WANPIPE

Multi-protocol WANPIPE Driver for Linux®

CONFIGURATION MANUAL

SANGOMA TECHNOLOGIES

By Nenad Corbic

LIMITED USE WARRANTY

WANPIPE drivers are distributed with the purchase of Sangoma S-series cards. The drivers, where distributed in source form, are free software; you can redistribute and/or modify them under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or any later version.

Where code is provided in object form only, this code remains the property of Sangoma Technologies Inc. and may be used only in conjunction with Sangoma products. Object code may not be unassembled or reverse engineered for any reason.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Sangoma Technologies

Markham, Ontario, Canada

Contents

Standards Used in This Manual

The table below shows text conventions used in this manual.

Text	Description
[ ]	Text within square brackets represents keyboard keys. For example [Enter] or [p]. Combinations of keys held down together are indicated using the plus sign as in the example: [Alt+p].
< >	Text within pointy brackets represents any command or argument after a command. For example, <arg> represents an argument.
`Courier Font`	Text in this font indicates a directory name, a file name or a command string.
Bold	Bold text is used to draw attention to a new concept, or to highlight choices in description of a configuration.

The table below displays icons used to draw attention to items of note.

Icon	Description
ⓘ	Information, general interest note.
✪	Important, recommended procedure or configuration note.
⚠	Critical procedure or configuration note.

1. Configuration Process

In this process, generate a detailed configuration file that describes the hardware, protocol and IP options as well as location of the adapter firmware. Create a new configuration file for each WANPIPE device.

ⓘ *Note: A WANPIPE device does not describe a physical card, but a logical implementation of the number of physical lines connected to a Sangoma adapter. For example, a S5141 card contains a single CPU with two physical ports: a High-Speed port (up to 4Mbps) and a Low-Speed port (up to 512Kbps). Each port can support an independent physical connection.*

Start wancfg

To simplify the WANPIPE configuration process, use a GUI configuration utility called `wancfg`. It is located in the `/usr/sbin` directory. Start the utility with the following command:

```
/usr/sbin/wancfg
```

ⓘ *Note:* `wancfg` *has extensive help files for each WANPIPE option.*

ⓘ *Note:* `wancfg` *requires BASH2 support. Sample WANPIPE configuration files are located in* `/etc/wanpipe/samples` *directory. Use these files if the system does not support BASH2.*

Once the `wancfg` process starts, follow these steps:

1. From the Main Menu, select **Create a new Configuration File**. Position the curser on **<Select>** and press **[Enter]**. The **Select a New Wan Device Name** screen appears.

2. Select a WANPIPE device.

   If configuring the first device, for example, select **wanpipe1**

   With the curser on **<Select>**, press **[Enter]**. The **Main WANPIPE Configuration: New Configuration** screen appears.

3. In this screen, the protocol definition indicates **Undefined**. With the curser on **<Select>**, press **[Enter]**. The **WAN Protocol** list appears. Choose a protocol from the list.

   In this example, choose **Frame Relay**.

   With the curser on **<Back>**, press **[Enter]** to return to the **Main WANPIPE Configuration: New Configuration** screen.

4. The screen title changes to show the selected protocol. The protocol definition also shows the selected protocol.

   Two new setup categories indicate **Undefined**. Select **Hardware Setup**. With the curser on **<Select>**, press **[Enter]**. The **Physical Link** screen appears.

5. Select **Probe Hardware**. With the curser on **<Select>**, press **[Enter]**. The **Select a WANPIPE Device** screen appears.

6. Select a device from the list on this screen. Note the right-hand portion of each line item shows the **Port** option. Usually `PORT=PRI` is used. Make a note of this value.

⚠ *Important: After the hardware probe section, you will notice that the next four fields have been filled out. Thus, adapter type and PCI info should be skipped. All other options should be left as DEFAULT unless you have special information.*

7. Select **Exit** to continue.

8. If CSU/DSU is detected, the **CSU/DSU Setup** screen appears. If this screen does not appear, continue with next numbered step.

   The new S514-4 and S514-7 T1/E1 cards that contain an onboard T1/E1 CSU/DSUs are configured in /usr/sbin/wancfg.

   Check with the line provider for the speed of the fractional T1/E1 line (for example, 64K, 128k ... 1.5M). The fractional T1 consists of 24 channels and E1 consists of 32 channels each 64K. Using the CSU/DSU, configure the line speed by enabling or disabling channels 1 to 24/32.

For example:

64K line	: Enable channel 1, disable 2 to 24
128K line	: Enable channel 1 and 2, disable 3 to 24
Full T1 (1.5M)	: Enable 1 to 24
Full E1 (2.048M)	: Enable 1 to 32

9. If Network Interface protocols are detected, the **Network Interface Setup** line appears as **Undefined**. If this line does not appear, continue with next numbered step.

   Options that appear on the network interface set-up screen depend on the type of connection detected.

