload type identifier	z.B. PCM, MP3, MPEG-2, H.261
• source identification	z.B. Sprecher bei einer Telekonferenz
• encryption	

- in Quicktime / RealAudio als Transportprotokoll verwendet

RTP: Packet-Header

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

V:	version number (2)
P:	padding
CC:	CSRC count (>1 if payload mixes data from several sources)
M:	marker, e.g. "frame boundary"
PT:	payload type
sn, ts	sequence number, timestamp

RTCP: RTP control protocol

- periodische Übertragung von Kontrolldaten
- zwischen allen Beteiligten (Server <-> Clients)

Funktionen:

- Statusmeldungen der Clients (Verlustrate, Jitter, ...)
- eindeutige Kennzeichnung aller Quellen (CNAME)
- Adaption der RTCP Paketrate (wegen Skalierbarkeit)
- ○ Session-Control

Pakettypen:

SR	sender report	transmission statistics
RR	receiver report	perception statistics
SDES	source description	incl. CNAME
BYE	end of participation	
APP	application specific function	

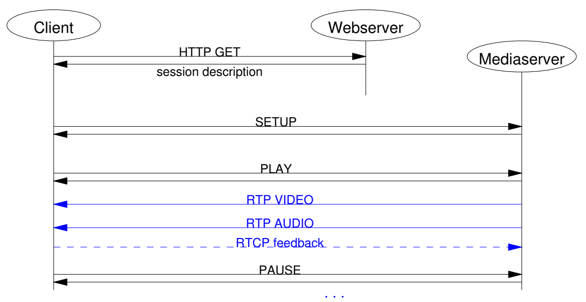
RTCP: Receiver Report

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|    RC   |   PT=RR=201   |             length            | header
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     SSRC of packet sender                     |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                 SSRC_1 (SSRC of first source)                 | report
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ block
| fraction lost |       cumulative number of packets lost       |   1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           extended highest sequence number received           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      interarrival jitter                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         last SR (LSR)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   delay since last SR (DLSR)                  |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                 SSRC_2 (SSRC of second source)                | report
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ block
:                               ...                             :   2
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                  profile-specific extensions                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```


RTSP

"real time streaming protocol" := "Internet Videorecorder"

- Steuerung von Medienübertragung: play, ff, rewind, ...
- Medien-URLs: rtsp://media.example.com:554/twister/
- RTP/RTCP/RSVP für Datentransfer



RTSP: URLs

- siehe RTSP-Präsentation (H. Schulzrinne)

Microsoft: WindowsMedia

- seit ca. Q4/98 enormes Engagement von Microsoft:

www.microsoft.com/windowsmedia/

- Windows Media Player 6.4 / 7 / 8 beta
  - eigenes Audioformat
  - eigenes Videoformat
  - Formate nicht mehr dokumentiert
  - Verschlüsselung und Wasserzeichen
  - aber auch Unterstützung von Standards
- Windows Media Tools
  - Windows Media Encoder
  - Presentation Broadcast (Powerpoint+Audio)
  - Windows Media Rights Manager

WindowsMedia: Ziele, Features

*"deliver digital media, such as songs and videos,
through the Internet in a protected and secure manner. [...]
Provides tools for protecting digital files so you distribute
them and maintain copyright protection. [...]
Solutions are flexible enough to be easily adapted to your
current business model or to create new ones."*

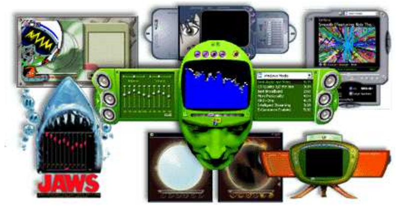
Microsoft Media Rights Manager 7 SDK

Komponenten des WindowsMedia Konzepts:

- Windows Media Player 7
- Windows Media Tools
- Windows Media Rights Manager 7

- Player auch für nicht-Windows Plattformen verfügbar
derzeit: Macintosh, Solaris (!), Pocket-PC, "palm-size PC"

WindowsMedia: Media Player 7

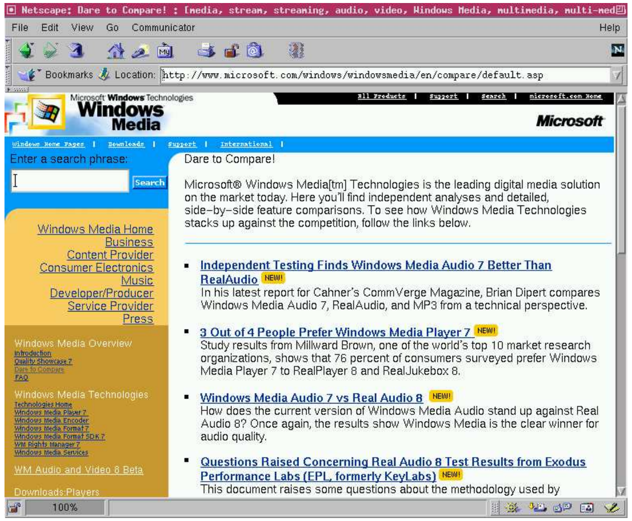


Skins :-(

wesentliche Veränderungen seit Player 6.x auch unter der Oberfläche:

- online feedback über Mediennutzung
- unterstützt Windows Media Rights Manager
- z.B. "counted operations", "expiration", ...

Microsoft WindowsMedia: Website



[12/2000]

AVI: Audio Video Interleaved

RIFF header	video data	audio data	. . .	index

RIFF header	list	v	i	d	e	o	.	audio	v	i	d	e	o	.	audio	v	i	d	e	o	.	audio	. . .	index

normal / interleaved

- RIFF-Dateiformat für Multimedia / Videos
- eingeführt mit Video for Windows (Win 3.11)
- definiert ca. 20 Chunk-Typen
- List-Chunk erlaubt "verwobene" Daten
  - Audiodaten, Videoframes (BMP), Audiodaten, ...
  - Datei muß vor Abspielen nicht voll geladen werden

- Spezifikation in Windows API, Übersicht in c't 94/11 S.327
- mittlerweile von ASF abgelöst (s.u.)

