

IP 網における DV 動画像伝送プログラムの改良

光枝 慶一[†] 都築 伸二[†] 山田 芳郎[†]

[†] 愛媛大学工学部電気電子工学科 〒790-8577 愛媛県松山市文京町3

E-mail: †{k-mitu,tsuzuki,yamada}@sarah.ee.ehime-u.ac.jp

あらまし 本稿では、筆者らが開発を進めている、デジタルビデオ (DV) カメラの高精細動画像を、IP 網を介してリアルタイムに伝送するプログラムにおいて、ユーザインタフェース・伝送方式・パケットロス対策について改良したのでその内容について述べる。また、同様のソフトウェアとして広く認知されている DVTS との機能比較を示す。

キーワード ビデオ会議システム, DV over IP, IEEE1394, RTP, IPv6

Improvement of DV Transmission Program over IP Network

Keiichi MITSUEDA[†], Shinji TSUZUKI[†], and Yoshio YAMADA[†]

[†] Faculty of Engineering, Ehime University Bunkyou-tyou 3, Matsuyama-shi, Ehime, 790-8577 Japan

E-mail: †{k-mitu,tsuzuki,yamada}@sarah.ee.ehime-u.ac.jp

Abstract The authors have been developing a real-time transmission program of digital video (DV) data from a high quality video camera over IP networks, and upgraded recently. In this paper, the upgraded points, that is user-interface, communication protocols, and a solution of packet loss problems are reported. Moreover, the developed program is compared to DVTS that is a well-known program and provides similar functions to our program, and superior points of the proposed program are explained.

Key words Video conference system, DV over IP, IEEE1394, RTP, IPv6

1. ま え が き

筆者らは、汎用の Windows PC を用いて、デジタルビデオ (DV) カメラで撮影した高精細動画像を、高速 IP 網を介してリアルタイムに伝送することを目的としたプログラムの開発を行っている [1], [2].

NTSC クラスの動画品質の符号化方式としては MPEG2 あるいは DV (Digital Video) がある。前者は網の伝送容量に応じて、1M~15Mbps の範囲で画像圧縮の程度を変更できる特長があるものの、高効率な符号化を行うため、画像のフレーム間予測等を行っており、画像のコマ送りやオーサリングを行う場合は DV に劣る。

一方、DV はフレーム間予測を行っていないためオーサリング用途に適するものの、フルレートでは 30Mbps 程度のストリームデータとなるため、インターネットでの配信が困難であり、JGN (Japan Gigabit Network) 等の専用回線や高速 LAN 内で用いられることが多かった。しかし、近年では FTTH などの高速なアクセス回線が普及してきており、これらの回線を用いた DV 伝送の可能性が出はじめています。

汎用の PC で動作する DV の伝送ソフトウェアとしては、DVTS [3] があげられるが、筆者らは、(株)FA システムエンジニアリング社 (以下、(株)FA SE) ^(注1) と共同で、以下のようなコンセプトに基づいて、独自の DV 動画像の伝送装置を開発しており、現在、下記 (5) に注力して開発を行っている。

- (1) DVTS のように DV データの配信に特化せず、DV データのパーソナルなネットワーク・アクセスツールを実現
- (2) 汎用の Windows PC で動作し、アプライアンス化できるようにする
- (3) IEEE1394 インターフェースから取り込んだ DV カメラ等のストリームデータを IP 網に送信
- (4) 受信データは PC のディスプレイや IEEE1394 インターフェースに出力
- (5) 双方向で使う場合は、パーソナルなデスクトップビデオ会議を実現
- (6) 単方向で使う場合は、DV データをネットワーク越しにランダムアクセスする VOD (Video On Demand) システム

