WANPIPE™

Multi-protocol WANPIPE Driver for Linux®

USER'S MANUAL

Author: Nenad Corbic

Introduction

The WANPIPE routing/API solution is comprised of Sangoma “S” series intelligent adapters/cards, Linux device drivers and shell/GUI configuration and startup scripts. The Sangoma “S” adapters along with the WANPIPE device drivers provide a physical and WAN data link layer (OSI model) to the Linux IP Routing Stack. The Linux IP stack, using its internal routing table, routes packets to and from the WANPIPE drivers.

Installing WANPIPE Driver Package

Obtaining The Wanpipe Software

1. **Sangoma FTP**

   The WANPIPE package can be downloaded from the "/linux/current_wanpipe" directory located on Sangoma's FTP site (**ftp.sangoma.com**).

   The latest software version is under the following name:

   `wanpipe-beta-X.Y.Z.tgz`:Latest Development Release
   `wanpipe-X.Y.Z.tgz`: Latest Stable Release

   where XY.Z is a product version number. This archive should always be installed on a CLEAN Kernel.

2. **Sangoma CD-ROM**

   Mount the CD-ROM
   ```
   Ex: mount /dev/cdrom /mnt
   ```

   Copy the Wanpipe-X.Y.Z.tgz release from the CD-ROM Linux/ directory into the / directory of your system
   ```
   Ex: cp /mnt/Unix-type\ systems/Linux/wanpipe-X.Y.Z.tgz /
   ```

WARNING: THIS VERSION ONLY SUPPORTS S508/S514 and 508/S514/FT1 cards.

PLEASE CONTACT SANGOMA TECHNOLOGIES INC. IF YOU WANT TO UPGRADE YOUR OLD S502E or S503 CARD.

Choosing the Kernel Version

The latest WANPIPE packages contain patches for both 2.0.X and 2.2.X kernels. Please read the README files for more update information.

Unpack distribution archive

1. Log in as super user (root) and change your current directory to the root directory by executing 'cd /' command.

   Note: ‘/’ is the top most directory in the file system, NOT the user root home directory (which is usually /root).

   You can also use the 'su -' command to log in as root from any other user account.

2. Copy the WANPIPE distribution package, previously downloaded from Sangoma ftp site, to the / directory and un-tar it.

   ex: tar xvfz wanpipe-x.y.z.tgz

   The files from this distribution are copied into the /usr/local/wanrouter/ directory.

3. If you received your distribution on a diskette, insert it into a disk drive and execute the following command:

   ```
   tar xf /dev/fd0
   ```

   Note: The above command assumes that the diskette is in the primary disk drive (known as drive A: in MS DOS). If you want to use secondary drive (drive B:), substitute 'fd0' for 'fd1'.

Package Components

The WANPIPE Driver package includes the following components:

Driver Installation script	(/usr/local/wanrouter/Setup)
Driver start-up script	(/usr/sbin/wanrouter)
Driver configuration utility	(/usr/sbin/wanconfig)
Driver configuration files	(/usr/local/wanrouter/wanrouter.rc)
Interface configuration files	(/usr/local/wanrouter/interfaces/*)
Firmware modules	(/usr/local/wanrouter/firmware/*)
Kernel patch files	(/usr/local/wanrouter/patches)

Frame Relay Debug Monitor	(/usr/sbin/fpipemon)
Cisco HDLC Debug Monitor	(/usr/sbin/cpipemon)
PPP Debug Monitor	(/usr/sbin/ppipemon)
X25 Debug Monitor	(/usr/sbin/xpipemon)
WANPIPE Keyboard Led Debugger	(/usr/sbin/wpkbdmon)
WANPIPE PPP Backup Daemon	(/usr/sbin/wpbackupd)
Adapter's Memory Viewer	(/usr/sbin/sdladump)

FT1 CSU/DSU Configuration Util.	(/usr/local/wanrouter/config/cfgft1/cfgft1 also linked to /usr/local/wanrouter/cfgft1)
Wanpipe Configuration Utility	(/usr/local/wanrouter/config/wacfg/wancfg also linked to /usr/local/wanrouter/wancfg)

Sample files	(/usr/local/wanrouter/samples)
X25 API sample code	(/usr/local/wanrouter/api/x25)
CHDLC API sample code	(/usr/local/wanrouter/api/chdlc)
Frame Relay API sample code	(/usr/local/wanrouter/api/fr)

WANPIPE Package Setup

The WANPIPE package can be installed in two ways:

1. A first time WANPIPE installation with a clean, un-patched kernel.
2. An Upgrade to the latest WANPIPE release.

The **setup** installation script, located in /usr/local/wanrouter directory will guide you through ether installation procedure, whether it's a fresh install or an upgrade. The installation utility will probe the kernel for previous WANPIPE installations and will offer appropriate installation options.