   Frame Relay supports many interfaces, each bound to a DLCI. There are questions related to the number of DLCIs and configuration information for each DLCI.

   In the case of Point-to-Point protocols like **PPP** and **CHDLC**, there
   is only one interface per physical port.

   Fill in ISP-specific data from information received from your Service Provider as described in Appendix A of this guide. Other settings in this section remain as default values.

10. Once all network interfaces are set, keep hitting the **Back** button until you reach the Main Menu.

    At the Main Menu, select **Exit** and save the config file.

    The new config file will be located in `/etc/wanpipe` directory.

Testing the Configuration

Start the device to test configuration. For example:

```
wanrouter start <devicename>
```

ⓘ *Note: Default value for <devicename> is wanpipe1.*

ⓘ *Note: The card does not have to be connected to start the wanrouter.*

If the device fails to start, do the following:

1. Run `wanrouter conflog`.

   This will show a report with syntax errors found in the configuration file.

2. Run `wanrouter messages.`

   This will show a report with device driver configuration operational errors.

3. Contact your Sangoma representative.

System Boot Options

To start this newly configured device on boot up, edit `wanrouter.rc` by appending this device by name in the `WAN_DEVICES` section. For example:

```
WAN_DEVICES = “wanpipe1, wanpipe2 ? “
```

By default, wanpipe1 is already in place. Add each device after number 1 to reflect your unique system configuration.

All device names must be inside quotation marks and separated by a single space. Please refer to the Appendix A for information on how to configure each protocol.

2. Configuring the Onboard FT1 CSU/DSU

With Sangoma S508FT1 and S514-3 FT1 cards, configure the on-board CSU/DSU separately using the `/usr/sbin /cfgft1` utility.

✪ *Note: This section relates to only the S514-3 and S508FT1 cards. The new S514-4 and S514-7 T1/E1 cards use* `/usr/sbin/wancfg` *for this function.*

Before proceeding with the CSU/DSU configuration:

1. The CSU/DSU configuration information must be obtained from the T1 provider.

Check with your T1 provider for the speed of the fractional T1 line (64K, 128k ... 1.5M). The fractional T1 consists of 24 channels each 64K. Using the CSU/DSU, configure the line speed by enabling or disabling channels 1 to 24.

For example:

64K line	: Enable channel 1, disable 2 to 24
128K line	: Enable channel 1 and 2, disable 3 to 24
Full T1 (1.5M)	: Enable 1 to 24

Start and stop the WANPIPE device using the wanrouter command. This will test the wanpipe#.conf file and make sure that the card is present.

```
wanrouter start

wanrouter stop
```

2. CFGFT1 Requirements

`/usr/sbin/cfgft1` is a GUI application written in, BASH version 2, shell script. You must have BASH version 2.0 or greater to run this script.

The `cfgft1` utility uses the `/bin/bash2` executable.

ⓘ *Note: The* `cfgft1` *utility is NOT supported for 2.0.X Kernels. In case of 2.0.X Kernels use the* `/usr/sbin/cpipemon debugging/configuration` *utility. For more information, run the program without any arguments, and read the help information.*

3. Run `CFGFT1`

   The `CFGFT1` utility contains all help files necessary to configure the CSU/DSU.

   ```
   /usr/sbin/cfgft1 wanpipe1
   ```

4. The `CFGFT1` utility has three modes of operation.

   Standard Configuration Mode

   Unless the system has special consideration, standard configuration is the simplest method and will be sufficient for most configuration cases.

   Advanced Configuration Mode

   This is a text based configuration mode, where commands are sent to the CSU/DSU directly. It should be used if standard configuration does not meet the requirements.

   Auto Detect Configuration Mode

   This option works only for B8ZS encoding and ESF framing modes. It will try to detect the speed of the line and automatically configure the CSU/DSU.

   For further information regarding the advanced CSU/DSU configuration option, please refer to Appendix B.

APPENDIX A – Configurations for Various Protocols

This Appendix contains configuration instructions for the following protocols:

- Frame Relay
- Multi-Port Frame Relay
- Cisco HDLC
- PPP
- Multi-Port Synchronous PPP
- ADSL
- ATM
- X.25
- Synchronous/Asynchronous/Multilink (TTY) PPP

Frame Relay Configuration

Frame Relay is a simplified form of Packet Switching similar in principle to X.25 in which synchronous frames of data are routed to different destinations depending on header information.

Frame Relay is cost effective, partly because the network buffering requirements are carefully optimized. Compared to X.25, with its store and forward mechanism and full error correction, network buffering is minimal. Frame Relay is also much faster than X.25. The frames are switched to their destination with only a few byte times delay, as opposed to several hundred milliseconds delay on X.25.

⚠ *Important: As with all other protocols, this version of PPP protocol is supported by WANPIPE in firmware. The limitation of this PPP protocol is that it cannot drive a Secondary port on the Sangoma adapter (WANPIPE S514-1).*

To overcome this limitation, see Multi-port PPP Configuration, later in this appendix.

