

# RTSP メソッドを用いた DVTS 制御機構の設計と実装

入野 仁志<sup>1</sup> 堀場 勝広<sup>1</sup> 小椋 康平<sup>1</sup> 杉浦 一徳<sup>3</sup> 中村 修<sup>2</sup> 村井 純<sup>2</sup>

慶應義塾大学大学院政策・メディア研究科<sup>1</sup> 慶應義塾大学環境情報学部<sup>2</sup> 独立行政法人通信総合研究所<sup>3</sup>

RTSP を用いたメディア再生の制御機能を内包した DVTS の設計と実装を行った。ネットワークを用い DV ストリーミングを可能にする代表的なアプリケーションとしての DVTS を応用し、映像送信の制御を可能とした。DVTS はこれまで映像の送信側、受信側で個別に動作し、連絡することが不可能であり利便性を損ねていた。本研究では映像の受信側と送信側で制御命令の送受を行い連絡を行い受信側が RTP セッションを開始出来るようになった。本研究では制御系として RFC に準拠した標準的な RTSP メソッドを利用し、汎用性を高めている。本実装によって DVTS の再生制御を行うと同時に、DVTS のインターオペラビリティを高めることが出来た。

## Design and Implementation of Media Controlable DVTS with RTSP

Hitoshi Irino<sup>1</sup>, Katsuhiko Horiba<sup>2</sup>, Kohei Ogura<sup>2</sup>, Sugiura Kazunori<sup>3</sup>, Osamu Nakamura<sup>2</sup>, Jun Murai<sup>2</sup>

<sup>1</sup>Graduate School of Media and Governance, Keio University

<sup>2</sup>Faculty of Environmental Information, Keio University

<sup>3</sup>Communications Research Laboratory

This research about design and implementation of extending DVTS including media control functions by RTSP. DVTS is a typical DV streaming application using broadband network. Existing DVTS doesn't implement any media control functions, furthermore DV sender and receiver runs individually. In this research DV sender and receiver exchange control packet based on RTSP standardized by RFC. Receiver can manage RTP session using extended DVTS implemented in this research. Extended DVTS has not only media control function but interoperability provided by adoption of RTSP.

## 1 はじめに

### 1.1 背景

広帯域データリンクが普及し、通信環境が向上している。WAN 環境におけるバックボーンの広帯域化のみならず、ラストワンマイルと呼ばれる一般家庭への接続回線において、ADSL や FTTH などの広帯域かつ常時接続の環境が一般化した。

広帯域データリンクが広く普及したことにより、これまで利用が困難であった時間あたりのデータ量が大きい、低圧縮で低遅延、高解像度、高フレーム数の高品質映像メディアを用いたコミュニケーションが可能となった。

高品質映像メディアを用いたコミュニケーションツールの代表例として DVTS(Digital Video Transport System)[1] があげられる。DVTS は安価に構築可能な高品質映像メディア転送システムとして数年前か

ら開発が行われ、現在では広く利用されている。

### 1.2 本研究の目的

現在の DVTS は送信側と受信側でそれぞれ別のプログラムが動作し、それぞれ独立した操作が必要である。その結果、使用者の敷居を高くしている。また、DVTS を応用した映像配信機構を構築する場合にも映像ソースに対する制御機構を持たないため、配信機構の構築の妨げとなっていた。

DVTS は映像の入出力に民生用の DV(Digital Video)機器を用いる。DV 機器の多くは外部機器からの制御が可能となっている。

本研究では DVTS に DV 機器の再生制御機構部分を拡張しその利便向上を計る。また、同時に DVTS を応用した映像配信機構の構築の敷居を下げ、高品質映像転送システムである DVTS をより一般化することも目的とする。

<sup>1,2</sup>Keio University Shonan Fujisawa Campus  
5322, Endo, Fujisawa, Kanagawa 252, Japan  
E-Mail: irino@sfc.wide.ad.jp

<sup>3</sup>4-2-1 Nukui-Kitamachi, Koganei, Tokyo, 184-8795 Japan

## 2 現状の DVTS について

DVTS は DV 機器から IEEE1394[2] インタフェースを用いて DV フォーマットのデータを読みだし、そのデータに対し IP パケット化を行い、転送を行うシステムである。DVTS は DV フォーマットを用いるため以下の特徴を持つ