(注1) : <http://www.fase.co.jp/>

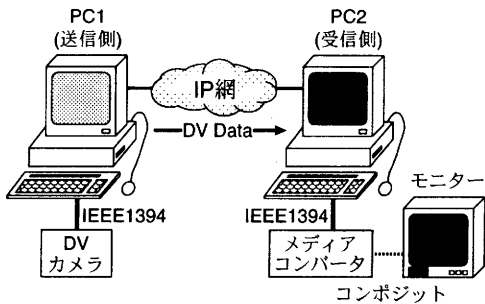


図1 使用機器の構成
Fig. 1 Proposed DV Transmission System

ムや、データを遠隔編集するシステムを実現

なお、IEEE1394インターフェースのドライバ周辺および受信データをディスプレイに描画するソフトウェア (Windowsのdirect showを使用)、本プログラムを動作させるためのPC(アプライアンス化するために、コンパクトフラッシュで起動する組み込み用小型PCを開発。その形状から、以下、PC-Cubeと呼ぶ)を(株)FASEが、IP網での伝送ソフトウェア (WinSockを使用)を筆者らが担当している。

筆者らが担当するIP網での伝送ソフトウェアは、ベストエフォート型のIP網を想定しており、輻輳によるパケット欠落(以後、パケットロスと呼ぶ)や、パケットの到着時間の揺ぎによる遅延時間の累積が問題となる。通常、ストリームデータはUDPで送信するため、パケットロスが生じても再送しない。従って画像の一部が欠落してブロックノイズとなったり、音声途切れたりクリックノイズが発生する。また、遅延時間が累積していけば、ビデオ会議のような対話型通信が困難になるため、違和感なく対話できるように150m~200m秒以下の遅延に抑える必要がある。これらの問題に対して、文献[1]では、伝送レートの変換や無音検出による遅延時間の短縮方法を提案した。

本稿では、文献[1]以降に行った、プログラムの改良点(ユーザーインターフェース・伝送方式・パケットロス対策)及び、DVTSとの機能比較について述べる。

2. 使用機器の構成

本研究で使用したPC周辺の構成を図1に示す。送信側のPC1にはIEEE1394インターフェースによってDVカメラが接続しており、IP網を介して受信側のメディアコンバータに接続したモニターやPC2のディスプレイに動画像を表示する。また、表1には本プログラムで動作確認したPCのスペックを示す。いずれも送受信同時に行うことができるスペックであり、タスクマネージャで見た送受信時のCPU使用率とメモリ使用量も示した。ネットワークインターフェースは全て100base-TXで、IEEE1394ボードはOHCI(Open Host Controller Interface)準拠のものである。また、DVカメラはSONY製DCR-TRV30を、メディアコンバータはCanopus製ADVC-100を使用した。

表1 使用したPCの仕様
Table 1 Specification of Used PCs

	PC-A	PC-B	PC-C
メーカー	DELL	日本 Gateway	(株)FASE
CPU	PentiumIV 2.4GHz	Pentium IV 1.4GHz	Pentium M 1.1GHz
メモリ	512MB	256MB	512MB
グラフィック	NVIDIA GeForce4 Ti4200	RADEON 7200 AGP	Intel 855GM
OS	WindowsXP Home Edition	Windows2000 Professional	Windows2000 Professional
CPU使用率	50%以下	100%	50%以下
メモリ使用量 (送受信実行前 /同時実行中)	169MB / 244MB	107MB / 169MB	77MB / 157MB

3. DV動画像伝送プログラム

3.1 ユーザーインターフェース

文献[1]執筆時には本プログラムのユーザーインターフェースはCUI(Character User Interface)であったが、よりユーザーフレンドリーなGUI(Graphical User Interface)への変更を行った。

本プログラムのユーザーインターフェースを図2に示す。図の左半分が送信プログラム用で、右半分が受信プログラム用である。GUIにより、ポート番号やプロトコル、送信元などの指定ができるようになっており、デバッグウィンドウや設定メニューは表示/非表示の切替えができるようにした。

また、受信した動画像をPCのディスプレイに表示するためのプログラムのユーザーインターフェースを図3に示す。このプログラムは、(株)FASE製で、受信した動画像の録画や、録画した動画像をサムネイルとして管理することができる。