WANPIPE Frame Relay Options

WANPIPE Frame Relay uses modes of operation, as displayed below:

Table 1 - Modes of Operation for Frame Relay

Mode	Description
WANPIPE MODE	The Linux Kernel uses Frame Relay logical channels to route packets to remote networks, using TCP/IP protocol. Each logical channel is represented by a network interface, where each interface contains unique IP information. The Kernel uses the IP information to route incoming packets to remote networks.
API MODE	Frame Relay API mode is used to send non-IP traffic over a Frame Relay link. The API interface allows the user to build a custom application on top of the Frame Relay link in order to transmit custom data packets (i.e. Non IP). Voice-over IP, Data capture and packet analysis are examples of custom applications.
BRIDGING MODE	The 'Kernel bridge' is used to bridge multiple Frame Relay logical channels together into a single LAN. This option is desirable if IP addresses are scarce, or if building a single LAN architecture. Multiple remote LANs can be bridged together into a single LAN using the Frame Relay (WAN) links. Please refer to `WANPIPEEthernetBridge.txt` for further information.

Information Needed From Your ISP

✪ *Note: For a single DLCD, configure with Autodetect option in wancfg.*

- List of DLCI (channels) used

  DLCI is a logical Frame Relay link/channel (16 – 4096).

- IP address for each DLCI channel (WANPIPE MODE)

  For example: Local: 201.1.1.1, Remote: 201.1.1.2

- Clocking Mode

  In most cases clocking will be External (i.e. the ISP will supply the clock)

- Frame Relay Signaling

  Frame Relay has number of signaling options: LMI, ANSI, Q933 (default is ANSI)

- Frame Relay Station

  Frame Relay has two modes of operation:

  - CPE: customer premises equipment. As an end user, a Frame Relay connection should always be set to this mode.

  - NODE: switch emulation: This option should only be used in back-to-back test situation, with two Sangoma card. Sangoma can act as a switch, however, in most cases that is performed by the ISP.

- CSU/DSU Configuration:

Sangoma S574-4/7/8, T1/E1 CARDS and S514-3/S508FT1 cards are supplied with an onboard CSU/DSU that needs to be configured, based on the type of line to which it is connected.

Multi-Port Frame Relay Configuration

As mentioned in the section above, the resident Sangoma WANPIPE Frame Relay configuration is limited in that it cannot run on a Secondary port on the Sangoma adapter (S514/S508). Support for Multi-Port Frame Relay was developed to address this limitation. The Multi-Port Frame Relay is a standard implementation of Frame Relay protocol implemented in the Linux Kernel. It is not resident in WANPIPE firmware.

⚠ *Important: Since the Multi-Port Frame Relay is implemented in the Kernel, the second port is freed up. As a result, multiple independent Frame Relay connections can be established on both Sangoma adapter ports simultaneously.*

WANPIPE Multi-Port Frame Relay Options

WANPIPE Multi-Port Frame Relay uses modes of operation, as displayed below:

Table 2 - Modes of Operation for Multi-Port Frame Relay

Mode	Description
WANPIPE MODE	The Linux Kernel uses Frame Relay logical channels to route packets to remote networks, using TCP/IP protocol. Each logical channel is represented by a network interface, where each interface contains unique IP information. Kernel uses the IP information to route incoming packets to remote networks.
API MODE	Not supported
BRIDGING MODE	Not supported

Information Needed From Your ISP

- List of DLCI (channels) used

  DLCI is a logical Frame Relay link/channel (16 - 4096).

- IP address for each DLCI channel (WANPIPE MODE)

  For example: Local: 201.1.1.1, Remote: 201.1.1.2

- Clocking Mode

  In most cases clocking will be External (i.e. the ISP will supply the clock)

- Frame Relay Signaling

  Frame Relay has number of signaling options: LMI, ANSI, Q933 (default is ANSI)

- Frame Relay Station

  Frame Relay has two modes of operation:

  - **CPE:** customer premises equipment. As an end user, a Frame Relay connection should always be set to this mode.

  - **NODE: Switch Emulation:** This option should only be used in a back-to-back test situation, with two Sangoma cards. A Sangoma card can act as a switch. In most cases, however, switching is performed by the ISP.

- CSU/DSU Configuration:

  Sangoma S514-4/7/8 T1/E1 and S514-3/S508FT1 cards are supplied with an onboard CSU/DSU that need to be configured, based on physical line parameters supplied by Telco.

Cisco HDLC Configuration

Cisco HDLC is a point-to-point protocol implemented on top of HDLC layer 2. As the name implies CHDLC is a protocol mostly used to connect to the Cisco external routers.