- 高品質な映像の送受が可能
  - 現行 TV 相当以上の 720×480 の高解像度な画室で表示可能である
  - 最大で現行 TV 相当の 29.97fps の高いフレームレートで表示可能である
- フルフレーム時に片方向 1 ストリームで 30Mbps の帯域を要する

通信方式には下記の理由から UDP を用いる。

- 遅延を最小限にするためにほとんどバッファリングを行わない。そのため輻輳制御によるスループットの低下が起こると映像に乱れが発生する
- DV フォーマットはフレーム内圧縮方式であるため比較的データに欠損に強い。そのため多少のデータ欠損はコミュニケーションにほとんど支障をきたさない。

UDP はパケットの到達順序を保証しないため、パケット到達順序を把握する場合はアプリケーション層で行なう必要がある。DVTS では RTP(Realtime Transport Protocol)[3] を用いている。以上から DVTS が送受信するパケットを一般に DV/RTP[4] パケットと呼ぶ。

### 2.1 DVTS の構成

DVTS は送信側と受信側に分かれて動作する。送信プログラムは外部 DV 機器から IEEE1394 インタフェースを介して映像データを受信し DV/RTP 化して送信を行う `dvsend` である受信プログラムは DV/RTP の受信を行い IEEE1394 インタフェースを利用して外部 DV 機器にデータを送る `dvrecv`、計算機の画面描画を行なう `xdvshow` の 2 つがある。DVTS の動作概要を図 1 に示す

### 2.2 利用形態とその問題点

DVTS の利用形態としては以下の 2 つがある。受信者はあらかじめ `dvrecv` か `xdvshow` を起動し DV/RTP を受信することが可能な状態で待機し、送信側が後から DV/RTP の送信を始める。または、送信側が先行して `dvsend` から DV/RTP を送出し、受信側が都合のよい時に DV/RTP の受信を行う。前者は受信側からセッションを開始できず、後者はセッションへの参加自体は受信側からで行えるが、映像ソ-

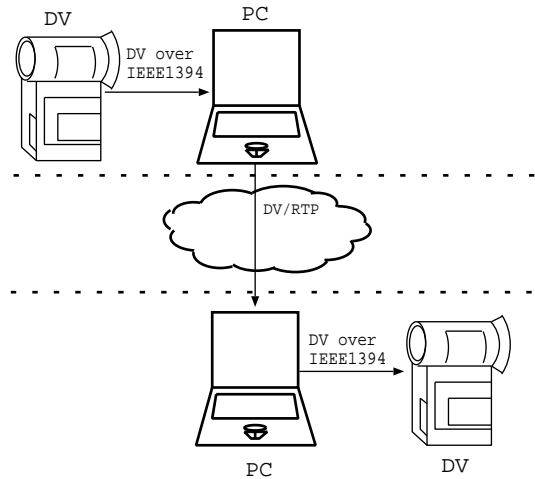


図 1: DVTS の動作概要図

スが DV テープの場合は、テープが先行して動作するため映像ソースを意図した位置からの再生することが出来ない。また受信側が DV/RTP の受信を始めるまでの間、約 30Mbps の無駄なトラフィックを発生させてしまう。

## 3 DVTS 制御機構の設計

第 2 章において既存の DVTS の問題点を述べた。以下に問題点を整理する。

- DV/RTP セッションを受信側から開始することができない
- 映像ソースに対する制御機構を持たない

この問題の解決に DVTS の動作を制御する通信を追加する。この通信を用いて映像受信側からの DV/RTP セッション開始、終了を可能にする。

### 3.1 DVTS 制御機構の設計要件

3 節で述べた制御通信を行なう機構を本研究では DVTS 制御機構と呼ぶ。DVTS 制御機構は既に述べた問題点から以下の機能を持つ必要がある。

- 映像受信側からの再生、停止要求に応じて DV/RTP セッションの制御を行なう。
- 映像受信側からの再生、停止要求に応じて外部 DV 機器が制御可能な場合、その制御を行なう。

### 3.2 DVTS 制御機構の設計

映像受信側から DV/RTP セッションの開始を行なうためには映像送信側で制御機構の通信を待ち受ける必要がある。従って送信側をサーバ、受信側をクライアントとしたサーバ・クライアントモデルで制御機構の通信を行なう。送信側の動作概要を図 2 に示す。

