

VDIF/SUDP Data Transfer Software quick Startup Guide

Ver.1.1 2011.9.12 M.Sekido

1. はじめに

VDIF/SUDP のソフトウェアパッケージは、VLBI の大容量データを長距離伝送するためのソフトウェアである。以下に述べるようにデータフォーマットに VDIF を使用し、SUDP というプロトコルを使って長距離のデータ伝送を行う。ソースコードは C++ で書かれており、GCC などの標準的なコンパイラでコンパイル可能である。TCP/IP、UDP/IP のネットワーク部のコードには、David Lapsley 氏が開発した vtp のクラスライブラリを改修して使用している。

送信サーバ `sdup_send2` と受信サーバ `sdup_recv3` は外部から TCP/IP のコマンドを受けとり、送信、受信モードの設定や送信・受信の開始、停止などが制御できる。また、受信側では、VDIF、K5/VSSP32 のデータフォーマットで記録を行う。

データ伝送の構造:UDP パケットのペイロード先頭に 8 バイトのヘッダをつけ、ヘッダには 4 Byte の sequential number とパケットサイズを含む。このヘッダ情報により、パケットの重複や損失を計測することができる。以下、この UDP/IP 上の伝送プロトコルを SUDP (Simple UDP) と呼ぶことにする。SUDP は伝送プロトコルであり、受信側でペイロード部分だけを抜き出して使用する。

SUDP のペイロードには、VLBI データを VDIF フォーマット[1]で搭載する。VDIF フォーマットは、時刻、チャンネル数、などのデータが含まれており、1 パケットごとにデータの取得時刻を正確に同定できる。

2. ソースコード

CVS レポジトリ：vsspfs1.nict.go.jp/cvs/VDIF

tar アーカイブの場所：<http://www3.nict.go.jp/w/w114/stsi/e-VLBI/VDIF-SUDP/>

3. インストール手順

(1) VDIF と vtp-1.0.Hm のソースコードを

`tar ball (VDIFyyyymmdd.tgz)` は `cvs` で取得する。

置き場所は、オリジナルでは `$(HOME)/PROJ/VLBI/e-VLBI/` 以下

(2) `cpp` のライブラリ (`cpp-lib.tgz`) を `$(HOME)/prog/cpp/` 以下にコピーする

(3) `$(HOME)/tool` のディレクトリを作成し、`path` の環境変数に加えておく。

(4) `cd $(HOME)/PROJ/VLBI/e-VLBI/VDIF/app; make all;`

でバイナリファイルを生成する。

`make link` で `$(HOME)/tool/` にシンボリックリンクを作成する。

4. 使いかた

(1) 送信側で (K5/VSI からデータを取得しネット送信するものとする。)

```
sdup_send2 -R 200 -d VSI
```

としてサーバを起動する。それぞれのオプションの意味は以下のとおり

- d VSI: K5/VSIからデータを取得し、伝送するオプション
- R 200: 伝送レートを調整する頻度 (Hz) 200Hz で調整を行う

(注 1) ルート以外のユーザがコンパイルした直後は/dev/mem などルート権限でなければアクセスできないファイルへのアクセス権がないため、エラーメッセージを出して停止することがある。この場合は root で \$(HOME)/PROJ/VLBI/e-VLBI/VDIF/app/chm.sh を実行すると必要なコマンドのパーミッションを変更し、スティッキイビットを付与することで、一般ユーザが使えるようになる。

(2) 受信側で

```
sudp_recv3 -x Mk5Geo
```

として受信サーバを起動する。

- x Mk5Geo : というオプションにより、Mark5 の VSI データの ch1,ch8 の LSB データを K5/VSSP32 の 14,15ch にマッピングしなおす機能がある。

(注 2) 送信サーバでこのオプションを付与して、送信側でチャンネル Mapping を変更することも可能であるが、Wetzell-Tukuba 伝送の場合には Wetzell に設置している K5/VSI の PC の性能が十分でないためか (?) 受信側でのパケットロスが増加するので、チャンネルの並べ替えは、受信側で行うようにしている。

(3) 送信側サーバに対して TCP(9902)で以下の ASCII コマンドを送信し、データを送信状態にする。

```
in2net=connect:<target host>:9901;  
play_rate=data:256;  
mode=16:1:16;  
in2net=on;
```

<説明>

- 1 行目: 送信先のホストとポートの設定
- 2 行目: 送信レートの設定
- 3 行目: データ取得モード 16Msps 1bit 16ch を指定
- 4 行目: 送信開始

(注 3) TCP/IP で ASCII 文字列を送受するツール(top-talk.pl)もパッケージに含まれている。詳細は後述する。

(4) 受信側サーバに対して、TCP(9903)で以下のコマンドを送信すると記録の開始ができる。

```
data_out=multi:4;  
modegrp=0:16:1:4:0;  
modegrp=1:16:1:4:4;  
modegrp=2:16:1:4:8;  
modegrp=3:16:1:4:12;  
data_out=FILE0:<file group1>;  
data_out=FILE1:<file group2>;
```

```
data_out=FILE2:<file group3>;
data_out=FILE3:<file group4>;
in2net=on;
```