⚠ *Important: This version of CHDLC protocol is supported by WANPIPE in firmware. CHDLC is the ONLY firmware protocol that can support both WANPIPE adapter ports simultaneously. Because it can also provide raw HDLC streaming, it is often used as a base for all other HDLC based protocols such as (PPP and FR).*

WANPIPE CHDLC Options

WANPIPE CHDLC uses modes of operation as displayed below:

Table 3 - Modes of Operation for CHDLC

Mode	Description
WANPIPE MODE	The Linux Kernel uses the CHDLC point-to-point link to route packets to a remote network, using TCP/IP protocol. A CHDLC point-to-point connection is represented by a single network interface that contains IP information obtained from the ISP. The Kernel uses the IP information to route incoming packets to remote a network.
API MODE	CHDLC mode used to send non-IP traffic over a CHDLC point-to-point link. The API interface allows the user to build a custom application on top of the CHDLC link in order to transmit custom data packets (i.e. Non IP). An example of a custom application would be a Satellite Receive Only data collector or Data capture and packet analysis tool.
BRIDGING MODE	The 'Kernel bridge' is used to bridge multiple Frame Relay logical channels together into a single LAN. This option is desirable if IP addresses are scarce, or in building a single LAN architecture. Thus, multiple remote LANs can be bridged together into a single LAN using the Frame Relay (WAN) links. Please refer to WANPIPEEthernetBridge. (pdf/txt) for further information.

Information Needed From Your ISP

- IP address for the CHDLC point-to-point connection (WANPIPE MODE)

  For example: Local: 201.1.1.1, Remote: 201.1.1.2

- Clocking Mode

  In most cases clocking will be External (i.e. the ISP will supply the clock)

- CSU/DSU Configuration

  Sangoma S574-4/7/8, T1/E1 CARDS and S514-3/S508FT1 cards are supplied with an onboard CSU/DSU that needs to be configured, based on the type of line to which it is connected.

PPP Configuration

Point-To-Point Protocol (PPP) is a protocol implemented on top of the second HDLC layer. PPP is a standard protocol used when connecting over a point-to-point link.

⚠ *Important: As with all other protocols, this version of PPP protocol is supported by WANPIPE in firmware. The limitation of this PPP protocol is that it cannot drive a Secondary port on the Sangoma adapter (WANPIPE S514-1).*

To overcome this, use Multi-port PPP, described later in this appendix.

WANPIPE PPP Options

WANPIPE Point-to-Point Protocol (PPP) operates only in one mode.

Table 4 - Mode of Operation for PPP

Mode	Description
WANPIPE MODE	Packets are routed to a remote network, using TCP/IP protocol. A PPP connection is a single network interface with IP information from the ISP. The Kernel uses the IP information to route incoming packets to remote a network.
API MODE	Not supported

Information Needed From Your ISP

- IP address for the PPP point-to-point connection (WANPIPE MODE)

  For example: Local: 201.1.1.1, Remote: 201.1.1.2

- Clocking Mode

  In most cases clocking will be External (i.e. the ISP will supply the clock)

- CSU/DSU Configuration

  Configure the onboard CSU/DSU based on the type of line to which the Sangoma card is connected.

Multi-Port Synchronous PPP Configuration

As mentioned in the section above, the resident Sangoma WANPIPE PPP configuration is limited in that it cannot run on a Secondary port. Support for Multi-Port PPP was developed to address this limitation. The Multi-Port Synchronous PPP is a standard implementation of PPP protocol implemented in the Linux Kernel. It is not resident in WANPIPE firmware.

⚠ *Important: Since the Multi-Port PPP is implemented in the Kernel, the second port is freed up. As a result, multiple independent PPP connections can be established on both Sangoma adapter ports simultaneously.*

WANPIPE Multi-Port Sync PPP Options

WANPIPE Multi-Port PPP operates only in one mode.

Table 5 - Mode of Operation for Multi-Port Sync PPP

Mode	**Description**
WANPIPE MODE	The Linux Kernel uses the PPP point-to-point link to route packets to a remote network, using TCP/IP protocol. A PPP point-to-point connection is represented by a single network interface that contains IP information obtained from the ISP. The Kernel uses the IP information to route incoming packets to remote a network.
API MODE	Not supported

Information Needed From Your ISP

- IP address for the PPP point-to-point connection (WANPIPE MODE)

  For example: Local: 201.1.1.1, Remote: 201.1.1.2

- Clocking Mode

  In most cases clocking will be External (i.e. the ISP will supply the clock)

- CSU/DSU Configuration

  Sangoma S574-4/7/8, T1/E1 CARDS and S514-3/S508FT1 cards are supplied with an onboard CSU/DSU that needs to be configured, based on the type of line to which it is connected.

ADSL Configuration

The S518 card and WANPIPE together provide universal ADSL support for all versions and distributions of Linux. The S518 provides a robust, stable ADSL link at the highest possible line speeds to any currently installed DSLAM.

All ADSL networks use ATM ALL5 protocol as its data link layer. ATM in turn can transport multiple protocols such as: Classical IP, Bridged Ethernet (PPPoE), and PPPoA. WANPIPE supports full ATM AAL5 protocol as well as support for all upper layer protocols.

WANPIPE ADSL Options

WANPIPE ADSL operates only in two modes.