図2に示したプログラムを、それぞれ送信側PCと受信側PCで実行すると、DVの伝送が行われ、受信側PCのIEEE1394インターフェースから受信したデータが出力される。ここでPCのディスプレイにも表示したければ、図3に示したプログラムを実行すればよい。

3.2 DVフォーマット

本プログラムの特長である、レート変換や無音検出アルゴリズムでは、DVのデータの性質を用いている。そこで本節では、DVのデータフォーマットについて説明する[2],[4]。

DVデータは、80ByteのDIFブロックが複数集まって構成される。150個の連続したDIFブロックの集合をDIFシーケンスと呼び、さらに10個のDIFシーケンスで画像1フレームが構成される。各DIFブロックは、ヘッダ3ByteがIDと呼ばれ、残りの77Byteにデータが格納されている。図4にDIFブロックのデータ構造を示し、以下にDIFブロックのIDの詳細を示す。

第0バイト

0-2ビット: DIFブロックの種類

(000:HEADER, 001:SUBCODE, 010:VAUX,

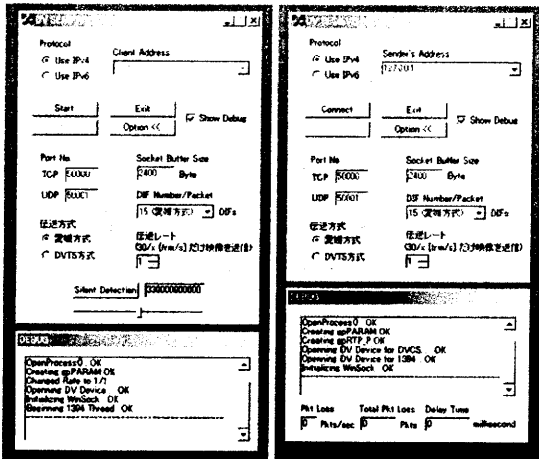


図2 ユーザインターフェイス
Fig.2 User Interface

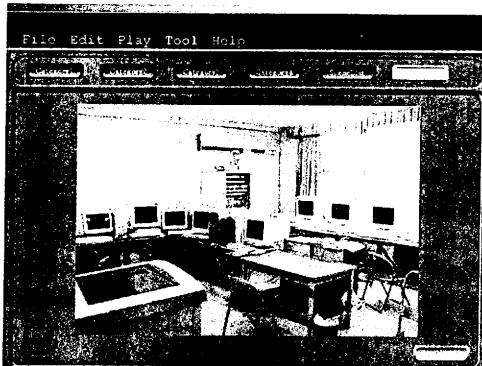


図3 受信動画画像の表示例
Fig.3 Display Image of Received DV-Data

011:AUDIO, 100:VIDEO)

3ビット: デフォルト1

4-7ビット: 任意(1,0どちらでもよい)

第1バイト

0-3ビット: DIFシーケンス番号

(0-9の2進表示で, DIFシーケンス毎に1ずつ増加)

4ビット: ビットレート(0:25Mbps, 1:50Mbps)

5-7ビット: デフォルト1

第2バイト

0-7ビット: DIFブロック番号(0-134の2進表示)

また, DIFシーケンスは, HEADER:1ヶ, SUBCODE:2ヶ, VAUX:3ヶ, AUDIO:9ヶ, VIDEO:135ヶの, 計150ヶのDIFブロックから構成されており, 図5に示すように並んでいる。