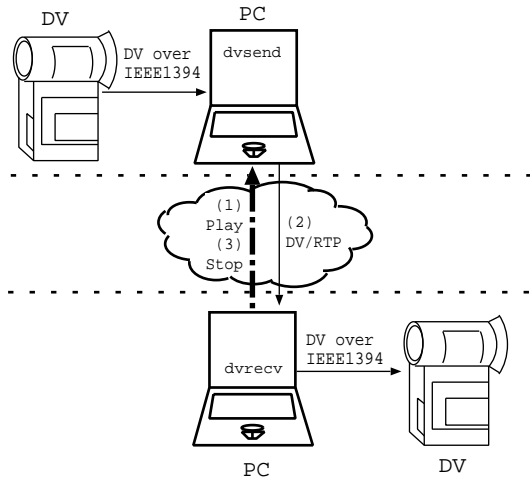


図 2: 制御機構を備えた DVTS の動作概要図

送信側の動作順序を以下に示す。

1. 再生要求の待ち受け (図 2中 (1))
2. 再生要求に従い DV/RTP セッションの開始 (図 2中 (2))
3. 停止要求に従い DV/RTP セッションの停止 (図 2中 (3))

### 3.3 制御機構の protocols: RTSP の利用

本制御機構では再生、停止といった制御命令をインターネットを介して送受を行なう。インターネット上のストリーミングメディアの制御プロトコルとして RFC2326 において RTSP (Real-Time Streaming Protocol)[5] が定められている。RTSP はクライアント/サーバ型のマルチメディアプレゼンテーション制御プロトコルであり、IP ネットワークでマルチメディアを効率よく配信することを目的として設計されている。RTSP は HTTP に良く似たテキストメソッドで構成されている。図 3 に RTSP の一例を示す。

```
PLAY rtsp://dvts.jp/dv.dv RTSP/1.0
CSeq: 835
Session: 12345678
```

図 3: RTSP の例: クライアントからサーバへ送られる PLAY メソッド

RTSP は停止、早送り、巻き戻し、指定した位置からの再生といった“ VCR 形式 ”の制御機能が定義されている。本研究で構築する DVTS 制御機構は DV/RTP セッションの再生、停止を行うことが

目的であるため RTSP はプロトコルスペックとして十分であると言える。RTSP は TCP, UDP の両方の通信が想定されているが、本機構は TCP を用いる。TCP を用いる理由は以下のとおりである。

- 制御命令は確実な到達性が確保されなければならない。
- 制御命令は単発的に発生する十分に小さいパケットの送受で行われる。そのため DV/RTP セッションと異なり輻輳制御が行われる事が特に問題にならない。

### 3.4 DV 機器の制御

送信側では DV 機器は IEEE1394 インタフェースを介して計算機と接続され映像の送受を行っている。IEEE1394 は映像や音声の送受の他に IEEE1394 機器の制御が可能である。IEEE1394 のデータ転送方式にはアイソクロナス (同期) 転送とアシンクロナス (非同期) 転送の 2 つの転送方式がある。映像・音声などの実時間性が重視されるデータ転送にはアイソクロナス転送を用い、IEEE1394 機器の制御命令のように実時間性はあまり重要でなく到達確実性が重要視される転送にはアシンクロナス転送を用いる。

IEEE1394 バス上の装置を制御するためのプロトコルとして FCP (Function Control Protocol) がある。IEEE1394 バス上の AV 機器に対する制御命令を AV/C コマンドと呼び、そのデータ構造を AV/C コマンドフレームと呼ぶ。AV/C コマンドフレームは前述した FCP フレームにカプセル化される。図 4 に FCP フレーム及び AV/C コマンドフレームを示す

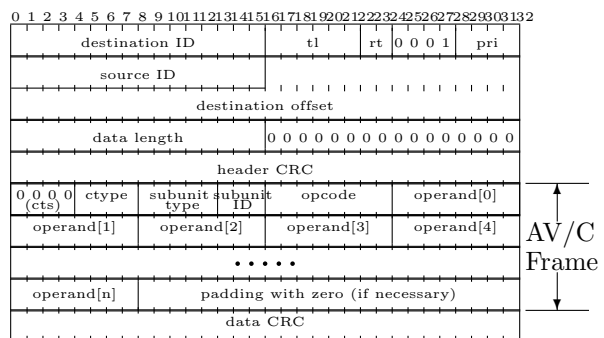


図 4: FCP フレーム及び内包される AV/C コマンドフレーム

機器を制御するためには制御する接続機器の ID を知る必要がある (図 4中 destination ID)。IEEE1394 バス上では機器の ID は接続時に自動的かつ自律的に決定されるため、この ID をその機器のユーザが決定することはできない。そのため本研究が想定し