Table 6 - Modes of Operation for ADSL

Mode	Description
WANPIPE MODE	The Linux Kernel uses the ADSL link to route packets to a remote network, using TCP/IP protocol. This mode is available to both Classical IP over ATM and Ethernet over ATM configurations. In this mode, IP info must be supplied by the user. (IP Info is usually obtained by Telco) An ADSL connection is represented by a single network interface that contains IP information obtained from the ISP. The Kernel uses the IP information to route incoming packets to remote a network.
PPPoE MODE	This mode is only available when ATM is configured to carry Bridged Ethernet protocol. The PPPoE layer in turn operates over an Ethernet protocol. In this mode, no IP information is needed: PPPoE will negotiate IP information with the Telco.

Information Needed From Your ISP

- ATM Configuration: VPI and VCI numbers.

  For example: Vpi=0 Vci=35

- ATM Protocol Encapsulation: i.e. protocol running over ATM layer:

  Bridged Ethernet over ATM (LLC): Often used by PPPoE
  Bridged Ethernet over ATM (VC)
  Classical IP over ATM (LLC)
  Routed IP over ATM (VC)
  PPP over ATM (LLC)
  PPP over ATM (VC)

- Network Layer Protocol

  IP, Ethernet, PPPoE, PPPoA

- IP address (WANPIPE MODE)

  For example: Local: 201.1.1.1, Remote: 201.1.1.2

- Higher protocols such as PPPoE and PPPoA are not implemented in WANPIPE drivers. One must use third party utilities over WANPIPE interfaces to implement the desired protocol.

- PPPoE Configuration:

  Download: ftp.sangoma.com/linux/utilities/rp-pppoe-3.4.tar.gz

  Untar it in a temporary directory and read the README file for installation and configuration instructions.

  Refer to APPENDIX B in this guide for more information on PPPoE.

- PPPoA Configuration:

  One must configure the Kernel "pppd" daemon over, wanpipe /dev/ttyWP{X} device, which is created on wanrouter startup.

  Refer to APPENDIX C in this guide for more information on PPPoA.

ATM Configuration

ATM protocol support for S514 (T1/E1/V35) cards. Using the ATM protocol and S514/T1/E1/V35 adapters, customers can connect to an ATM backbone network via T1 or E1 line.

Wanpipe ATM protocol supports ALL5 framing. ATM in turn can transport multiple protocols such as: Classical IP, Bridged Ethernet (PPPoE). WANPIPE supports full ATM AAL5 protocol as well as support for all upper layer protocols.

ⓘ *Note: PPPoA is not yet supported by Wanpipe ATM.*

WANPIPE ATM Options

WANPIPE ADSL operates only in two modes.

Table 7 - Modes of Operation for ATM

Mode	Description
WANPIPE MODE	The Linux Kernel uses the ADSL link to route packets to a remote network, using TCP/IP protocol. This mode is available to both Classical IP over ATM and Ethernet over ATM configurations. In this mode, IP info must be supplied by the user. (IP Info is usually obtained by Telco) An ADSL connection is represented by a single network interface that contains IP information obtained from the ISP. The Kernel uses the IP information to route incoming packets to remote a network.
PPPoE MODE	This mode is only available when ATM is configured to carry Bridged Ethernet protocol. The PPPoE layer in turn operates over an Ethernet protocol. In this mode, no IP information is needed: PPPoE will negotiate IP information with the Telco.

Information Needed From Your ISP

- ATM Configuration: VPI and VCI numbers.

  For example: Vpi=0 Vci=35

- ATM Protocol Encapsulation: i.e. protocol running over ATM layer:

  Bridged Ethernet over ATM (LLC): Often used by PPPoE

  Bridged Ethernet over ATM (VC)

  Classical IP over ATM (LLC)

  Routed IP over ATM (VC)

  PPP over ATM (LLC) (Not yet supported)

  PPP over ATM (VC) (Not yet supported)

- Network Layer Protocol

  IP, Ethernet, PPPoE, PPPoA (not yet supported)

- IP address (WANPIPE MODE)

  For example: Local: 201.1.1.1, Remote: 201.1.1.2

Higher protocols such as PPPoE and PPPoA are not implemented in WANPIPE drivers. Use third party utilities over WANPIPE interfaces to implement the desired protocol.

- PPPoE Configuration:

  Download: ftp.sangoma.com/linux/utilities/rp-pppoe-3.4.tar.gz

  Untar it in a temporary directory and read the README file for installation and configuration instructions.

  Refer to APPENDIX B in this guide for more information on PPPoE.

- PPPoA Configuration: (Not yet supported)

  Configure the Kernel "pppd" daemon over, wanpipe /dev/ttyWP{X} device, which is created on wanrouter startup.

  Refer to APPENDIX C in this guide for more information on PPPoA.

X.25 Configuration

X.25 Packet Switched networks allow remote devices to communicate with each other across high-speed digital links without the expense of individual leased lines. Packet Switching is a technique whereby the network routes individual packets of HDLC data between different destinations based on addressing within each packet.