Furthermore, the **setup** installation utility can also re-compile and install new WANPIPE device driver modules based on the currently running Linux kernel, thus eliminating manual kernel/module recompilation.

To start WANPIPE installation, change directory to /usr/local/wanrouter and run the **setup** installation program.

```
cd /usr/local/wanrouter
./setup install
```

This script will perform the following functions:

1. WANPIPE INSTALLATION
   - Verify distribution integrity and fix file permissions
   - Check for required packages, bash2, libncurses and GCC.
     Note: Wanpipe GUI utilities require BASH version 2 or greater.
     If the Setup fails the bash check, make sure bash in installed
     before attempting to configure WANPIPE.

2. UPDATING WANPIPE DEVICE DRIVERS: LINUX KERNEL
   - Probe the Linux kernel for previous WANPIPE installation
   - If the kernel is clean, appropriate patches will be used updated WANPIPE drivers.
   - If the kernel already has WANPIPE installed, drivers will be upgraded to the latest version.

3. WANPIPE KERNEL DRIVER SETUP
   - This option will attempt to compile currently installed WANPIPE device drivers located in the Linux source tree and update/install WANPIPE modules.
   - Note: the kernel source in /usr/src/linux must have the same configuration and version number of the currently running image. Otherwise, module dependencies will fail.
4. WANPIPE BOOTSTRAP CONFIGURATION
   - Option to install bootstrap startup and shutdown scripts that will start and stop WANPIPE drivers on system startup and shutdown.

5. WANPIPE META CONFIGURATION
    - Wanpipe utilities depend on a wanrouter.rc META file that defines locations of WANPIPE lock, configuration and interface files.
    - You will be prompted to define these paths. If unsure leave everything as default.

6. WANPIPE UTILITES SETUP
    - This option will compile all WANPIPE utilities necessary to run WANPIPE devices.

7. WANPIPE INSTALATION : COMPLETE
    - This is the last message in the installation process. This message will ONLY be displayed if the WANPIPE device driver modules have been compiled. In this case, the next step is to configure a WANPIPE driver and start the router.

      PLEASE SKIP THE NEXT SECTION AND PROCEED TO WANPIPE CONFIGURATION.

7. WANPIPE INSTALLATION: INCOMPLETE
    - This is also the last message in the installation process. This message will ONLY be displayed if the WANPPIE device drivers HAVE NOT been compiled, thus you are required to proceed with the kernel/module re-compilation in order to finish up the WANPIPE installation.

      PLEASE REFER TO THE LINUX COMPILATON INSTRUCTIONS BELOW

Linux Kernel Configuration/Compilation

Configure Linux Kernel

Change your current directory to /usr/src/linux:

```
     make menuconfig'      (menu- driven configuration)
or   make xconfig'         (if you are running X-Windows.)
```

Note:
If you have never configured Linux kernel before, please read /usr/src/linux/README or ask someone who did for assistance. In short, the Linux configuration script will ask you a series of questions regarding your system configuration. Most questions require single-letter answer: 'y' to enable feature, 'n' to disable it or 'm' to implement feature as a module. Entering 'h' brings up a help text (if available).

IMPORTANT:

When configuring the kernel, ***MAKE SURE YOU ANSWER*** **NO** to the following questions when they come up:

- Set version information on all symbols for modules (CONFIG_MODVERSIONS)
- Frame relay DLCI support (CONFIG_DLCI)

When configuring kernel, ***MAKE SURE YOU ANSWER*** **YES or M** to the following questions when they come up:

Code maturity level options

Prompt for development drivers	(CONFIG_EXPERIMENTAL)	

Loadable Module Support

Enable loadable module support	(CONFIG_MODULES)	
Kernel daemon support	(CONFIG_KERNELD)	***2.0.X kernels only***
	(CONFIG_KMOD)	***2.2.X/2.4.X kernels only***

General Setup

Networking support	(CONFIG_NET)	
PCI support	(CONFIG_PCI)	

Networking Options

Packet Socket	(CONFIG_PACKET)	***2.2.X / 2.4.X kernels only***
Unix domain sockets	(CONFIG_UNIX)	
TCP/IP networking	(CONFIG_INET)	
IP: forwarding/gatewaying	(CONFIG_IP_FORWARD)	***2.0.X kernels only***

IP: optimize as router not host (CONFIG_IP_ROUTER)
802.1d Ethernet Bridging (CONFIG_BRIDGE) **2.2.X kernels only,**
used by WANPIPE frame relay Ethernet bridging.