ている遠隔からの制御を考えた場合、IEEE1394 バス上に制御可能な DV 機器がつながっているか否かを確認する必要がある。

DV 機器の判別は以下の条件で行う。

1. 機器が AV/C プロトコルに準拠しているか  
IEEE1394 機器が必ず持つコンフィギュレーション ROM 内のユニットディレクトリに機器の種類が記されている。ユニットディレクトリの `unit_spec_id` が `0xA02D` かつ `unit_sw_version` が `0x10000` もしくは `0x10001` のとき AV/C プロトコルに準拠した機器である
2. 機器が VCR, Camera, Recorder であるか  
機器が AV/C 機器であった場合、`subunit type` の値を用いることでさらに詳細な AV/C 機器の種類を判別できる。DVTS に用いられる機器は多くの場合 DV カメラである。そのため、`subunit type` が VCR, CAMERA, RECORDER であった場合に DV 機器がつながっていると判別する

### 3.5 設計のまとめ

本章で述べた事を以下に整理する。本機構が持つべき機能は以下の通りである

- 映像受信側からの再生、停止などの制御要求には RTSP を用いる。
- この制御要求に応じて映像送信側は DV/RTP セッションの制御を行なう。
- IEEE1394 バス上に DV 機器が存在するか否かを調べる
- 受信側からの再生、停止要求に応じて IEEE1394 バス上にある DV 機器が制御可能な場合、その制御を行なう。

## 4 DVTS 制御機構の実装

### 4.1 実装環境

実装に用いたソフトウェア環境を表 1 に示す。

表 1: 実装に用いたソフトウェア環境

OS	Linux 2.4.22 Kernel
使用ライブラリ	libraw1394 0.90 libavc1394 0.41
プログラミング言語	gcc 3.23
ベースとなる DVTS	DVTS 1.0a

Linu 上で IEEE1394 の任意のデータのアイソクロナス転送・アシンクロナス転送は `/dev/raw1394` デバイスファイルを介して行う。libraw1394 は Linux 上でのみで動作しユーザランドから `/dev/raw1394` デ

バイスファイルの扱いを簡易化するためのライブラリである。libavc1394 は IEEE1394 バス上の AV/C プロトコルに準拠した機器の制御及び情報の取得、IEEE1394 機器のコンフィギュレーション ROM からの情報の取得を簡易化するためのライブラリである

### 4.2 実装概要

既存の DVTS 送信プログラムである `dvsend` の拡張を行った。実装の概略を図 5 に示す。図 5 中、太線で囲まれた部分が、本実装による拡張部分である。既存の `dvsend` は実行時に宛先の IP アドレスなどをオプションとして指定するが、本実装は何もオプションがなかったときに、RTSP パケットを待機してメディア制御を行う状態にした。

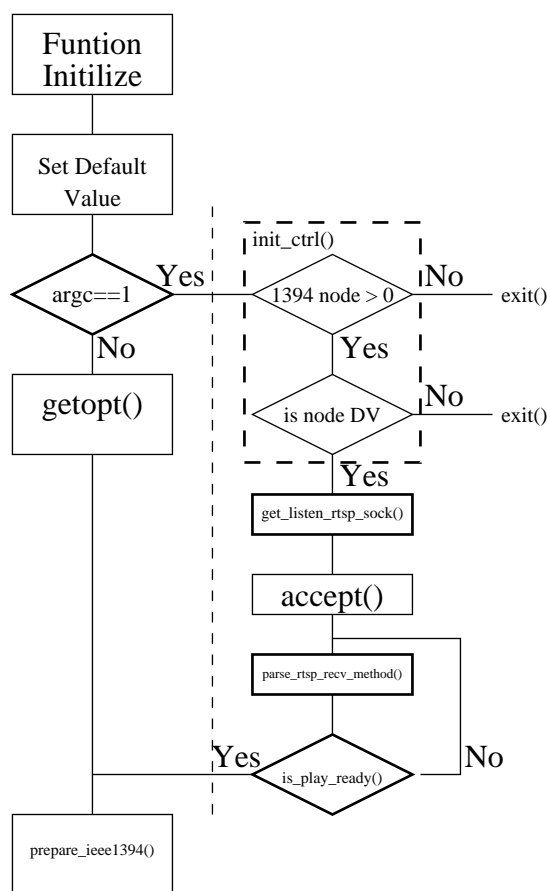


図 5: 本研究による新規実装部分の概略流れ図

新規に開発する部分は大きく分けて以下の 2 つに別れる。次節以降にこの 2 つの部分の詳細に関して述べる。

- IEEE1394 アシンクロナス転送の準備と IEEE1394 機器の判別を行う部分 (図 5 内 `init_ctrl()`)
- ソケットを介して RTSP を受信し、解析を行い、それに伴って `dvsend` の送信を開始する部分 (図

5内 `parse_rtsp_recv_method()`

### 4.3 IEEE1394 機器の判別

図 5内 `init_ctrl()` で行う処理を詳説する .