AVI: Beispiel

```
(0x00000000)      ID:<RIFF>   Size:0x000fe964
                  Form Type = <AVI >
(0x0000000c)          ID:<LIST>   Size:0x000007d4
                      List Type = <hdrl>
(0x00000018)              ID:<avih>   Size:0x00000038
(0x00000058)              ID:<LIST>   Size:0x00000474
                          List Type = <strl>
(0x00000064)                  ID:<strh>   Size:0x00000038
(0x000000a4)                  ID:<strf>   Size:0x00000428
(0x000004d4)              ID:<LIST>   Size:0x0000005c
                          List Type = <strl>
(0x000004e0)                  ID:<strh>   Size:0x00000038
(0x00000520)                  ID:<strf>   Size:0x00000010
(0x00000538)              ID:<vedt>   Size:0x00000008
(0x00000548)              ID:<JUNK>   Size:0x00000298
(0x000007e8)          ID:<LIST>   Size:0x000fd184
                      List Type = <movi>
(0x000007f4)              ID:<LIST>   Size:0x000005b0
                          List Type = <rec >
(0x00000800)                  ID:<01wb>   Size:0x000005a4
...                           /* einige Male nur Sound */
(0x00004c98)              ID:<LIST>   Size:0x000019da
                          List Type = <rec >
(0x00004ca4)                  ID:<00dc>   Size:0x00001422
(0x000060ce)                  ID:<01wb>   Size:0x000005a4
...                           /* diverse Male Sound und Bild */
(0x000db804)              ID:<LIST>   Size:0x00002d72
                          List Type = <rec >
(0x000db810)                  ID:<00dc>   Size:0x00002d66
...                           /* einige Male nur Bild */
(0x000fd974)          ID:<idx1>   Size:0x00000ff0
```

RIFF: AVI

Header

[c't 11/94 327ff]

Daten (MOVI):
Audio / Video / ...

Index

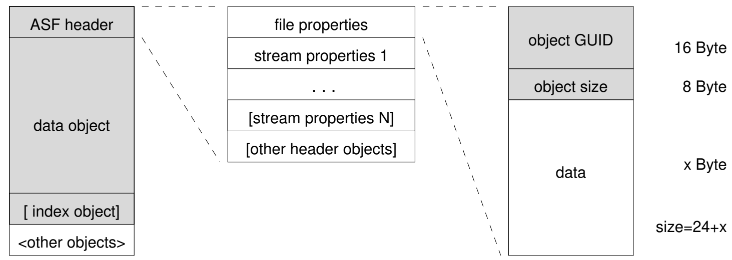
ASF: Advanced Streaming Format

"... an extensible file format designed to store synchronized multimedia data. It supports data delivery over a wide variety of networks and protocols, while still proving suitable for local playback. The explicit goal of ASF is to provide a basis for industry-wide multimedia interoperability, with ASF being adopted by all major streaming solution providers."

[Microsoft, Real Networks 1998]

- RIFF-ähnliche Struktur: header, index, interleaved data
- Chunks (=:"objects") per GUID gekennzeichnet
  - bei Bedarf Registrierung der GUIDs bei Microsoft
  - erlaubt z.B. Codec-Download
  - auch der Player identiziert sich über seine GUID
  - sehr feine Copyright / Nutzungskontrolle
- siehe ASF-Spezifikation (Version 1.0, 26.02.1998)
- aber aktuelle Version (WMA) von M$ nicht mehr dokumentiert

ASF: Object



- Datei besteht aus ASF objects (statt RIFF chunks)
  - oberste Ebene: header / data / [Index] / [more objects]
  - weitere Ebenen durch Einteilen des data-object
  - insbesondere: mehrere Datenströme innerhalb des data-object
  - Header-Object bei Bedarf wiederholt senden
  - Größenfeld erlaubt Objekte bis zu 2**64 Byte

ASF: Audio object

Field Name	Type	#bits
Codec ID	GUID	128
Error Concealment Type	GUID	128
Bits per Sample	UINT	32
Samples per Second	UINT	32
Avg. Frame Size	UINT	32
Max. Frame Size	UINT	32
Samples per Frame	UINT	32
Flags	UINT	16+16
Number of Channels	UINT	16
. . .	. . .	?

- vordefinierte "media types" für:
  Audio / Video / Image / MIDI / Text / Timecode / . . .

ASF: GUID

GUID/UUID := "globally/universally unique IDs"

- 128-bit Signatur
- entwickelt für NCS (Apollo), übernommen in OSF/DCE und Windows
    - Ethernet-MAC Adresse plus Zeitmarke (eindeutig)
    - oder 47-bit Zufallsadresse plus Zeitmarke
    - Zeit mit 100ns Auflösung, Sequenznr. zur Korrektur (reboot)
    - eindeutig bis 3400 n.Chr.

ASF-Header:	`D6E229D1-35DA-11DA-9034-00A0C90349BE`
ASF-Data:	`D6E229D2-35DA-11DA-9034-00A0C90349BE`
ASF-Index:	`D6E229D3-35DA-11DA-9034-00A0C90349BE`
usw.	`time-low-mid--high|seq.|ethernet----`

- jeder Windows-Rechner eindeutig identifzierbar

WindowsMedia: Media Encoder

Windows Media Encoder:

- Basiskomponente für Windows-Media Authoring
- liest diverse Formate (WAV, MP3, ..., MPEG-1, ...)
- und schreibt WMA

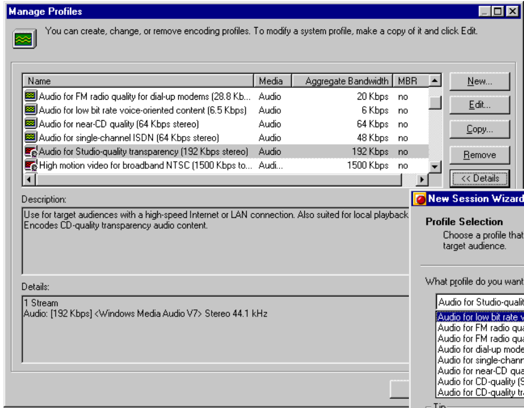
- erlaubt diverse Text-Infos (Autor, Copyright, Titelname, ...)
- diverse "profiles" mit Voreinstellungen