WAN Router (CONFIG_WAN_ROUTER)
set to "M" for 2.2.X or 2.4.X kernels
set to "Y" for 2.0.X kernels.

Network Device Support
Network device support (CONFIG_NETDEVICES)

Wan Interfaces
WAN Drivers (CONFIG_WAN_DRIVERS)
Vendor Sangoma (CONFIG_VENDOR_SANGOMA)
set to "M" for 2.2.X kernels / set to "Y" for 2.0.X kernels.

In previous version of WANPIPE, a maximum number of WANPIPE devices needed to be defined (CONFIG_WANPIPE_CARDS). In this release, all Sangoma devices are probed during module startup. Devices found are dynamically setup.

WANPIPE Frame Relay support (CONFIG_WANPIPE_FR)

Say 'Y' to this option, if you are planning to connect WANPIPE card to a frame relay network. If you say 'N', the frame relay support will not be included in the driver (saves about 16K of kernel memory).

WANPIPE Cisco HDLC support (CONFIG_WANPIPE_CHDLC)

Say 'Y' to this option, if you are planning to connect WANPIPE card to a leased line using Cisco HDLC protocol (WANPIPE or API). If you say 'N', the Cisco HDLC support will not be included in the driver.

WANPIPE PPP support (CONFIG_WANPIPE_PPP)

Say 'Y' to this option, if you are planning to connect WANPIPE card to a leased line using PPP protocol (WANPIPE). If you say 'N', the PPP support will not be included in the driver.

WANPIPE X25 support (CONFIG_WANPIPE_X25)

Say 'Y' to this option, if you are planning to connect WANPIPE card to a leased line using X25 protocol (WANPIPE or API). If you say 'N', the X25 support will not be included in the driver.

WANPIPE Multi-Port PPP support (CONFIG_WANPIPE_MULTPPP)

Say 'Y' to this option, if you are planning to connect WANPIPE card to multiple leased line, using the kernel PPP protocol (WANPIPE). Since the protocol is implemented in the kernel PPP protocol can run on both adapter ports. If you say 'N', the Multi-Port PPP support will not be included in the driver.

Filesystems

/proc filesystem support	(CONFIG_PROC_FS)

Re-build Linux kernel

If you never compiled Linux kernel before, please read /usr/src/linux/README or ask someone who did for assistance. Note that if your machine is slow and/or has little memory, re-building Linux kernel may take several hours.

After successfully configuring the kernel using 'make menuconfig' perform the following (from /usr/src/linux directory):

`make dep`	:to rebuild source code dependencies,
`make clean`	:to delete all stale object files, **Note:** Save your '/usr/src/linux/.conf' file to some other location and rename it to (ex: conf_old). This file contains your linux configuration; thus, if you ever have to reconfigure a fresh kernel, copy this file back into the linux directory, rename it to '.config' and run 'make menuconfig'.
`make bzImage`	(2.2.X / 2.4.X kernels) to build new kernel.
or `make zImage`	(2.0.X kernels) to build new kernel. **Note:** 'make zImage' will fail if the kernel size is too big. Thus, use 'make bzImage'. (for 2.2.x kernels)

IMPORTANT:

Before we proceed to build new modules, it's advisable to remove the old, already existing modules.

- Change directory to /lib/modules
- Rename the "kernel_name" directory to "kernel_name.old".
  (Ex. If we are working with kernel 2.2.14, then rename the 2.2.14 directory to 2.2.14.old)
- Change directory back to /usr/src/linux.

`make modules`	to build kernel modules,
`make modules_install`	to install modules.

Updating LILO

This section only applies if the kernel was compiled using bzImage or zImage.

After successful compilation, copy the new image usr/src/linux/arch/i386/boot/zImage (or bzImage which ever was compiled) into the /boot directory and rename it to vmlinuz_2.2.X (where X is the kernel number.)
(ex. cp /usr/src/linux/arch/i386/boot/bzImage /boot/vmlinuz-2.2.14)

Note:
Before you copy the new kernel into the /boot directory rename the old kernel image in /boot.
(ex. mv vmlinuz-2.2.10 vmlinuz-2.2.10.old)

In /etc directory, edit the lilo.conf file and update the changes. For more info run 'man lilo'.