<説明>

- 1 行目：データ出力先が4つあることを設定
- 2 行目：group 0 は 16Msps/1bit/4ch のモードで最初の 0ch 目からの 4ch データを出力する。
- 3 行目：group 1 は 16Msps/1bit/4ch のモードで最初の 4ch 目からの 4ch データを出力する。
- 4 行目：group 2 は 16Msps/1bit/4ch のモードで最初の 8ch 目からの 4ch データを出力する。
- 5 行目：group 3 は 16Msps/1bit/4ch のモードで最初の 12ch 目からの 4ch データを出力する。
- 6 行目：group 0 の出力ファイル名設定
- 7 行目：group 1 の出力ファイル名設定
- 8 行目：group 2 の出力ファイル名設定
- 9 行目：group 3 の出力ファイル名設定
- 10 行目：送信開始

受信停止のコマンドは

```
in2net=off;
```

5. 周辺ツール

[sudp_recv2]

sudp_send2 では VDIF のフレーム/パケットを SUDP の短いヘッダをつけて UDP/IP で送信される。sudp_recv2 はそのストリームを受信するツールである。

(1) 使い方その1：受信テスト

sudp_recv2 を 引数なしで使用すると、受信される VDIF/SUDP のパケットからデータレートなどの受信モードを識別し、計測された受信レートなどを毎秒表示する。

(2) 使い方その2：記録テスト

-o <output file> オプションを付けて起動すると、受信データをファイルに記録する。フォーマットは VDIF、または VSSP32(デフォルト)であり、-t <FMT>オプションで指定する。記録する時間の長さは -l <sec>オプションで指定する。

実行例) sudp_recv2 -t VDIF -o /mnt/raid/test1.dat -l 30

<オプション説明> (-h の引数をつけて起動すると簡単な使いかたが表示される。)

- p <port>, Data receiving port number. default=9901
- o <file>[:<file2>...] Output to files for multiple out.
- t <FMT>: Writing format, VDIF | VSSP. (default:VSSP32).
- n <nch>, Number of channels (1|4) for output. default=4.

-g <group>[:<grp>...], Writing Group number of nch. 0,1,2..
-m <'xxMsps/xbit/xxch'>: Output Data mode
-M <'xxMsps/xbit/xxch'>: Input Data mode
-s <byte> Packet size (default=1500).
-l <sec> length of receiving data (default=0 means endless).
-v: verbose level, 0-4 (default -1: no logging).

If [-o <outfile>] is not specified, data is just receiving to memory and no output.

[tcp-talk.pl]

tcp/ip で ASCII 文字列を送信するツール tcp-talk.pl がパッケージに含まれており、インストール後は \$(HOME)/PROJ/VLBI/e-VLBI/VDIF/app/ にある。引数無しで実行すると使いかたが表示される。このツールを使って一つ一つコマンド列を実行することもできる。送信側の制御はこれで十分であるが、受信側は SCAN 毎に出力ファイル名などを変更する必要がある、このままでは不便である。

スケジュールにしたがって、受信サーバをコントロールするツール autoobs-rmtrcv.pl もパッケージに含まれている。

[autoobs-rmtrcv.pl]

このツール使うためには perl-module と Term-ANSIScreen のモジュールをインストールする必要がある。

\$(HOME)prog/perl/module に perl-module をコピーし

Term-ANSIScreen という perl module を CPAN から取得しインストールする。

autoobs-rmtrcv.pl を引数無しで実行すると、使いかたが表示される。

実行例を以下に示す。

```
autoobs-rmtrcv.pl -skd ~/schedule/k10051.skd -st V -dir /cont1/K10051/wetz-onln  
-n 16 -host sakura.spacegeodesy.go.jp:9903 -mode 16:1:4 -exe
```

<オプション説明>

-skd <schedule file>

-st <Station ID>: # ファイル名の先頭文字に使われる。

-dir <output dir>

-n <nch> # トータルのチャンネル数

-mode <mode> #K5/VSSP32 記録モード 16Msps/1bit/4ch

-exe を付けられない場合、設定だけを行って、記録開始コマンドを発行しない。

[send-start.pl]

sudp_send2 に TCP/IP でコマンドを送り、簡易にストリームの送信を行うためのツール。

<host1>にて sudp_send2 を sudp_send2 -d VSI -l 32Msps/2bit/16ch などとして起動し、host2 のポート番号 9903 に 16Msps/1bit/8ch で送信させるものとする。

実行例 send-start.pl host1:9902 host2:9903 16 1 8

[send-stop.pl]

sudp_send2, sudp_recv3 などサーバとして TCP/IP で制御を受けて動作するものは、in2net=off; により送信・受信を停止します。そのコマンド送信を行います。

実行例 send-stop.pl host1:9902

[rmt-recv-start.pl]

sudp_recv3 に対して記録モードの設定と開始のコマンドを送るツール。

実行例 rmt-recv-start.pl -a sirius-a1:9903 -odir /mnt/raid -mode 16:1:16 Oid O

<オプション説明>

- a <host>[:<port>]: 制御対象のホストとポート番号(default=9903)
- id <ID>: Station ID, default = 'O'.
- mode <Msps>:<bit>:<ch>: Recording mode, default = '16:1:16'.
- odir <dir>: Output directory.
- fn: File naming convention: (default 2)
 - 1 : XDDNNNN.dat (default)
 - 1 : XDDNNNN.#ch.dat
 - 2 : sidDDHMMSSG.dat
 - 2 : sidDDHMMSSG.#ch.dat

6. 参考文献

[1] VLBI Data Interchange Format (VDIF) Specification

<http://www.vlbi.org/vsi/docs/VDIF%20specification%20Release%201.0%20ratified.pdf>