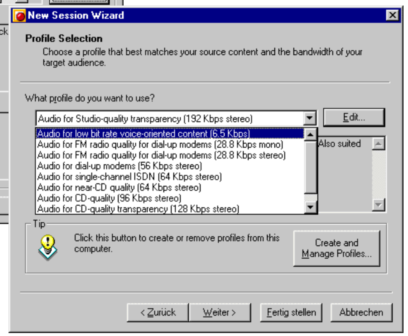
- Datenformat nicht mehr dokumentiert
- erweitertes ASF, plus Verschlüsselung

- Encoder läuft auch als Server
- bis 50 Unicast-Clients

WindowsMedia: Authoring

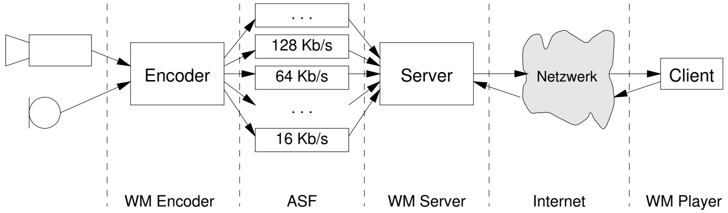
WindowsMedia: Audio Profiles, Wizard





- div. Presets für Sprache .. Studio
- einfache Benutzung per Assistent

WindowsMedia: Intelligent Streaming



- Encoder erzeugt (optional) spezielle ASF-Datei:
  - mehrere parallele Datenströme
  - mit verschiedenen Bitraten
  - entsprechende Qualitätsstufen

- Player sendet Feedback über Bandbreite und Paketverluste
- Server sendet nur den Datenstrom der max. möglichen Qualität

WindowsMedia: Security

- mögliche Restriktionen für WMA/WMV-Dateien:
  - play on PC
  - counted play
  - start date
  - expire date
  - burn to Audio CD
  - counted CD burn
  - transfer to portable device
  - ... and many more

- Player überprüft vorhandene Lizenzen
- oder versucht, neue Lizenz zu erhalten:

  "might issue a licence silently, so the consumer is unaware of the process"

  "... supports pre-delivered licences"

RealAudio

Real Networks : Marktführer für "streaming media"

- RealSystem G2: Audio- und Videostreaming
- auch für Modemverbindungen
- mit Bitraten < 28.8 kbps
- Klangqualität entsprechend schlecht (auch laut Vergleichstests)

- Player verfügbar für viele Plattformen
- z.B. als Plugin im Standardumfang von Netscape 4.x enthalten

- RealAudio8: seit 23.10.2000
- basiert auf Sony ATRAC3 (wie MiniDisc)
- CD-nahe Qualität ab 64 kbps

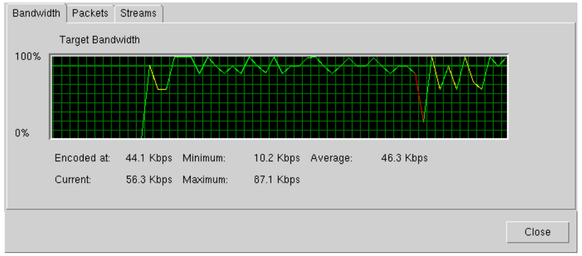
  Whitepaper unter www.realnetworks.com/realsystem/tech_overview.html

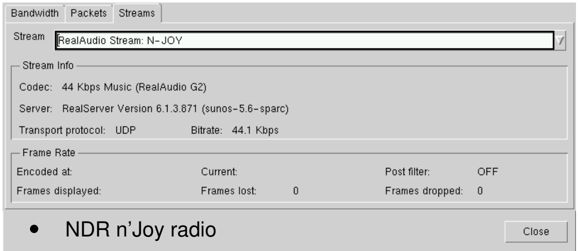
RealAudio: RealPlayer

[Linux x86 Version]

- unterstützt RealAudio G2 und RealAudio 8
- aber neues Format noch wenig verbreitet

RealAudio: Statistik





- NDR n'Joy radio

Liquid Audio

RealAudio: RealServer

RealServer Basic	25 users free
RealServer Plus:	60 concurrent users $ 1.995,95 RealAudio 8 (64 kbps) RealVideo (VHS quality)
RealServer Professional	100+ users $ 8339.00

- Solaris SPARC, Linux x86, Windows x86
- 256 .. 1024 MB RAM

Liquid Audio

viertes "big player" System: www.liquidaudio.com

Encoder:	(Liquifier Pro)
Server:	(Liquid Server)
Player	(Liquid Player)

- Datenformate MP3, AC-3, AAC, WMA
- Verschlüsselung der Dateien (RSA)
- digitales Wasserzeichen für Erkennung analoger Kopien
- diverse Optionen, z.B. Start- und End- Gültigkeitsdatum

- Player wird an die Hardware "gedongelt"
- Dateien können nur auf einem Player abgespielt werden

Apple Quicktime

"Quicktime" :

- Streaming-Lösung von Apple
- Player und Tools für Mac, Windows, Java
- bereits "lange am Markt" (1994?), daher stark verbreitet
- Version 5 aktuell im beta-Preview (Q1/2001)

- Audio, Video, Text, ...
- Präsentationen, VR-Panoramen, ...
- unterstützt viele Formate / Codecs
- Streaming via RTP/RTCP oder HTTP