`lilo` :execute lilo to update the changes to /etc/lilo.conf

When all this is done, reboot your machine to load the new image !!!

Enabling IP Forwarding

The IP Forwarding option enables kernel routing. By default this option is turned off for security reasons. If one enables IP forwarding the kernel will route all packets without any security policies. A more secure option is ipchains i.e. firewalling where one can control which addresses are routable and which are not.

Therefore, if the Linux machine is to be used as a router than enable IP forwarding by:

```
echo “1" > /proc/sys/net/ipv4/ip_forward
```

Note: If the computer is rebooted this option will go back to its default value:
The above command dynamically configures the ip_forwarding option.

To set the default value of ip_forward to TRUE, so it’ll always remain on.
Set `FORWARD_IPV4=true` in /etc/sysconfig/network (RedHat)
Set `IP_FORWARD=yes` in /etc/network/options (Debian).

Configuring WANPIPE

Next step in setting up your WANPIPE card, is to create a wanpipe#.conf configuration file where all hardware, protocol and interface information must be supplied. For ease of use and convenience, Sangoma has developed a GUI application that should be used to create this file.

Change directory to /usr/local/wanrouter/ and execute:

```
./wancfg
```

Note: `wancfg` is a symbolic link to /usr/local/wanrouter/config/wancfg/wancfg utility. Please read the README file in the above directory for more information about this utility.

IMPORTANT:
WANCFG 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 `wancfg` utility uses the `/bin/bash2` executable.

Note:

In previous release the `wancfg` used the `/bin/sh` variable that needed to be changed to point to `/bin/bash2`. This is not necessary any more.

Each wancfg option has a help option associated with it. These help files should be enough to successfully guide you through the WANPIPE configuration.

Starting And Stopping WAN Router

Before starting a WANPIPE device, make sure that the configuration file has been created using wancfg utility, and that a Sangoma “S” series card is present. In case of a failure please refer to the Troubleshooting section of this document.

Start WANPIPE:

run `'wanrouter start'` at the command prompt.
WANPIPE devices, defined in /usr/local/wanrouter.rc, WAN_DEVICES list will be started. If multiple devices have been configured, updated the wanrouter.rc file accordingly.

Stop WANPIPE:

run `'wanrouter stop'` at the command prompt.
WANPIPE devices defined in /usr/local/wanrouter.rc, WAN_DEVICES list will be shutdown.

Start a particular WANPIPE device:

run `‘wanrouter start wanpipe#’` at the command prompt (where # = 1 to 16).
A single WANPIPE Device will be started regardless of the /usr/local/wanrouter.rc file.

Stop a particular WANPIPE device:

run `‘wanrouter stop wanpipe#’` at the command prompt (where # = 1 to 16.)
A single WANPIPE device will be stopped regardless of the /usr/local/wanrouter.rc file.

List all active WANPIPE devices:

run `‘wanrouter list’` at the command prompt.

Check the WANPIPE version:

run `‘wanrouter version’` at the command prompt

Configuring the onboard CSU/DSU

Sangoma S508FT1 and S514FT1 cards come with an on board CSU/DSU, that needs to be configured separately using the `/usr/local/wanrouter/cfgft1` utility.

Before proceeding with the CSU/DSU configuration:

- The CSU/DSU configuration information must be obtained from the T1 provider.
  Check with the T1 provider at what speed the fractional T1 line is set to. (ex 64K, 128k ... 1.5M). The fractional T1 consists of 24 channels each 64K. Using the CSU/DSU one configures 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.

The configuration utility `' cfgfg1'` is found in `/usr/local/wanrouter` directory.
Note: The cfgft1 is symbolic link to the /usr/local/wanrouter/config/cfgft1/cfgft1 utility.

IMPORTANT:
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.
- 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.

Note:

In previous release the `cfgft1` used the `/bin/sh` variable that needed to be changed to point to `/bin/bash2`. This is not necessary any more

The CFGFT1 utility contains all help files necessary to configure the CSU/DSU.
Change directory to /usr/local/wanrouter

```
./cfgft1 wanpipe1
```

The CFGFT1 configurator has three mode of operations.