3.3 RTP準拠のヘッダ方式

文献[1]執筆時には, 独自形式のペケットヘッダを用いていたため, DVTSとの互換性はなかったが, 本稿ではDVTSとの送受

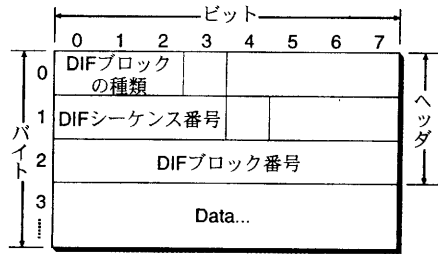


図4 DIFブロック
Fig.4 DIF Block

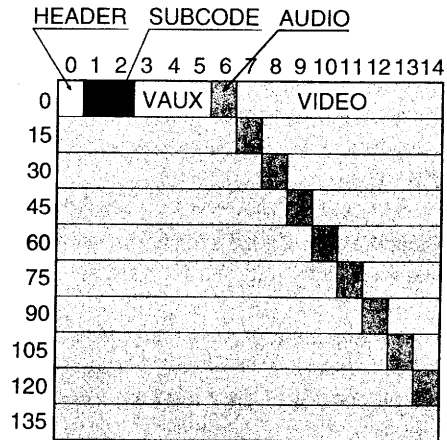


図5 DIFシーケンス
Fig.5 DIF Sequence

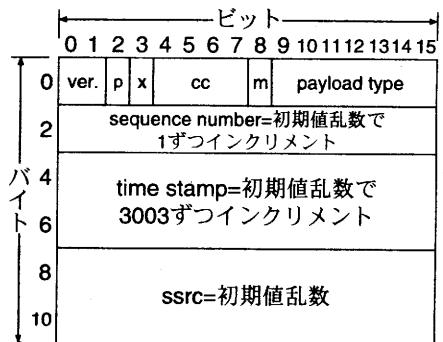


図6 RTPヘッダのフォーマット
Fig.6 Format of RTP Header

信も行えるようにするため, ペケットッダ部をRTP (Real-time Transport Protocol) 準拠の形式へ変更し, このRTPヘッダによりフレームの再構築を行うアルゴリズムに変更した。図6にRTPヘッダの形式を示し, 各フィールドの説明を以下に示す[5]。ただし, 各項目中の()の値は本プログラムでのデフォルト値である。

ver. RTPのバージョンを示す(2)

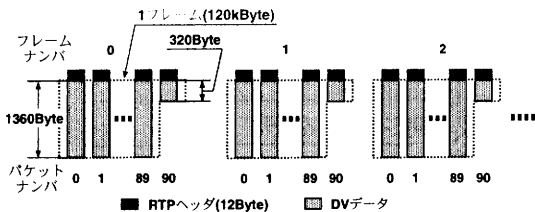


図7 DVデータのケットフォーマット

Fig. 7 Packet format of DV data

- p このパディングビットがセットされる場合、ケットは、そのペイロードの最後の部分にパディングオクテットを含む(0)
- x この拡張ヘッダビットがセットされる場合は、固定ヘッダに続いて拡張ヘッダがある(0)
- cc このCSRCカウントは、固定ヘッダに続くCSRC識別子の数を示す(0)
- m このマーカビットはフラグメントされたDVフレームの最後のケットでセットされ、DVフレームの境界を示す payload type RTPペイロードに含まれるDVデータのフォーマットを示す(0)
- sequence number 各RTPケット毎に1ずつインクリメントする。本プログラムではケットナンバーとして使用し、ケットロスのカウントもこの値を用いて行う
- time stamp 各DVフレームの最初のデータがサンプリングされた時間を示し、本プログラムではフレームナンバーとして用いられる
- ssrc 同期元の識別のために用いるフィールドであるが、本プログラムでは遅延時間測定のために時間情報を格納する

このケットヘッダ部をRTPに準拠させ、第3.4節で述べるDVフレームのフラグメントサイズを可変にすることにより、DVTSとの送受信が可能になった。

3.4 DVフレームのフラグメント

本ソフトウェアでは、画像1フレーム分のDVデータ(120k-Byte)を、指定したサイズにフラグメントして送信する。さらに、フラグメントされたDVデータには、第3.3節で述べたRTPヘッダ(12[Byte])を付けており、このヘッダに含まれる情報を元にDVフレームの再構成やケットロスと遅延時間の測定を行えるようにした。

