# Linux Bitstreaming Protocol

**Author**: Nenad Corbic

**Date**:  Sep 18, 2002

## Introduction

Linux Bitstreaming protocol streams raw line data to and from the user API via linux socket interface. The bitstreaming can be configured in two operation modes: API or SWITCH. Furthermore, the bitstreaming protocol has HDLC encoding/decoding engine.

## Hardware Support

Bitstreaming drivers supports all WANPIPE cards: S514-1 to 7.

## Driver Installation

❿Download the latest wanpipe release from
ftp.sangoma.com/linux/current_wanpipe.

❿Untar it in /tmp directory

```
tar xvfz wanpipe<version>.tgz
```

❿Make sure that the Linux source in /usr/src/linux directory is the source of the currently running image.

```
uname -r
head /usr/src/linux/Makefile
```

❿Run the Setup script with install and --protocol=BITSTRM options

```
./Setup install --protocol=BITSTRM
```

❿After a successful installation proceed to protocol configuration.
Otherwise, call Sangoma Tech Support:
905 474-1990
teckdesk@sangoma.com

## Configuration

All Bitstreaming driver configuration files should be generated using the /usr/sbin/wancfg configuration tool.

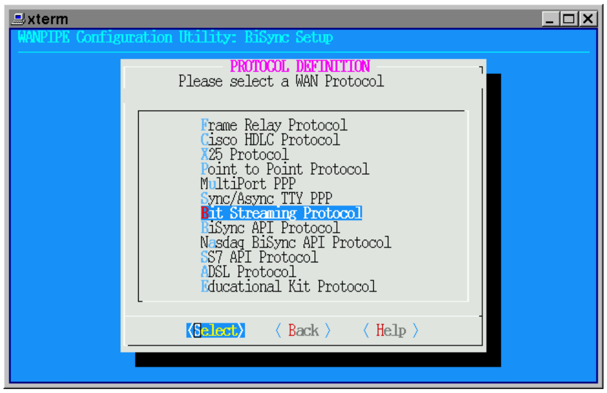
Wancfg utility contains extensive help files. These help files should be used as reference to this manual.

Start /usr/sbin/wancf from any directory:

```
/usr/sbin/wancfg <enter>
```

Protocol:

❿Bit Streaming

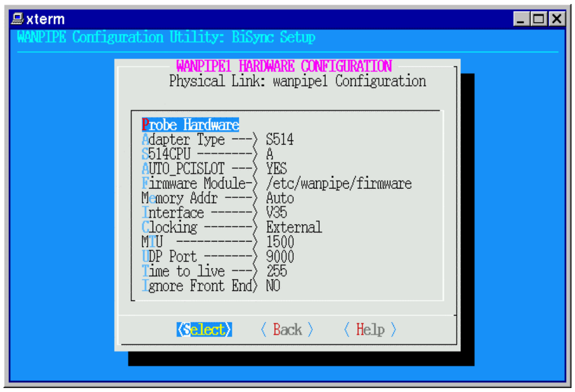


```
xterm
WANPIPE Configuration Utility: BiSync Setup

                PROTOCOL DEFINITION
             Please select a WAN Protocol

              Frame Relay Protocol
              Cisco HDLC Protocol
              X25 Protocol
              Point to Point Protocol
              MultiPort PPP
              Sync/Async TTY PPP
              Bit Streaming Protocol
              BiSync API Protocol
              Nasdaq BiSync API Protocol
              SS7 API Protocol
              ADSL Protocol
              Educational Kit Protocol

          <Select>    < Back >    < Help >
```

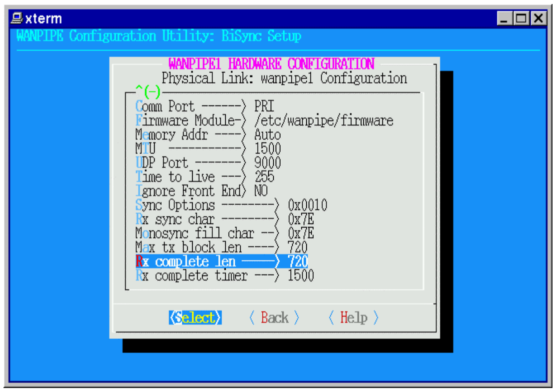
Hardware:

❿Run hardware probe and select a device

❿Specify the maximum Rx and Tx packet sizes.
Note: Both Tx and Rx must be equal when running in SWITCHED mode



```
xterm
WANPIPE Configuration Utility: BiSync Setup

          WANPIPE1 HARDWARE CONFIGURATION
       Physical Link: wanpipe1 Configuration

   Probe Hardware
   Adapter Type ---> S514
   S514CPU --------> A
   AUTO_PCISLOT ---> YES
   Firmware Module-> /etc/wanpipe/firmware
   Memory Addr ----> Auto
   Interface ------> V35
   Clocking -------> External
   MTU  -----------> 1500
   UDP Port -------> 9000
   Time to live ---> 255
   Ignore Front End> NO

      <Select>    < Back >    < Help >
```



```
xterm
WANPIPE Configuration Utility: BiSync Setup

          WANPIPE1 HARDWARE CONFIGURATION
       Physical Link: wanpipe1 Configuration
  ^(-)
   Comm Port ------> PRI
   Firmware Module-> /etc/wanpipe/firmware
   Memory Addr ----> Auto
   MTU  -----------> 1500
   UDP Port -------> 9000
   Time to live ---> 255
   Ignore Front End> NO
   Sync Options --------> 0x0010
   Rx sync char --------> 0x7E
   Monosync fill char --> 0x7E
   Max tx block len ----> 720
   Rx complete len -----> 720
   Rx complete timer ---> 1500

      <Select>    < Back >    < Help >
```

Interface Setup

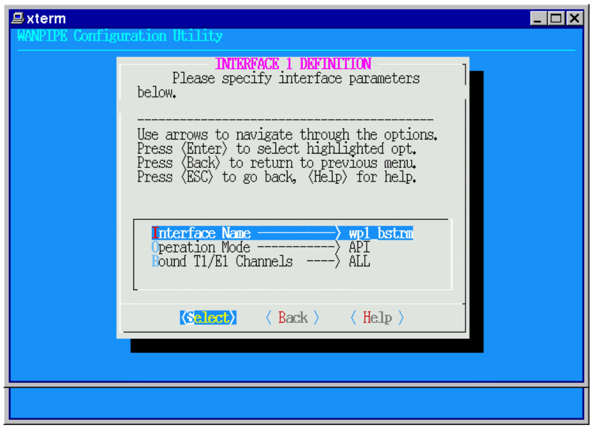
❿Specify the number of interfaces based on number of DS0 channels defined. When running over T1/E1 cards, multiple DS0 channels can be bound to a single interface.  Thus it is possible to have one or all T1/E1 DS0 channels bound to a single network interfaces. Determine how many interfaces this setup will need by evaluating the project needs.

❿Each network interface,  that is bound to a physical T1/E1 DS0 channel(s) can be setup in ether API or SWITCH mode.

API:

All bitstreaming data is passed up to the user custom API application. The API application can tx/rx data to and from bitstreaming driver.

Sample API applications are located in /etc/wanpipe/api/bitstrm.

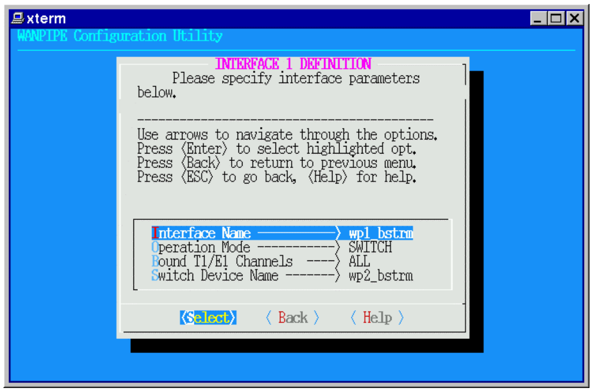


```
     Please specify interface parameters
below.

------------------------------------------
Use arrows to navigate through the options.
Press <Enter> to select highlighted opt.
Press <Back> to return to previous menu.
Press <ESC> to go back, <Help> for help.

  Interface Name -----------> wp1_bstrm
  Operation Mode -----------> API
  Bound T1/E1 Channels  ----> ALL
```

SWITCH:

A SWITCHED network interface, bound to arbitrary number of T1/E1 DS0 channels, can be connected to an interface of another T1/E1 card.  Thus, being able to switch T1/E1 DS0 channels in software between multiple T1/E1 lines.