- **Standard Configuration**
  This is the simplest configuration method. This method should be used by default and will be sufficient for 99% of the configuration cases.
- **Advanced Configuration**
  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**
  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 the Appendix B

Dynamic WANPIPE Configuration

Current WANPIPE drivers support dynamic configuration; thus, interfaces can be brought down, reconfigured and restarted while the WANPIPE device is active.

This is a very useful feature when using multiplexed protocols like frame relay, where one physical link can support up to 100 logical channels. In previous WANPIPE releases, a card would have to be restarted in order to change a single logical channel. However, the current WANPIPE drivers can be reconfigured dynamically using the `'wanconfig'` utility.

```
wanconfig [ -hvw ] [ -f <config-file> ] [ -U {arg options} ]
          [ -y <verbose-log> ] [ -z <kernel-log> ]
          [ card <wan-device-name> [ dev <dev-name> | nodev ] ]
          [ help | start | stop | up | down | add | del | reload
                 | restart | status | show | config ]
```

Note, the `wanconfig` utility configures the wanpipe drivers based on information contained in wanpipe#.conf file. Therefore, before driver re-configuration, one must update the wanpipe#.conf configuration file.

WANCONFIG Syntax:

- Unconfigure and deallocate all resources for interface wp1_ppp on card wanpipe1:
  Note: The network interface must be brought down first using `ifconfig`().

```
     wanconfig card wanpipe1 dev wp1_ppp down
or   wanconfig card wanpipe1 dev wp1_ppp stop
```

- Create and allocate resources for interface wp1_ppp on card wanpipe1,
  use data from wanpipe1.conf to (re)configure the interface.

```
     wanconfig card wanpipe1 dev wp1_ppp up
or   wanconfig card wanpipe1 dev wp1_ppp start
```

- Unconfigure and deallocate all interfaces on a single card:

```
     wanconfig card wanpipe1 down
```

- Configure and allocate all interfaces on a single card:

```
     wanconfig card wanpipe1 up
```

Note: For more information execute wanconfig –help.

Frame Relay Example:

Assuming that frame relay WANPIPE was configured as follows:

```
DLCI Numbers        16             17
Interface names     wp1_fr16       wp1_fr17
```

```
Local IP            201.1.1.1      202.1.1.1
PointoPoint IP      201.1.1.2      202.1.1.2
```

Thus, `ifconfig` displays interface names wp1_fr16 and wp1_fr17.

We would like to add a new DLCI number 18 with
local ip: 203.1.1.1 and remote ip 203.1.1.2:

- Run wancfg configuration utility and add a DLCI number/interface to the wanpipe1.conf configuration file. Call the interface wp1_fr18.

- Create and allocate resources for the new interface on card wanpipe1:
  ```
  wanconfig card wanpipe1 dev wp1_fr18 up
  ```

- Configure the ip addresses for wp1_fr18 and startup the interface using ifconfig.
  ```
  ifconfig wp1_fr18 203.1.1.1 netmask 255.255.255.0
                              pointopoint 203.1.1.2 up
  ```

- Run `ifconfig` to confirm that the changes took place.

  Note: `ifconfig` is a standard Linux tool for configuring network interfaces. For more information refer to ifconfig man page.

Using Protocol Monitors/Debuggers

Included with the Wanpipe drivers are UDP management monitors for each of the supported protocols.

fpipemon :	for Frame Relay protocol.
cpipemon :	for CHDLC protocol.
ppipemon :	for PPP protocol.
xpipemon :	for X25 protocol.
wpkbdmon:	for keyboard led statistics/debugging

You must be superuser (root) to run these monitors.
Command line Usage: invoke command on prompt (ex 'fpipemon')

Note:
The UDP management port to be utilized can be configured with the UDPPORT option in the wanpipe#.conf configuration file.

The monitor talks to the driver through a special UDP packet like an intelligent ping. All that is required for the monitor to work is that the UDP packets must get into the driver, and that the machine with the monitor must be reachable. You can therefore run the monitor to get user statistics from ***any machine that is network connected to the Sangoma host***.

There are two parameters needed to access the link information using these monitors:

- The UDP Port set in the wanpipe#.conf configuration file. The Default is UDP port 9000. Setting the UDP port to 0 disables the monitor. You can change the UPD port to distinguish between different Sangoma cards on a network

- An IP address that forces the stack to try to send the UDP packet through the wanpipe driver. For instance, if you are on the host machine which has the card installed, you would use the IP address of the **opposite end of the link**. Note that you would ***not use the IP address of the wanpipe card itself*** if the UDP was coming “from the top”, because the stack loopback function returns the UDP packet before it even reaches the wanpipe driver. If wanpipe is the default gateway, then any address not on the local network will do. The address does not have to exist, as the UDP packets are intercepted in the driver before they are transmitted, and are replied to in the form of a UDP response packet sent as if it was received off the link.