The protocol known as X.25 encompasses the first three layers of the **OSI 7-layered architecture** as defined by the International Organization for Standardization (ISO) as follows:

- Layer 1: The Physical Layer is concerned with electrical or signaling. It includes several standards such as V.35, RS232 and X.21.
- Layer 2: The Data Link Layer, which is an implementation of the ISO HDLC standard called Link Access Procedure Balanced (LAPB) and provides an error free link between two connected devices.
- Layer 3: The Network Layer, which provides communications between devices connected to a common network. In the case of X.25, this layer is referred to as the X.25 Packet Layer Protocol (PLP) and is primarily concerned with network routing functions and the multiplexing of simultaneous logical connections over a single physical connection.

WANPIPE X.25 Options

WANPIPE X.25 uses modes of operation as displayed below:

Table 8 - Modes of Operation for X.25

Mode	Description
WANPIPE MODE	The Linux Kernel uses X.25 logical channels to route packets to remote networks, using TCP/IP protocol. Each logical channel is represented by a network interface, where each interface contains unique IP information. Kernel uses the IP information to route incoming packets to remote networks.
API MODE	X.25 API mode is used to send non-IP traffic over an X.25 link. Using the WAN API suite, a vast range of applications can be developed, such as: Credit card verification, Voice-over IP, Protocol/Line Data Scope, Satellite Communication, protocol conversion and Legacy interconnect.

Information Needed From Your ISP

- List of Logical Channel Numbers (LCNs) used:

  X.25 LCNs can be configured as:

  - **Switched Virtual Circuits (SVC).** SVC is analogous to a telephone line. A call must be established before communication takes place.
  - **Permanent Virtual Circuits (PVC).** PVC line is always connected, thus not calls setup is required.

The ISP must provide LOWEST (SVC/PVC) and HIGHEST (SVC/PVC) numbers.

- IP address for each LCN channel (WANPIPE MODE only)

  For example: Local: 201.1.1.1, Remote: 201.1.1.2

- Clocking Mode

  In most cases clocking will be External (i.e. the ISP will supply the clock)

- X.25 Station

  X.25 has two modes of operation: DCE or DTE.

Synchronous/Asynchronous/Multilink (TTY) PPP Configuration

The standard WANPIPE PPP (supported in firmware) has the following limitations:

- Cannot run on a Secondary port
- Does not support Multilink operation

The **Sync/Async/Multilink PPP** was developed to address the limitations of standard WANPIPE PPP. Sync/Async/Multilink PPP is a standard implementation of the PPP protocol implemented in the Linux Kernel. It is not resident in WANPIPE firmware. Using the Sangoma adapter as a TTY serial card along with the Linux PPP Layer, a standard PPP connection can be established over a T1 Link or a Modem line.

The Sync/Async/Multilink PPP has two modes of operation:

Table 9 - Modes of Operation for Sync/Async/Multilink PPP

Mode	Description
SYNCHRONOUS WITH MULTILINK OPTION	To establish a PPP connection over sync T1 lines.
ASYNCHRONOUS	To establish a PPP connection via MODEM over a telephone line.

ⓘ *Note: Both modes are using for ROUTING purposes. That means there is no API support.*

Because the Sync/Async PPP is implemented in the Kernel, the second port is freed up. As a result, multiple independent PPP connections can be established on both Sangoma adapter ports simultaneously.

SYNC Mode Configuration

Using the PPPD daemon, Kernel Sync-PPP layer and the WANPIPE sync TTY driver, a PPP protocol connection can be established via the Sangoma adapter, over a T1 leased line.

The 2.4.0 Kernel PPP layer supports MULTILINK protocol. It can be used to bundle any number of Sangoma adapters (T1 lines) into one, under a single IP address, efficiently obtaining multiple T1 throughputs.

ⓘ *Note: The remote side must also implement MULTILINK PPP protocol.*

ASYNC Mode Configuration

Using the PPPD daemon, Kernel Async PPP layer and the WANPIPE async TTY driver, a PPP protocol connection can be established via the Sangoma adapter and a modem, over a telephone line.

The WANPIPE Async TTY driver simulates a serial TTY driver that is normally used to interface the MODEM to the Linux Kernel.

ⓘ *Note: This option only works on a SECONDARY Port of the S514-PCI/S508-ISA card.*

Device /dev/ttyWP(0,1,2..)

To interface a PPPD daemon to the WANPIPE TTY driver a /dev/ttyWPX X={0,1,3...) device must be created. For example:

```
mknod -m 666 /dev/ttyWP0 c 240 0
```

ⓘ *Note: 240 is the Major Number and 0 is the Minor Number.*

⚠ *Important: This option should only be used if the MULTILINK option desired, to bundle T1 connections together or to simulate a serial async device driver. Otherwise, it is recommended that standard WANPIPE PPP be used.*

Information Needed From Your Telco and ISP

- IP address for the PPP point-to-point connection
    - If STATIC IP addressing is used. For example:

```
Local:  201.1.1.1    Remote: 201.1.1.2
```

Sync/Async TTY PPP Configuration and Operation

The WANPIPE TTY driver has very few options since main configuration options will be defined during the PPPD daemon configuration.