#### 1. handle の取得

`libraw1394` を用いた IEEE1394 機器との通信を行う際は , ファイルディスクリプタではなく `raw1394_handle_t` 型の `handle` と呼ばれるデータ構造を用いる . `handle` の取得は関数 `raw1394_new_handle()` を用いる .

#### 2. IEEE1394 バス上の接続機器台数の取得

IEEE1394 バス上に接続されている機器の台数を取得するために関数 `raw1394_get_nodecount` を用いる . この時複数台接続されていた場合は以降の処理を接続されている機器すべてに行い , 条件に合致した最初の機器を用いる .

#### 3. 機器の種別の判別

機器の種別の判別を行うためにまずコンフィギュレーション ROM の値を取得する必要がある . コンフィギュレーション ROM の値を取得には `rom1394_get_node_type()` を用いる . その結果 AV/C 機器であった場合には `avc1394_check_subunit_type()` を用いて機器の `subunit type` の値を取得し TAPE\_RECORDER, TAPE\_VCR, VIDEO\_CAMERA であるか調べる .

#### 4. 実際の機器の制御

実際の機器制御は `libavc1394` が提供する関数を用いる . DV の再生には関数 `avc1394_vcr_play()` , 停止には関数 `avc1394_vcr_stop()` を用いる .

`init_ctrl` のプリプロセッサ宣言を図 6 に示す . `init_ctrl` の第 2 引数である `dvsend_ctrl` 構造体と其中で用いられている `ieee1394_raw` 構造体を図 7 に示す . IEEE1394 上にデータを流すための API は各 OS で全く異なり , 本実装は Linux 上で行ったが , 既存の DVTS はその他の OS でも動作するマルチプラットフォーム化が行われているため , 将来的な本機構のマルチプラットフォーム化を考慮して , Linux 上のみには存在しない構造体を直接使うことを避けた .

```
int
init_ctrl __P((struct dvsend_param *,
              struct dvsend_ctrl *));
```

図 6: `init_ctrl` のプリプロセッサ宣言

```
struct dvsend_ctrl{
    struct ieee1394raw ieee1394raw;
}
struct ieee1394raw {
    int ch;
    raw1394handle_t handle;
    int raw1394_port;
};
```

図 7: `dvsend_ctrl` 構造体及び `ieee1394_raw` 構造体

### 4.4 RTSP パケットの受信とその解析

本節では図 5内 `parse_rtsp_recv_method()` で行う処理に関して述べる . RTSP の様々なメソッドのうち RFC2326 において "required" とされている "OPTIONS", "SETUP", "PLAY", "TEARDOWN", に関して実装を行った実装を簡易化し移植性を高めるために POSIX `regex` (正規表現) 関数を用いた . 処理内容は大きく分けて以下の流れになる .

- メソッドタイプの取得 `read()` システムコールを用いて , パケットを受信しそのパケットを正規表現を用いて "OPTIONS", "SETUP", "PLAY", "TEARDOWN" のメソッドが含まれているか判別する . またその後続く RTSP の url および , RTSP の version を判別する .
- CSeq の値の取得 CSeq は RTSP のリクエスト-レスポンスの組に付けられる連続的な番号である . この値を取得し一つ前に受け取った RTSP パケットと比較して値が 1 増加しているかを検査する .
- クライアントの RTP セッションの要求の取得 SETUP メソッドの場合はそのあとに続く Transport Option をもとに , クライアントが希望している RTP セッションがユニキャストかマルチキャストかの情報を取得し , クライアントの希望する RTP セッションに用いるポート番号を取得する .
- クライアントへのレスポンス SETUP メソッドの場合は必ずレスポンスを返す必要がある . クライアントからの要求を元にセッション番号と時間情報を生成して応答を返す .

## 5 評価

本章では , 本実装の RTSP 部分の定性評価を行う .