  You can also use the name of the interface instead of an IP address, and the monitor will automatically resolve that into the address of the other end of the Point-to-Point.

For example, on the network below, assume that the link FR is supported by two Sangoma cards.

```
                                LAN
     --- fr2    FR      fr1 ---  eth0       ?????  eth1  ---
    | A |------------------| B |-----------?????-------| C |
     ---    wanpipe link  ---               ?????        ---
 IP: 201.1.1.1        201.1.1.2    192.168.1.5     192.168.1.4

 UDP PORT: 9000         9001
```

Network Interfaces: Machine A = fr2
Machine B = fr1 and eth0
Machine C = eth1

From Machine B, the wanpipe host, you would use the following command to read the modem status on the card in B:

```
fpipemon -i fr1 -u 9001 -c xm
                  OR:
fpipemon -i 201.1.1.1 -u 9001 -c xm
```

Note the address used!

From Machine C, you would use the same IP address. Machine C would not know about interface fr1, so you have to use an explicit IP address:

```
fpipemon -i 201.1.1.1 -u 9001 -c xm
```

The packet is routed from C over the ethernet to B where it attempts to send the packet to A. The packet gets intercepted by the driver, the request is answered and the response is sent back to C.

Machines B and C can also access data on A. In this case, there is no loopback route in the way (we are coming from “underneath”). So we can use A’s address:

```
fpipemon -i 201.1.1.1 -u 9000 -c xm
```

Note that it is the same address as in the previous example, but because the UDP port is different, we get the data off the card in A, not B.

Likewise, A can access information off the card in B:

```
fpipemon -i fr2 -u 9000 -c xm
                    OR:
fpipemon -i 201.1.1.2 -u 9000 -c xm
```

Security

Security considerations for the monitors is handled as follows:

- You can simply turn them off by setting the UDP PORT to 0 in wanpipe#.conf.
- Certain operations are only allowed from "above". For instance, you can only run the traces from above the stack. Any commands that can change the state of the link (such as testing the DSU/CSU) are only allowed from above the stack.

- You can set the TTL for the replies to the UDP packets in wanpipe#.conf. So for instance, if TTL is set to 1, only users that are logged in will be able to execute configuration and debugging commands.

For further information view the debugger help files by executing: `./(f,c,x,p)pipemon`

Trouble Shooting and Diagnostics

If you encounter errors during installation and/or start-up, try to determine which command, utility or process causes the error. Carefully record and examine all warnings and error messages. Read appropriate section of this manual again and make sure you have not missed anything important.

Consult the FAQ on Sangoma Web Page www.sangoma.com.

Router Initialization Problems

When the WAN Router start-up is complete, all WAN devices defined in the [devices] section of /etc/wanpipe#.conf file should appear in the /proc/net/router directory.

Similarly, all network interfaces defined in the [interfaces] section should appear in the /proc/net/dev directory (provided that corresponding WAN device was successfully configured).

If some of the devices and/or interfaces are missing, then verify configuration file and check '/var/log/wanrouter' and '/var/log/messages' files for error messages and/or warnings.

WANPIPE Driver logs extensive diagnostics during its initialization into the system log file. To monitor logged system messages use the following command:

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

You can watch for any errors and/or warnings during WANPIPE loading and initialization.

Router Configuration Problems

To verify the WAN device configuration, use Config and Status pseudo-files found in /proc/net/wanrouter directory. Doing 'cat' on these files reveals current device state and configuration. The sample output is shown below:

```
        cat /proc/net/wanrouter/Config

Device name | port |IRQ|DMA|mem.addr|mem.size|option1|option2|option3|option4
wanpipe2    |0x364 | 05|0  |0xEC000 |0x2000  |   5030|   7200|   65536|   6200
wanpipe1    |0x300 | 12| 0 |0xEE000 |0x2000  |   5080|  16000| 131072 |   5800

        cat /proc/net/wanrouter/Status

Device name |station|interface|clocking        |baud rate| MTU |ndev|link state
wanpipe2    | DCE   | RS-232  |external |  56000  | 1600 |  2   |connected
wanpipe1    | DTE   | RS-232  |external |  56000  |  256 |  2   |connecting
```