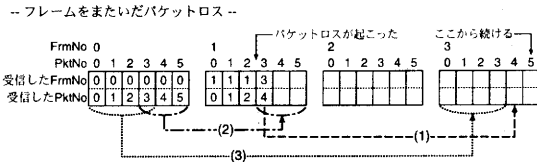
図7は、1ケットに含まれるDIFの数が17個になるようにフラグメントした場合である。この時、1フレームのフラグメント数は89個で、フレームの最後のケット以外、つまりケットナンバー0~87のケットサイズは、

$$12 [\text{Byte}] + (17 [\text{DIF}] \times 80 [\text{Byte}]) = 1372 [\text{Byte}]$$

となる。しかし、フレームの最後のケットは、

$$12 [\text{Byte}] + (120 [\text{kByte}] - 1360 [\text{Byte}] \times 88) = 332 [\text{Byte}]$$

となり、他のケットのサイズとは異なる。ここで、1ケットに含まれるDIFの数を変えて、例えば10などにすれば、全てのケットのサイズを等しくすることも可能である。



-- フレームをまたいだケットロス --

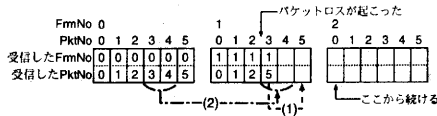


図8 ケットロス対策

Fig. 8 Solution of Packet Loss

3.5 IPv6への対応

次世代プロトコルとしてIPv6 (Internet Protocol version 6)の研究開発が進められており、OS及びネットワーク機器のIPv6対応が進んでいることから、本プログラムのIPv6への対応は必須であると考え、実装を行った。

具体的には、ソケットアドレス構造体をIPv6アドレスを扱えるものに変更し、いくつかのIPv6用関数を組み込んだ[6]。IPv4/v6の変更は、ポート番号や送信元の指定と同様に、プログラム実行時に図2のGUIで設定できるようにした。

3.6 ケットロス対策

本研究で対象としているベストエフォート型のIP網において、輻輳によりケットロスが生じる場合は、DVの伝送レートを変更し、伝送データ量を減らすことによってロス率を軽減できる。しかし、伝送レートを落とすとしても、ベストエフォート型のIP網を使用する以上、ケットロスの発生を防ぐことはできない。そこで、本プログラムでは軽微なケットロスが発生した場合には、以下で述べるように過去に受信したフレームで、損失した部分を補間する機能を実装した。

本プログラムでは、ケットロスについて「フレームをまたいだケットロス」と「フレーム内でのケットロス」に分けて考える。実装したケットロスの対策例を、図8に示し、このアルゴリズムを以下に示す。ただし、図では簡単化のため、1フレームを6個にフラグメントしたとする。

- ケットロスが起きても、一旦間違った場所にデータを格納する。
- 間違った場所に格納されたデータを正しい位置に移動する(図中(1))。
- 過去に正常に受信しているフレームから、損失部分に相当するデータをコピーして補間する(図中(2),(3))。

以上のアルゴリズムにより、少数のケットロスであれば、実用上問題ない程度に修復できるようになった。しかし、ケットロスが連続して発生し、かつ画面のシーンが変わった場合などは画像が乱れて見え、また音声がとぎれてしまう。この場合はさらに伝送レートを落として対応する必要があると思われる。

表2 DVTSとの機能比較

Table 2 Functional Comparison with DVTS

○：対応 / ×：未対応 / △：対応予定

機能	DVTS (UNIX版) (※1)	DVTS (Windows版) (※2)	本プログラム
(1) IPv6での送受信	○	○	○
(2) マルチキャスト送受信	○	(※3)	△
(3) 可変伝送レート	(※4)	(※4)	(※4)
(4) パケットロスの表示	○	○	○
(5) 使用するポート番号の指定	○	○	○
(6) パケットサイズの変更	○	×	○
(7) 音声冗長機能	○	×	△
(8) 送信プレビュー	×	○	△
(9) 受信時画面表示	(※5)	○	○
(10) 受信時IEEE1394出力	○	×	○
(11) 録画機能	×	×	○
(12) 遅延時間の表示(※6)	×	×	○
(13) 無音検出による遅延時間の短縮(※7)	×	×	○