- modulare Architektur
- Java-Anbindung mit eigener API

Quicktime: RTP vs. HTTP

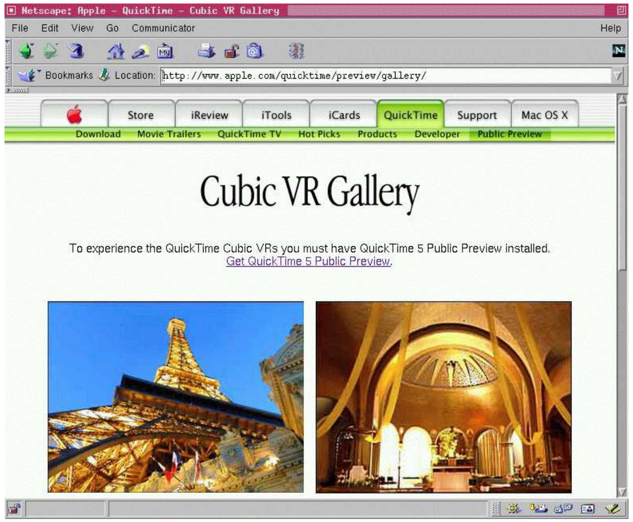
RTP:

+ Live-Übertragungen, Multicast
+ echtes Streaming
+ auch für einzelne Tracks (z.B. nur Audio)
+ Steuerung via RTSP (Szene überspringen...)

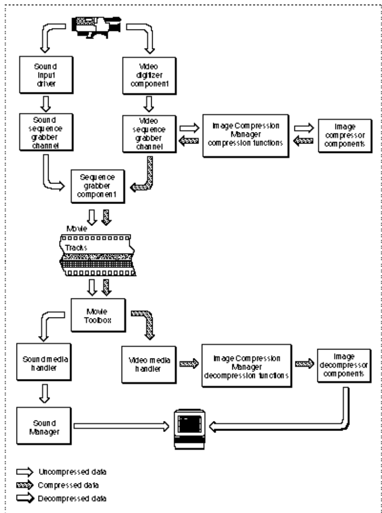
HTTP:

+/− TCP/IP, aber keine Paketverluste
− nur für gesamte Datei (alle Tracks)
+ transparent für Firewalls (meistens)
+ für alle Medientypen (z.B. Sprites)

Quicktime VR: Panoramabilder



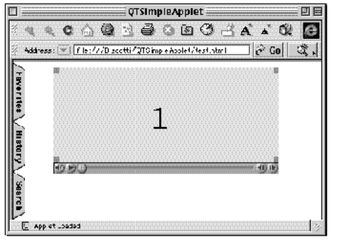
Quicktime: Authoring



- text

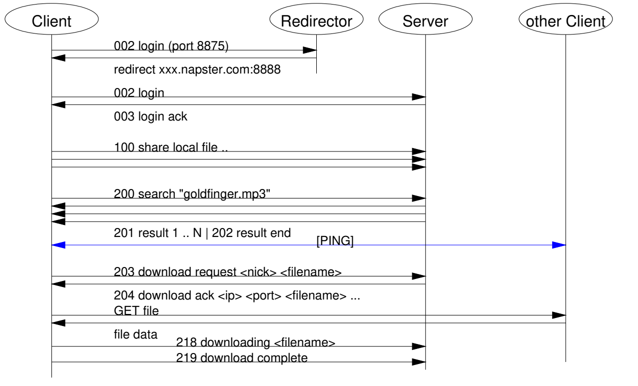
Quicktime: Java API

```
public class QTSimpleApplet extends Applet {

private Drawable myQTContent;
private QTCanvas myQTCanvas;

public void init () {
  try {
    QTSession.open();
    setLayout (new BorderLayout());
    myQTCanvas = new QTCanvas (QTCanvas.kInitialSize, 0.5F, 0.5F);
    add (myQTCanvas, "Center");

    QTFile file = new QTFile (getCodeBase().getFile() +
    getParameter("file"));
    myQTContent = QTFactory.makeDrawable (file);
  } catch (Exception qtE) { ... }
}

public void start () {
  try {
    if (myQTCanvas != null)
      myQTCanvas.setClient (myQTContent, true);
  } catch (QTException e) { ... }
}

...
```



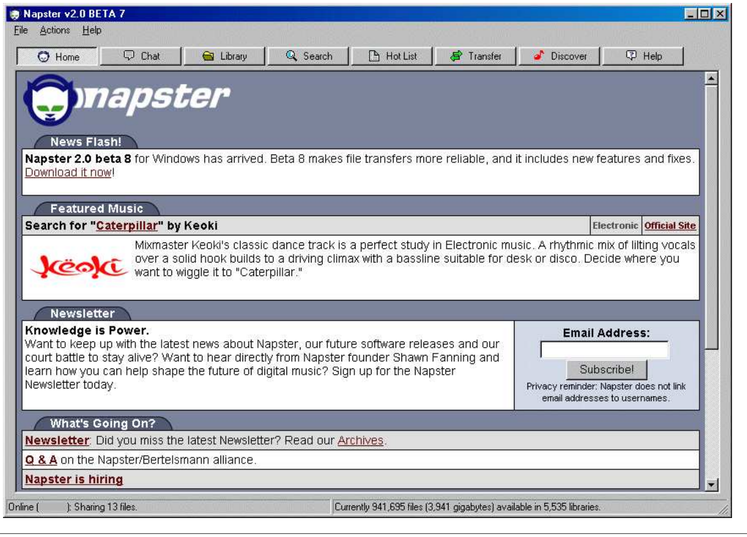
- "arguably simpler than C code"
- "C code differs for Mac and Windows"

Napster



- Internet-Tauschbörse für MP3-Dateien
- Konzept natürlich auch allgemeiner einsetzbar

Napster: Homepage



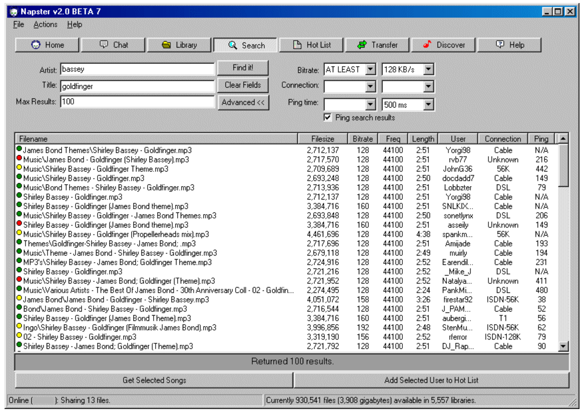
Napster: Konzept

- einfaches Protokoll, ca. 30 Befehle
- eingeschränkt transparent für Firewalls

Napster-Server:

- Metadaten (z.B. user passwords)
- Index aller gerade verfügbaren "shared files"
- verarbeitet die Suchanfragen
- aber Downloads laufen direkt zwischen Clients
  - Benutzerprofile
  - detaillierte Statistiken
  - geringe Belastung der Server
- Servercluster für Lastausgleich und Skalierbarkeit (6M users)
- schwere Sicherheitslücke via Auto-Update des Clients

Napster: Screenshot



- Suche: Anfrage und Ergebnisliste
- jeder Download wird vom Server protokolliert

Napster: Login

- Protokoll setzt auf TCP/IP auf
- Datenformat: <length> <type> <data>
  - little endian byte order

Beispiel login:

length	
<2>	"login" command type
<nickname>	user nickname inside Napster
<password>	user password
<port>	data transfer port, 0=behind firewall
<client info>	string with client version ID
<link type>	0=unknown, 1=14.4, 2=28.8, ... 10=T3
[<build number>]	windows build version

Napster: Client Search Request

length	
<200>	"search" command
[<artist name>]	filename contains "artist" | "song" substr
<max results>	0 .. 100 results returned
[<song name>]	same as artist name :-(
[<linespeed op type>]	0=unknown, 1=14.4 kbps, ...
[<bitrate op type>]	MP3 bitrate kbps
[freq op type>]	MP3 samplerate
[local only]	search all / current Napster server

200 FILENAME CONTAINS "goldfinger" MAX_RESULTS 25

BITRATE "AT LEAST" "128" LINESPEED "AT LEAST" 10

Napster: Search Response

length	
<201>	"search response" command
"filename"	
<md5>	MD5 hash of file
<size>	file size in bytes
<bitrate>	MP3 bitrate in kbps
<frequency>	MP3 samplerate
<length>	play time in seconds
<nickname>	the person sharing the file
<ip>	the ip of the user with the file
<link type>	0= unknown, ...
[<weight>]	positive="better" match

"goldfinger.mp3" 7d733...bcd 2558199 128 44100 159

"JamesBond" 134.100.13.183 4

Streaming: Marktbedeutung Napster

heise online news

Über 6,7 Millionen Napster-User allein in den USA
[06.10.2000 10:13]

Die Musik-Tauschbörse **Napster[1]** entwickelt sich derzeit zur am schnellsten jemals über das Internet verbreiteten Software. Wie die Marktforscher von **MediaMetrix[2]** jetzt bekannt gaben, ist die Zahl der User, die Napster von zu Hause aus benutzen, in den USA von 1,1 Millionen im Februar dieses Jahres auf über 6,7 Millionen im September gestiegen. Damit sind von den insgesamt knapp 80 Millionen US-Amerikanern, die von zu Hause aus im Internet surfen, 8,5 Prozent Napster-User. Und vom Arbeitsplatz aus frequentieren den Dienst zusätzlich über 1,1 Millionen Surfer. Welche Reichweite Napster weltweit hat, teilten die Marktforscher nicht mit.

Der Spaß für die Napster-User könnte aber bald vorbei sein. Derzeit kämpft die Tauschbörse von einem amerikanische Berufungsgericht ums **Überleben[3]**. Die Vereinigung der amerikanische Musikindustrie hatte gegen Napster wegen angeblicher Verstöße gegen Urheberrechtsbestimmungen geklagt. (**axv[4]**/c't)

URL dieses Artikels:
http://www.heise.de/newsticker/data/axv-06.10.00-000/

Links in diesem Artikel:
[1] http://www.napster.com
[2] http://www.mmxi.com
[3] http://www.heise.de/newsticker/data/jk-03.10.00-001/
[4] mailto:axv@ct.heise.de

- 6.7 Mio. (aktive) Benutzer
- Auswirkungen auf CD Verkäufe?!
- Tauschbörse / Raubkopien ?!

Streaming: Marktbedeutung ?!

[www.modernhumorist.com]

Gnutella

- Protokoll zum Datenaustausch zwischen vernetzten Rechnern
- definiert Suchanfragen und eigentlichen Transfer (via HTTP)

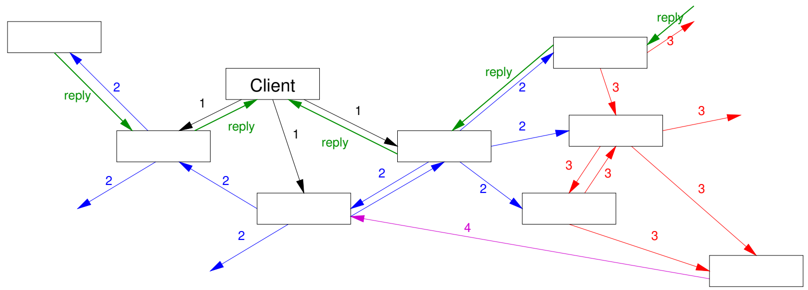
- völlig verteilte Architektur, kein zentraler Server (!)
- jeder Client dient gleichzeitig als Server und Suchmaschine

  =: "servant" (servent?)

- entwickelt von Gnullsoft (Nullsoft, WinAMP)
- nach Kauf von AOL sofort eingestellt
- aber bereits "überall" im Internet erhältich, gnutella.wega.com
- reverse-engineering des Protokolls, Clients für Win/Mac/X11/Java/...