WANPIPE TTY OPTIONS

Use the `wancfg` utility to create the `wanpipe1.conf` configuration files.

- **TTY_MINOR –** Represents a TTY port.
  - Options: 0,1,2,3 ? Default: 0

  It binds a WANPIPE device driver to the `/dev/ttyWPX` device, where X= (0,1,3,..). For example, MINOR number 0 binds a WANPIPE TTY driver to `/dev/ttyWP0`. When **pppd daemon** opens, the `/dev/ttyWP0` it will reach the device driver whose Minor number is zero.

✪ *Note: TTY_MINOR must be a different value for each WANPIPE device.*

- **TTY_MODE – WANPIPE** TTY driver operation mode.
  - Options: Sync or Async Default: Sync

  The driver operation mode must be specified here, since the driver cannot obtain the operation mode from the pppd configuration calls. If the driver operation mode is synchronous then the pppd must be invoked with the sync option.

  All subsequent drivers must be configured with the same TTY MODE and different MINOR numbers.

wancfg Utility

The `wancfg` utility will configure the **pppd daemon** according to the TTY_MODE selected. It will also prompt the user for MULTILINK support. Three files will be created for each WANPIPE device:

`/etc/ppp/options.ttyWPX`	X is TTY_MINOR number.
`/etc/ppp/peers/isp_wanpipeX`	X is a device number (1-16)
`/dev/ttyWPX`	X is TTY_MINOR number.

- WANPIPE TTY drivers must be started before the pppd attempts to open a `/dev/ttyWPX` device. For example:

```
wanrouter start wanpipe1
```

- Once the WANPIPE device is started, the PPP connection can be established by calling the pppd call script (created by `wancfg`). For example:

```
pppd call isp_wanpipe1
```

Manual PPPD Daemon Configuration

Depending on the TTY MODE used, the pppd configuration file must be configured for synchronous or asynchronous operation.

The pppd daemon uses an options configuration file found in /etc/ppp directory. An options configuration file exists for each /dev/ttyX device. For a /dev/ttyWP0 device, an async options file or a sync options file must be created in /etc/ppp directory, as described below:

Asynchronous PPPD Daemon Configuration

Start async configuration with the following command:

```
/etc/ppp/options.ttyWP0
```

Table 10 – Async Options.ttyWP0 File

Option	Description
asyncmap 0	
modem	Use the modem signals.
silent	Wait until ppp request is received before starting ppp protocol (optional).
Debug	Enable debugging (optional).
Crtscts	Enable crtscts hardware flow control.
noipdefault	Use dynamic IP addressing. Obtain IP addresses from the remote side. *Note: Disable this option if using STATIC IP addressing. (async only)*
--detach	Do not detach from the terminal window.
defaultroute	IP address of this interface should be set as default in the routing table.

Synchronous PPPD Daemon Configuration

Start sync configuration with the following command:

```
/etc/ppp/options.ttyWP0
```

Table 11 – Sync Options.ttyWP0 File

Option	Description
asyncmap 0	
silent	#Wait until ppp request is received before starting ppp protocol (optional).
Debug	#Enable debugging (optional).
--detach	#Do not detach from the terminal window.
defaultroute	#IP address of this interface should be set as default in the routing table.

Call pppd Scripting

A **call pppd** script can also be defined to simplify the pppd argument line. The call script must be defined in /etc/ppp/peers directory. The example call script will be called isp_async or isp_sync as described below:

Async PPP Call Script

```
/etc/ppp/peers/isp_wanpipe1:
```

ttyWP0	Wanpipe TTY driver
38400	Baud Rate
connect '/etc/ppp/redialer'	A modem dial up script

✪ *Note: In async mode, WANPIPE TTY drivers are always set to internal clockin, and the baud rate needs to be set here. The driver obtains the baud rate through pppd configuration calls, not wanpipe1.conf configuration file as in the synchronous configuration.*

Sync PPP Call Script

```
/etc/ppp/peers/isp_wanpipe1:
```

ttyWP0	Wanpipe TTY driver
sync	Connect via sync line
<IP local>:<IP remote>	Set to IP addresses obtained from ISP.

✪ *Note: The baud rate is not needed since the Sync Wanpipe TTY drivers obtain the baud rate from the Wanpipe configuration files (wanpipe1.conf).*

To start the pppd daemon using the above script, use this command:

```
pppd call isp_wanpipe1
```

Multilink PPPD Configuration

One of the major reasons for WANPIPE TTY driver development was MULTILINK PPP. The 2.4.X Kernels support this protocol, which can bundle multiple WANPIPE T1 cards into a singe logical connection to achieve greater throughput.