To obtain device statistics, do 'cat' on WAN device entry in /proc/net/wanrouter directory as follows:

```
cat /proc/net/wanrouter/{device}
```

Router Interface Problem

To verify TCP/IP configuration of WAN interfaces, use Linux' ifconfig and route utilities. Interface status and statistics can be viewed with

'ifconfig {interface}' command. The output should look similar to this:

```
wan0 Link encap:UNSPEC       HWaddr (whatever)
    inet addr:<local_ip_addr> P-t-p:<remote_ip_addr>Mask:<netmask>
            UP POINTOPOINT RUNNING  MTU:1500  Metric:1
            Rx packets:0 errors:0 dropped:0 overruns:0
            Tx packets:0 errors:0 dropped:0 overruns:0
            Interrupt:9 Base address:0x360 Memory ee000-effff
```

The interface name will be the one that was defined in /etc/wanpipe#.conf [interfaces] section. If this interface doesn't show up, after wanrouter start (check using ifconfig) a mistake was made while configuring the wanrouter.

Make sure that interface name in wanpipe#.conf [interfaces] section is the same as the file name in /usr/local/wanrouter/interfaces/{name}

ex. /etc/wanpipe#.conf

```
[interfaces]
wp1_ppp = wanpipe1, ,WANPIPE
```

/usr/local/wanrouter/interfaces/wp1_ppp

Make sure all fields are correct in /usr/local/wanrouter/interfaces/{name} file. Refer to section 2.2.

Router Table Problem (Cannot PING)

View routing table by typing 'route -n' at the command prompt and make sure that routes to all remote networks and hosts exist. The output should look similar to this:

```
Destination     Gateway Genmask         Flags   Window  Use     Iface
<remote_ip_addr> *      255.255.255.0   UH      0       0       wan0
```

Also make sure your ip_forwarding is turned on:

`cat /proc/sys/net/ipv4/ip_forward` should be set to 1.

If you are running a firewall check your policy table make sure you have
no typos.

If all the above looks ok, you should be able to ping remote host. If pings are not getting through, try to determine which part of the connection fails. There can be at least four possibilities:

- pings are not being transmitted by the local machine
- pings are not being received by the remote machine
- pings are not being replied to by the remote machine
- replies are not being received by the local machine

Quite often you can tell which part of the connection is not functioning by simply looking at the interface statistics output by the ifconfig utility at both local and remote machines.

Sdla Dump Utility

A debugging utility, sdladump, is also included with the distribution which allows viewing adapter local memory in hexadecimal dump format during run-time. Its command line syntax is:

```
sdladump {device} [{offset} [{length}]]
```

where: {device} name of the WANPIPE adapter, as it appears in the /proc/net/wanrouter/directory, e.g. wanpipe1

{offset}	adapter local memory address. Default is 0
{length}	size of the adapter memory area to be dumped. Default is 256 bytes (0x100)

Note that adapter must be configured before you can use sdladump utility.

Proc File System Statistics

Another useful debugging utility is the proc file system. You can use the following command :

```
cat /proc/net/dev
```

to get the information about your interface.

In this case ppp0 is the interface built using WANPIPE. This information gives you how the driver for your device is behaving. Following is the explanation of the heading used:

packets :	States the number of TX/RX packets
errs:	States the number of TX/RX retries. The packet was not sent and will be retried. Please note that this is not an error count, in spite of the name.
drop:	States the number of TX/RX dropped packets. This occurs when the link (interface) is down and application is trying to transmit.
fifo:	Not used by WANPIPE
colls:	States the number of TX/RX collisions. This means that the device was busy and the packet will be retried.
carrier:	Not used by WANPIPE

Line Problems (T1)

See www.sangoma.com/linedebug.htm.

Technical Support

If you are unable to diagnose and/or fix a problem yourself, you can ask Sangoma Technologies Inc. for technical support via fax or e-mail:

FAX: (905)474-9223
E-mail: ncorbic@sangoma.com or dm@sangoma.com

When sending your request, please provide us with the following information:

- Detailed description of the problem
- System type (CPU, speed, RAM size)
- Linux kernel version number
- WANPIPE driver version number
- Adapter type
- Communications link type and parameters (e.g. line speed)
- Interface type (RS-232/V.35), DSU/CSU type (if any), etc.
- Contents of the following files:
  - /etc/wanrouter.rc
  - /etc/wanpipe#.conf
  - /var/log/wanrouter
  - /var/log/messages
  - /usr/local/wanrouter/interfaces/*
- Output of the following commands
  - cat /proc/interrupts
  - cat /proc/ioports
  - cat /proc/net/dev
  - cat /proc/net/wanrouter/Config
  - cat /proc/net/wanrouter/Status
  - cat /proc/net/wanrouter/{device}
  - ifconfig {interface}
  - route -n

All requests for technical support are normally replied to within 24 hours (excluding weekends and holidays).