- *"the source code has not been released yet [...], due to AOL locking it in a closet underneath a large pile of Time Warner's dirty laundry"*
  Gnutella 0.56 FAQ

Gnutella: Suchstrategie

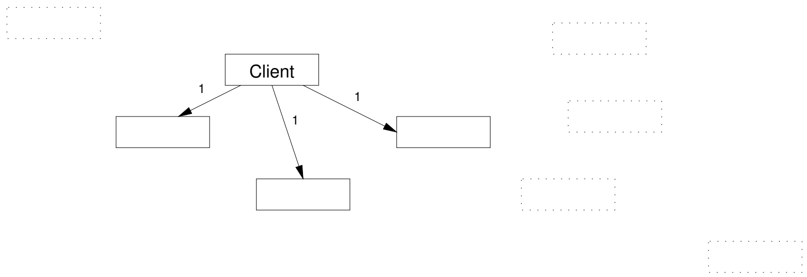


- Client befragt seine unmittelbaren Nachbarn, z.B. mit TTL=25
- diese wiederum ihre unmittelbaren Nachbarn, --TTL

=> Paketflut (exponentielles Wachstum!?)
=> Kennzeichnung jeder Anfrage mit GUID
=> jeder Servant bearbeitet eine Anfrage (höchstens) einmal

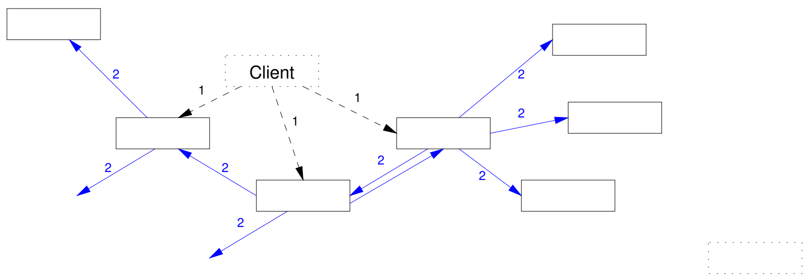
- Anfragen durchsuchen gesamtes Gnutella-Net

Gnutella: Suchstrategie (1)



- Client befragt seine unmittelbaren Nachbarn, TTL=25
- Anzahl der Nachbarn abhängig von verfügarer Bandbreite
  - schnelle Hosts erlauben viele Verbindungen
  - Modem/ISDN-Hosts mit wenigen Verbindungen

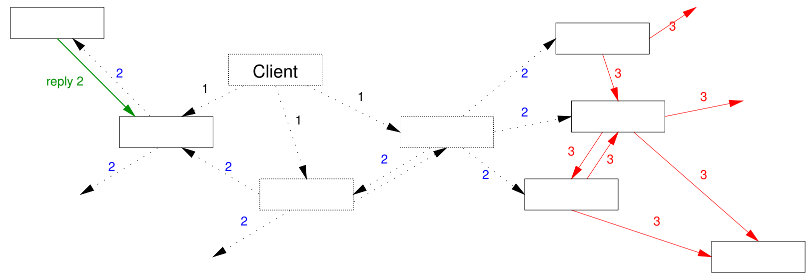
Gnutella: Suchstrategie (2)



- Client befragt seine unmittelbaren Nachbarn, TTL=25
- diese wiederum ihre unmittelbaren Nachbarn
- ausgenommen den fragenden Rechner selbst

=> Paketflut (exponentielles Wachstum, zirkuläre Anfragen)
=> Kennzeichnung jeder Anfrage mit GUID
=> jeder Servant bearbeitet eine Anfrage (höchstens) einmal

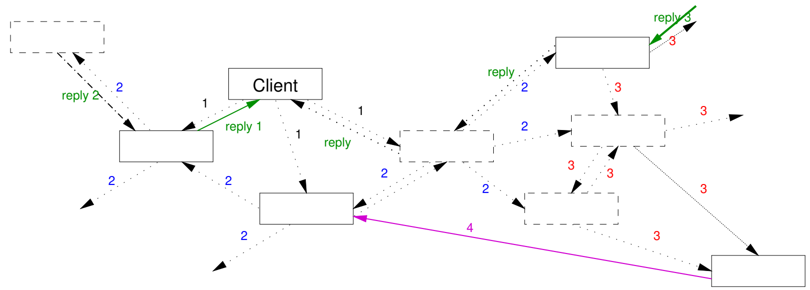
Gnutella: Suchstrategie (3)



- Anfrage erreicht alle sichtbaren Nachbarn
- in wenigen Zyklen

- Suchantwort läuft umgekehrten Weg zurück
- möglich wegen GUIDs und TTL-Sortierung
- Zurückverfolgen der Suche stark erschwert

Gnutella: Suchstrategie (4)



- Anfragen durchsuchen gesamtes Gnutella-Net
- Antwort kann aber recht lange dauern (mehrere Minuten)

- aktuelle Clients überlasten das Netz durch zu viele PINGs

Gnutella: Protokoll

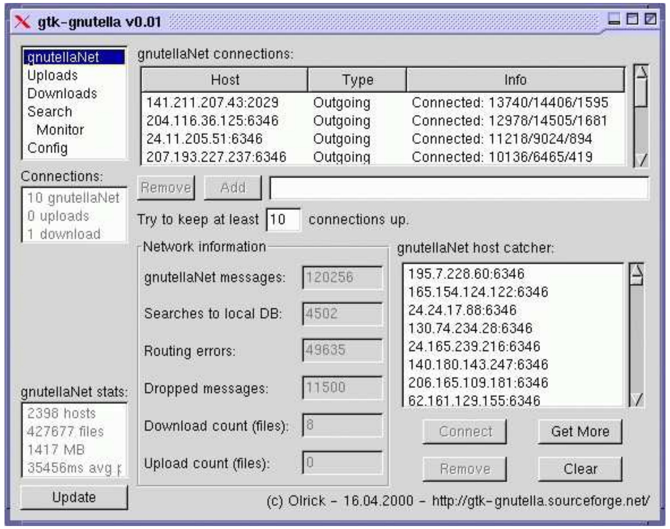
- textbasiertes Protokoll auf HTTP
- fünf "Pakettypen" definiert:

	Paket	Funktion
0x00	Ping	zur Suche nach Gnutella-Servants
0x01	Pong	Antwort eines Servants
0x80	Query	Suchanfrage nach Daten
0x81	QueryHit	Suchergebnis mit IP/Dateiname
0x40	Push	Datenübertragung via Firewalls

GUID	type	TTL	hops	plength	payload ...
0 – 15	16	17	18	19 – 22	23 – ...