## 5.1 本実装の RTSP の RFC への準拠

本研究において実装を行った RTSP メソッドと RFC において規定されている RTSP メソッドとの対応関係を表 2 に示す。

表 2: RFC における要求と本実装との対応表

RTSP METHOD	実装状況	RFC での要求
OPTIONS		必須
DESCRIBE	×	推奨
ANNOUNCE	×	任意
SETUP		必須
PLAY		必須
PAUSE	×	推奨
TEARDOWN		必須
GET_PARAMETER	×	任意
SET_PARAMETER	×	任意
REDIRECT	×	任意
RECORD	×	任意

本実装は最低限のメソッドを実装しており、RFC に準拠しているといえる。

## 5.2 パケットアナライザを用いた本実装の RTSP パケットの正確さの評価

本機構を用いて RTSP 通信を行い、同時にパケットアナライザツールである ethereal[6] を用いてパケットダンプを行った。ethereal のデコード機能を用いて TCP パケットを RTSP パケットフォーマットに準じて表示をさせ、正しく表示された。図 8 にその結果の一例を示す。

## 5.3 既存 RTSP クライアントとの通信

本実装と既存の RTSP との通信を行った、既存クライアントには openRTSP[7] を用いた。openRTSP との通信は正常に行う事ができなかった。その理由は openRTSP は DESCRIBE メソッドが実装されており、RTSP サーバからの sdp セッションを待つためである。本実装は 5.1 節で述べた最低限の実装しか行っていないため DESCRIBE メソッドは未実装である。

## 5.4 定性評価のまとめ

本実装は RFC での要求事項のうち最低限のメソッドに関して実装されている。また本実装が生成する RTSP パケットも正確であるといえる。しかしながら既存クライアントを利用するためには不十分であり、他のメソッドの実装が必要といえる。

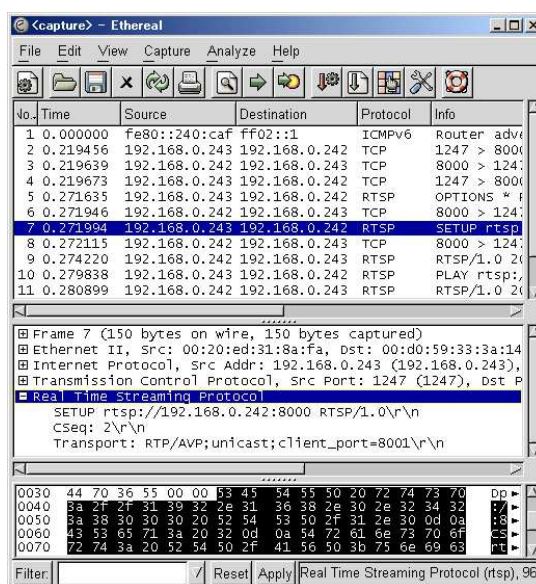


図 8: ethereal によるパケットダンプと RTSP パケットの表示

## 6 まとめ

本研究では DVTS が受信側から RTP セッションを開始できない問題に着目し、メディア制御機構の設計と実装を行った。制御系のプロトコルとして RFC で標準化が行われている RTSP を用いた。RFC の要求事項を満たしていることを評価しインターオペラビリティを確保した。今後は RFC が推奨する RTSP メソッドを実装することで、よりインターオペラビリティが向上し、より実用的になる。

## 参考文献

- [1] Akmichi Ogawa. DVTS(Digital Video Transport System). <http://www.sfc.wide.ad.jp/DVTS/>.
- [2] Institute of Electrical and Electronics Engineers, Inc. IEEE Std 1394-1995 High Performance Serial Bus. Aug 1996.
- [3] Audio-Video Transport Working Group. RFC1889 RTP: A Transport Protocol for Real-Time Applications. <http://www.ietf.org/rfc/rfc1889.txt>, pages 1-75, January 1996.
- [4] Akimichi Ogawa, Katsushi Kobayashi, Kazunori Sugiura, Osamu Nakamura, Jun Murai. Design and implementation of dv based video over rtp. <http://www.sfc.wide.ad.jp/DVTS/pv2000/index.html>.
- [5] H. Schulzrinne, Columbia U., A. Rao, Netscape, R. Lanphier, RealNetworks. Request for comments: 2326, real time streaming protocol (rtsp). <http://www.ietf.org/rfc/2326>, 1998.
- [6] <http://www.ethereal.com>.
- [7] <http://www.live.com/openRTSP/>.