For example:
DS0 Channels 1-3 from wanpipe1:w1_bstrm can be SWITCHED to DS0 Channels 1-3 from wanpipe2:wp2_bstrm and vice versa.



```
WANPIPE Configuration Utility

              INTERFACE 1 DEFINITION
      Please specify interface parameters
 below.

 ------------------------------------------
 Use arrows to navigate through the options.
 Press <Enter> to select highlighted opt.
 Press <Back> to return to previous menu.
 Press <ESC> to go back, <Help> for help.


   Interface Name -----------> wp1_bstrm
   Operation Mode -----------> SWITCH
   Bound T1/E1 Channels  ----> ALL
   Switch Device Name -------> wp2_bstrm

          <Select>    < Back >    < Help >
```

## Operation

❿Start Wanpipe

```
wanrouter start wanpipe1
wanrouter start wanpipe2
```

❿Refer to /var/log/messages for debug, status and error messages

```
tail -f  /var/log/messages
```

❿If you are running in API mode, proceed to start your API custom application. Otherwise, there is no user interaction when interfaces are in SWITCHEed mode.

❿In SWITHCED mode, make sure that sw devices have correctly bonded to each other.  Only /var/log/messages will display binding errors.

**Demo #1**

SystemA:  API  <------------------>  API :SystemB

The driver will provide a user level API that will allow users to pass transparent synchronous data to and from individual DS0 channels on a T1 or E1 circuit.  It will support 24 DS0 timeslots (T1) or 30 timeslots (E1).  The data will be full duplex and continuous on each DS0 channel.

Setup two Linux workstations A and B:

❿Insert a Sangoma S514-4 T1/E1 card into each workstation.
❿Install latest wanpipe drivers (refer to above instructions)
❿Connect two S514-4 cards with back-to-back T1 cables.

Create wanpipe1 using wancfg, on system A:

❿Protocol:                BitStreaming
❿Hardware:               T1 card (S514-4 or S514-7)
❿Max Rx/Tx MTU:   720 (bytes, default)
❿T1 configuration:     All DS0 channels
                                Clock = MASTER
❿Num of Interfaces: 1
❿If Op Mode:  API
❿If Active Ch: ALL

Create wanpipe1 using wancfg, on system A:

❿Protocol:                BitStreaming
❿Hardware:               T1 card (S514-4 or S514-7)
❿Max Rx/Tx MTU:   720 (bytes, default)
❿T1 configuration:     All DS0 channels
                                Clock = NORMAL
❿Num of Interfaces: 1
❿If Op Mode:  API
❿If Active Ch: ALL

Start both wanpipe cards on systems A and B:

❿wanrouter start wanpipe1
❿tail -f /var/log/messages
❿Look for T1 Connected message in /var/log/messages.

Create a test TX file which client will use to send data to the server

❿Run bstrm_gen_file
    ./bstrm_gen_file

Start the sample bitstreaming API utilities in /etc/wanpipe/api/bitstrm

❿Start the bitstrm API server on system A

```
./bitstrm_file_server wanpipe1 wp1_bstrm
```

❿Start the bitstrm API client on system B

```
./bitstrm_file_client wanpipe1 wp1_bstrm
```

The client will send a large file to the server over ALL 24 T1 DS0 channels. Furthermore, each DS0 channel will carry unique set of data packets to ensure that no DS0 channel crossover occurs. The server will monitor each byte received and will raise alarm if DS0 channel skew is detected.

During file transfer if an under run condition occurs, the server will raise alarm. After the transfer, server application will compare the received file with the copy of the original file and confirm that no corruption occurred.