- diverse Erweiterungen / Verbesserungen vorgeschlagen
- eigentlicher Download erfolgt direkt (peer-to-peer)

Gnutella: Screenshot (GTK-Version)



Gnutella: Pong, Query, QueryHit

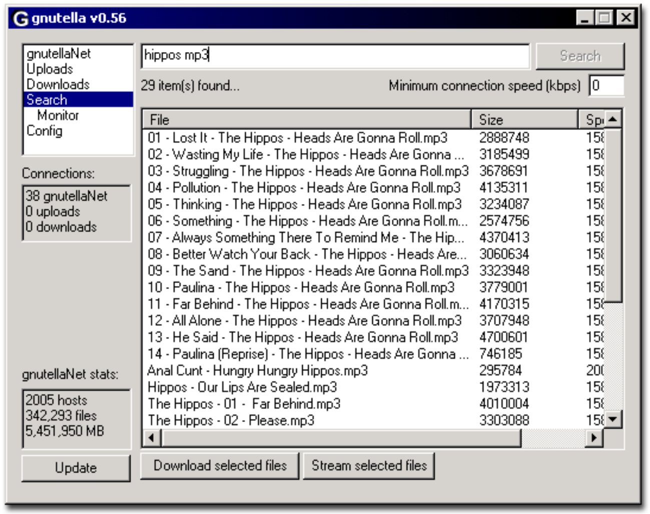
Port	Host IP	# shared files	# shared KBytes
0 – 1	2 – 5	6 – 9	10 – 13

min. speed	search criteria
0 – 1	2 – ...

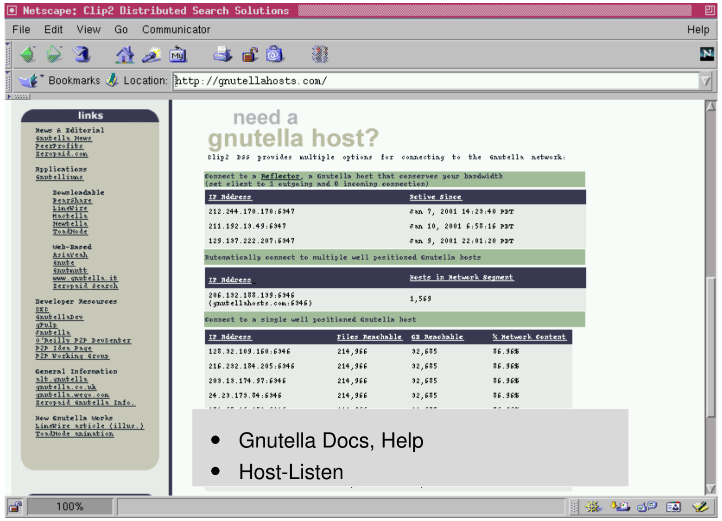
# hits	Port	Host IP	Speed	result set	servant ID
0	1 – 2	3 – 6	7 – 10	11 – ... n	n+16

- Host (=Servant) IPs in den PONG und QueryHit Antworten
- Zurückverfolgung also im Prinzip doch möglich

Gnutella: Screenshot Suchanfrage



Gnutella: Gnutellahosts.com



- Gnutella Docs, Help
- Host-Listen

Gnutella: vs. Napster

- jedes Gnutella-Client ist gleichzeitig Server
- und Teil der verteilten Suchalgorithmen

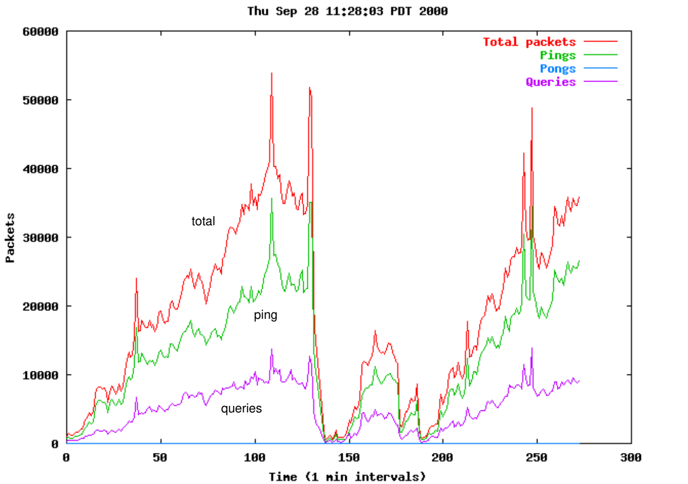
- Suche und Datentausch nicht auf MP3 beschränkt
- Protokoll "quasi" dokumentiert, Clients für viele Plattformen verfügbar

kein zentraler Server:

- keine Registrierung, keine Werbung, kein "single point of failure"
- kaum Angriffspunkte für Zurückverfolgung
- aber problematische Initialisierung (Finden des ersten Servers)
- weniger anwenderfreundlich als Napster

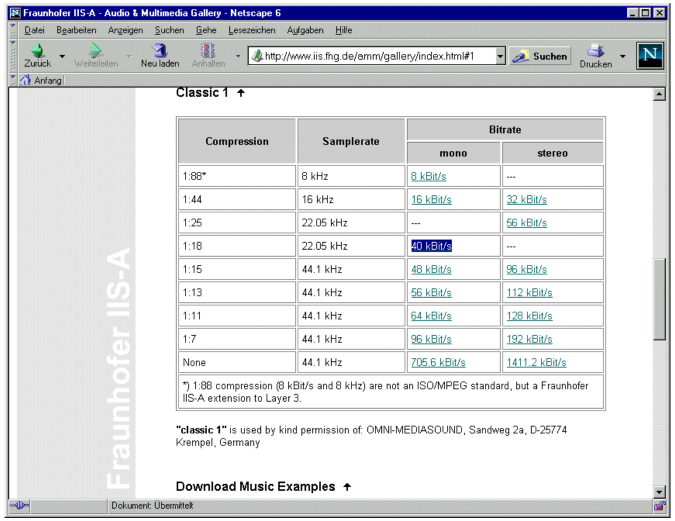
- verteilte Suche ist ineffizient aber robust
- Skalierbarkeit?! (jede Suche löst Paketflut aus)
- zukünftige Marktbedeutung?!