MULTILINK PPP protocol can be used in Sync or Async mode. The following configuration changes need to be applied to the above pppd call scripts in order to support multilink.

The /etc/options.ttyWPX files do not change. Only the call scripts do:

Async PPP Call Script, MULTILINK Support

Configuration for the first 56K line

```
/etc/ppp/peers/isp_wanpipe1
```

ttyWP0	First Wanpipe device
38400	Baud Rate
multilink	Enable Multilink support
connect '/etc/ppp/redialer'	A modem dial up script

Configuration for the second 56K line

```
/etc/ppp/peers/isp_wanpipe2
```

ttyWP1	Second Wanpipe device
38400	Baud Rate
multilink	Enable Multilink support
noip	Slaves have no IP info
connect '/etc/ppp/redialer'	A modem dial up script

Sync PPP Call Script, MULTILINK Support

Configuration for the first T1 line

```
/etc/ppp/peers/isp_wanpipe1
```

ttyWP0	First Wanpipe device
sync	Connect via sync line
multilink	Enable Multilink Protocol
<IP local>:<IP remote>	Set to IP addresses obtained from ISP.

Configuration for the second T1 line

```
/etc/ppp/peers/isp_wanpipe2
```

ttyWP1	Second Wanpipe device
sync	Connect via sync line
multilink	Enable Multilink Protocol
noip	Slaves have no IP info.

To start the pppd daemon and bundle the two links together follow the example:

```
pppd call isp_wanpipe1
```

Once the first ppp connection comes up then start the second.

```
pppd call isp_wanpipe2
```

APPENDIX B - PPPoE Configuration

Follow the steps below to configure WANPIPE Drivers for PPPoE.

1. Run `/usr/sbin/wancfg` and select **ADSL Protocol**.
2. Select **Hardware Option**.
3. Probe for the ADSL card and select it.
   - Select **EncapMode=ETH_LLC_OA** (i.e.: Bridged Ethernet LLC over ATM encapsulation)
   - If you have ATM VPI/VCI information, disable ATM auto configuration and specify the VPI/VCI combination. Otherwise, try to autodetect the VPI/VCI numbers.
4. Select **Network Interface Setup** and specify **PPPoE** operation mode.
5. Save and Exit.
6. Start Wanpipe with the command: `wanrouter start`

Configure PPPoE Over the WANPIPE Interface

1. Download PPPOE utility from the Sangoma web site
   ftp.sangoma.com/linux/utilites/rp-pppoe-3.4.tar.gz
2. Untar the `rp-pppoe-3.4.tar.gz` in `tmp/` directory and read the README file.
3. Install `rp-pppoe: ./go`
4. Configure `rp-pppoe: adsl_setup.`
   - Use the WANPIPE network interface as the Ethernet interface connected to a modem.
   - Specify username and password as well as DNS info.
5. Start `PPPoE:  adsl-start.`

APPENDIX C - PPPoA Configuration

Follow the steps below to configure WANPIPE Drivers for PPPoA.

1. Run `/usr/sbin/wancfg` and select **ADSL Protocol**.
2. Select **Hardware Option**.
3. Probe for ADSL card and select it.
   - Select **EncapMode=PPP_LLC_OA** (i.e.: PPP LLC over ATM encapsulation)
   - If you have ATM VPI/VCI information, disable ATM auto configuration and specify the VPI/VCI combination. Otherwise, try to autodetect the VPI/VCI numbers.
4. Save and Exit
5. Start Wanpipe with the command: `wanrouter start`

Configure PPPoA Over the WANPIPE Interface

Configure PPPD daemon for a synchronous PPP connection over the `/dev/ttyWP` device created by the WANPIPE driver. The PPPD configuration consists of three files.

1. /etc/ppp/options.<ttyname>
2. /etc/ppp/peer/<isp_call_script_name>
3. /etc/ppp/pap-secrets or /etc/ppp/chap-secrets
   /etc/ppp/options.ttyWP0

Table 12 - Sync Options.ttyWP0 File

Option	Description
asyncmap 0	
silent	#Wait until ppp request is received before starting ppp protocol (optional).
Debug	#Enable debugging (optional).
--detach	#Do not detach from the terminal window.
defaultroute	#IP address of this interface should be set as default in the routing table.

Call pppd Scripting

A **call pppd** script can also be defined to simplify the pppd argument line. The call script must be defined in `/etc/ppp/peers` directory. The example call script will be called `isp_sync` as described below:

Sync PPP Call Script

```
/etc/ppp/peers/isp_wanpipe1:
```

ttyWP0	Wanpipe TTY driver
sync	Connect via sync line
<IP local>:<IP remote>	Set to IP addresses obtained from ISP.

Note: The baud rate is not needed since the Sync Wanpipe TTY drivers obtain the baud rate from the Wanpipe configuration files (wanpipe1.conf).

To start the pppd daemon using the above script, use this command:

```
pppd call isp_wanpipe1
```