Appendix A

Multiple PCI cards

The WANPIPE modules perform a hardware probe during an initial load. Probe information found is written into the /var/log/messages log. Thus, if there are multiple cards in the system, load the WANPIPE modules using the modprobe command and view the log file to determine the slot numbers of each card.

`modprobe wanpipe` : to load WANPIPE modules into the kernel.
`modprobe –r wanpipe` : to unload WANPIPE modules.

Appendix B

The `/usr/local/wanrouter/cfgft1` utility supports the advanced CSU/DSU configuration option. The advanced configuration is text based and the following commands can be used to configure the on board CSU/DSU.

Advanced CSU/DSU Options

The Sangoma S508/FT-1 DSU has three memory banks:

- The factory default.
- The user configured bank.
- The scratch pad (working memory).

To check the current settings (bank settings), type **D0** <ENTER>.

Configuration changes are made as follows:

- Type C <ENTER> to open the configuration mode.
- Change the required settings.
- After making these changes, type D1 <ENTER> to see the temporary configuration bank.
- If these changes are correct, then type W<ENTER> to write the configuration to the non-volatile memory.
- Type Q <ENTER> to exit the configuration mode.

COMMANDS

+n	Add channel n (where n is 1-24) to the active list
-n	Delete channel n (where n is 1-24) from the active list
M0	Set framing mode to ESF (ExtendedsuperFrame)
M1	Set framing mode to D4
K0	Set encoding mode to B8ZS

K1	Set encoding mode to AMI
L0	Set line build out to CSU (0db) or DSX-1 (0-133ft)
L1	Set line build out to DSX-1 (133 - 266ft)
L2	Set line build out to DSX-1 (266 - 399ft)
L3	Set line build out to DSX-1 (399 - 533ft)
L4	Set line build out to DSX-1 (533 - 655ft)
L5	Set line build out to CSU (-7.5 db)
L6	Set line build out to CSU (-15 db)
L7	Set line build out to CSU (-22.5 db)
P0	Set clock mode to normal (SLAVE)
P1	Set clock mode to master (LDM mode only)
W	Write configuration to non-volatile memory
F	Get factory default configuration
Q	Exit configuration mode and move temporary (scratch pad) configuration to default configuration
D0	Display actual configuration
D1	Display configuration in temporary memory
E0	Unit does not echo commands sent to it
E1	Unit echos back commands sent to it
T0	User test modes enabled
T1	User test modes disabled

NB: Whenever you have finished your changes, you MUST do a WQ in order for them to "take". After that do a D0 to check that the changes have, indeed been written to permanent memory.

Advanced Configuration S514-S508/FT-1 example

Required configuration:
Channels 1-3 disabled,
Channels 22 and 23 enabled
Line build set to DSX-1 (399 - 533ft) ESF
framing mode Clock mode to normal

Type: `C-1-2-3+21+22K0L3M0P0D1WQ<ENTER>`

OR

`C<ENTER>`	Open configuration mode
`-1-2-3<ENTER>`	Remove channels 1, 2 and 3 from the active list
`+21+22 <ENTER>`	Add channels 21 and 22 to the active list
`K0<ENTER>`	Set the encoding mode to B8ZS
`L3<ENTER>`	Set the line build out to DSX-1 (399 - 533ft)
`M0<ENTER>`	Set the framing to ESF
`P0<ENTER>`	Set the clock mode to normal
`D1<ENTER>`	Display the temporary configuration before saving

```
W<ENTER>   Update the user memory
Q<ENTER>   Close the configuration mode
```

If you are adding or removing many D0 channels, break up the command into 2 or more commands, as the command line buffer has limited length. For instance, to drop the line speed to 128kbps you need to remove channels 3 to 24.

Do:

```
C<ENTER>
-3-4-5-6-7-8-9-10-11<ENTER>
     -12-13-14-15-16-17-18-19-20-21-22-23-24<ENTER>
WQ<ENTER>
D0<ENTER>
```

and the correct configuration should be shown.