Gnutella: Traffic



[www.cs.berkeley.edu/~mdw/class/cs294-8/hw2.html]

Vergleich: Fraunhofer MP3-Demos

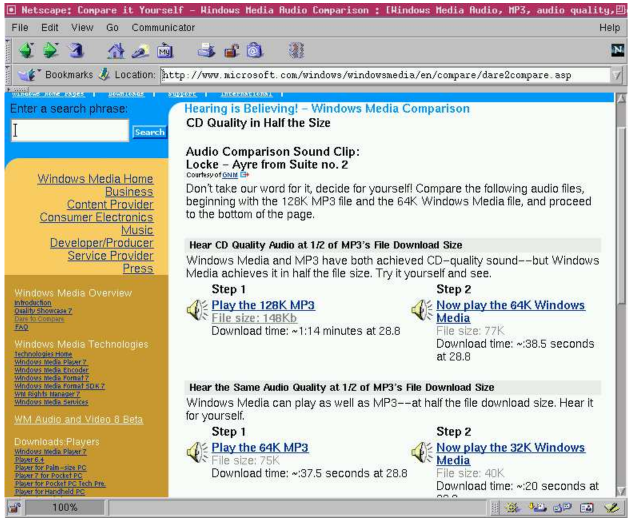


Compression	Samplerate	Bitrate mono	Bitrate stereo
1:88*	8 kHz	8 kBit/s	---
1:44	16 kHz	16 kBit/s	32 kBit/s
1:25	22.05 kHz	---	56 kBit/s
1:18	22.05 kHz	40 kBit/s	---
1:15	44.1 kHz	48 kBit/s	96 kBit/s
1:13	44.1 kHz	56 kBit/s	112 kBit/s
1:11	44.1 kHz	64 kBit/s	128 kBit/s
1:7	44.1 kHz	96 kBit/s	192 kBit/s
None	44.1 kHz	705.6 kBit/s	1411.2 kBit/s

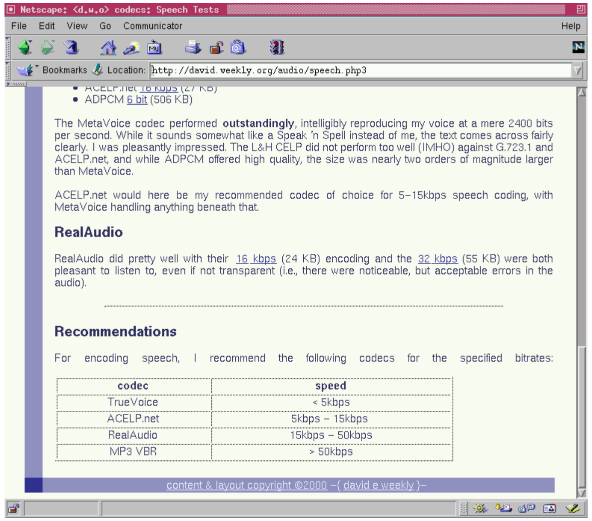
*) 1:88 compression (8 kBit/s and 8 kHz) are not an ISO/MPEG standard, but a Fraunhofer IIS-A extension to Layer 3.

"classic 1" is used by kind permission of OMNI-MEDIASOUND, Sandweg 2a, D-25774 Krempel, Germany

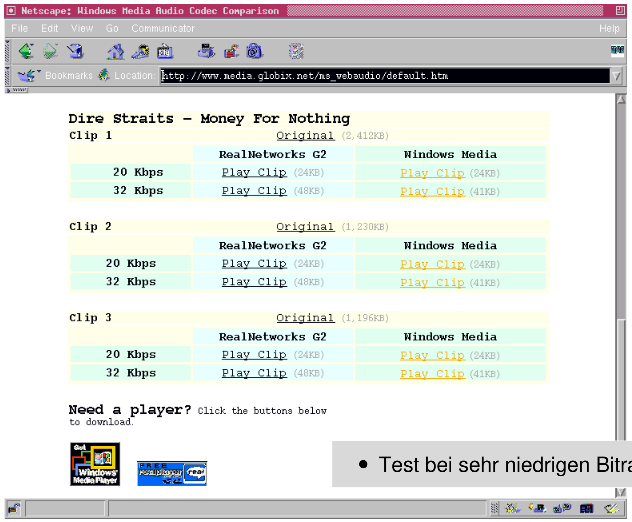
Vergleich: Microsoft WMA vs. MP3



Vergleich: Speech Codecs

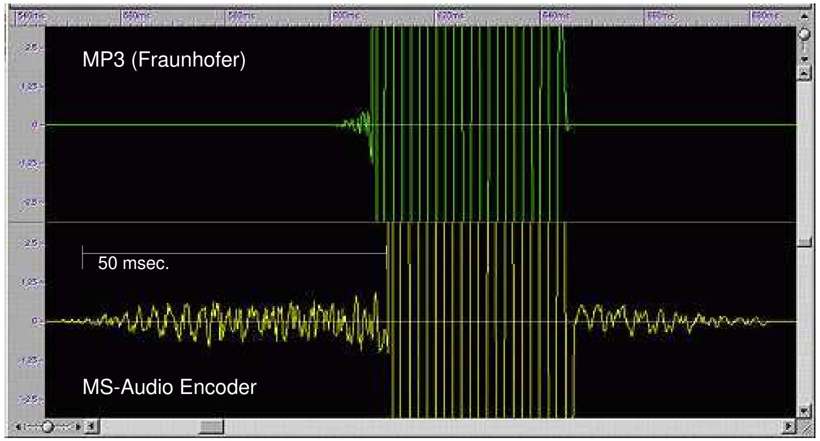


Vergleich: Microsoft WMA vs. G2



- Test bei sehr niedrigen Bitraten

Pre-Echos: MP3 vs. MS Audio



- krasses Beispiel für "pre-echos" (c't 99/10 p.52)
- aktuelle Encoder arbeiten (hoffentlich) besser