3.7 DVTSとの機能比較

DVTSと本プログラムとの機能比較を表2に示す。ただし、図中の(※)については以下に示すとおりである。本プログラムは機能(11)、(12)、(13)が特長である。

- (※1) FreeBSD, Linux, MacOS X用のUNIX版 dvts1.0a
- (※2) Windows2000, XP用のWindows版(2002年7月11日版)
- (※3) TTLが1のみ
- (※4) 手動で変更可
- (※5) 別のプログラムで実現
- (※6) RTPヘッダに時間情報を付加して送信し、遅延時間の表示を行う機能
- (※7) 音声がある一定の閾値以下のフレームを引くことによって音声の途切れを感じさせずに遅延時間を短縮させる機能[1]

4. おわりに

本稿では、文献[1]以降に行った、プログラムの改良点を述べ、DVTSとの機能比較を行った。

今後の実装としては、伝送レート変換と遅延短縮法[1]を組み合わせ、たアルゴリズムで動作するプログラムに改良する予定である。

本稿では、1対1で片方向の伝送のみについて述べたが、1台

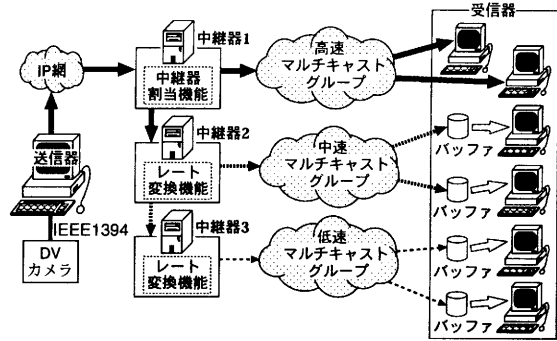


図9 DVストリームの放送型中継システム
Fig. 9 Broadcast Relay System of DV Stream

のPCで、送受信両方のプログラムを実行することによって双方向での伝送も可能である。また、図9に示すように、本プログラムで送信したDVストリームの中継し、その中継点でレートの変更やユニキャスト/マルチキャストの中継を行うことで、ユーザのアクセス回線に見合った配信が可能と考えている[7]。

さらには、多地点ビデオ会議システムの開発や、本プログラムをアプライアンス化する際には、現在のGUIをWebベースなどリモートから操作できるものにする必要がある。

謝 辞

本研究を進めるにあたり、共同研究を行っている(株)FAシステムエンジニアリング社様に深く感謝します。またJGN回線を使用した評価実験にご協力して頂いている、北陸先端科学技術大学院大学 丹康雄助教授に感謝致します。

本研究は、JGNプロジェクトNo.JGN-R13024に基づいて実施している。

文 献

- [1] 光枝, 都築, 平屋, 小池, 山田, “IP網におけるDV動画画像伝送装置開発のための基礎検討”, 電子情報通信学会 通信ソサイエティ, インターネットアーキテクチャ研究会, 信学技報IA2002-21, p.19-24, Oct. 2002.
- [2] 小池 泰樹, “IP網におけるリアルタイム動画画像伝送方式に関する研究”, 愛媛大学大学院理工学研究科博士前期課程電気電子工学専攻学位論文, Feb. 2002.
- [3] 小川見通, “DV伝送システム”, <http://www.sfc.wide.ad.jp/DVTS/index-j.html>.
- [4] 久保田 幸雄 編著, “図解 デジタルビデオ読本”, 株式会社 オーム社, 東京, 1996.
- [5] “RTP: A Transport Protocol for Real-Time Applications(RFC 1889)”, <http://www.faqs.org/>.
- [6] W. リチャード・スティーヴンス 著, 篠田 陽一 訳, “UNIXネットワークプログラミング 第2版 Vol.1”, 株式会社ピアソン・エデュケーション, 東京, 2001.
- [7] 井口, 光枝, 都築, 山田, “DV動画ストリームの放送型中継システムの設計”, 平成15年度電気関係学会四国支部連合大会講演論文集, p.196, Oct. 2003.