On each system A and B, during file transfer, one can use the sdladump utility to view packets in firmware memory:

```
sdladump wanpipe1 0x10000  #tx buffer 1
sdladump wanpipe1 0x11000  #tx buffer 2
...
sdladump wanpipe1 0x17000  #tx buffer 8
```

After a successful run, change the interface active DS0 channels in both systemA: wanpipe1.conf and systemB: wanpipe1.conf to 1, or 1-2, instead of ALL. Restart the cards and re-run the test again. At the same time run sdladump utility and one will notice that not all DS0 channels are being used.

**Demo #2**

```
System A:  wp1_bstrm API --------->  SWITCH wp1_bstrm SystemB
                                                    |
                                                    |
System A:  wp2_bstrm API <----------SWITCH wp2_bstrm SystemB
```

Two PCs  SystemA and SystemB, each equipped with two Sangoma S514-4TE1 cards will be connected back-to-back so that data can pass between them.

On System A, a program (client) will transmit specific, identifiable data streams on each 24 DS0s (30 DS0s for E1), via card wanpipe1.     The SystemB (wanpipe1) will switch data received, to the second card in the SystemB (wanpipe2).   The SystemB (wanpipe2) card will send the same data back to SystemA: wanpipe2 card.   A separate demonstration program (server), will receive data and save it into a new file.  Once the SystemA client is finished transmitting, the SystemA server will stop and compare the transmitted and received file.  If the files are not identical, alarm will be generated.

Connect SystemA wanpipe1  to SystemB wanpipe1 using back to back T1 crossover cable.

Connect SystemA wanpipe2 to SystemA wanpipe2 using back to back T1 crossover cable.

Create wanpipe1 using wancfg, on system A:

| | |
|---|---|
| ❿Protocol: | BitStreaming |
| ❿Hardware: | T1 card (S514-4 or S514-7) |
| ❿Max Rx/Tx MTU: | 720 (bytes, default) |
| ❿T1 configuration: | All DS0 channels<br>Clock: MASTER |
| ❿Num of Interfaces: | 1s |
| ❿If Op Mode: | API |
| ❿If Active Ch: | ALL |

Create wanpipe2 using wancfg, on system A:

Same as wanpipe1

Create wanpipe1 using wancfg, on system B:

- Protocol: BitStreaming
- Hardware: T1 card (S514-4 or S514-7)
- Max Rx/Tx MTU: 720 (bytes, default)
- T1 configuration: All DS0 channels  
  Clock: NORMAL
- Num of Interfaces: 1
- If Op Mode: SWITCH
- If Active Ch: ALL
- sw device name: wp2_bstrm

Create wanpipe2 using wancfg, on system B:

- Protocol: BitStreaming
- Hardware: T1 card (S514-4 or S514-7)
- Max Rx/Tx MTU: 720 (bytes, default)
- T1 configuration: All DS0 channels  
  Clock: NORMAL
- Num of Interfaces: 1
- If Op Mode: SWITCH
- If Active Ch: ALL
- sw device name: wp1_bstrm

Start wanpipe1 and wanpipe2 on System A:

- wanrouter start wanpipe1
- wanrouter start wanpipe2

Start wanpipe1 and wanpipe2 on System B:

- wanrouter start wanpipe1
- wanrouter start wanpipe2

On both systems check the /var/log/messages and make sure that both wanpipe1 and wanpipe2 are connected. Look for T1 connected and Link connected messages.

On SystemB make sure that wanpipe1 wp1_bstrm and wanpipe2 wp2_bstrm have bounded together to establish the T1 SWITCH.

Open two Xwindows terminals on System A and change directory to

/etc/wanpipe/api/bitstrm

Create the tx file used by client:

```
❿bstrm_gen_file <enter>
```

In first window start bitstrm server

```
❿bstrm_file_server wanpipe2 wp2_bstrm
```

In second window start bitstrm client

```
❿bstrm_file_client wanpipe1 wp1_bstrm
```

The server will receive data from client over the SWITCHED DS0 channels on SystemB.   The server will return an error message if an under run has occurred, or if the file gets corrupted.