The Fairisle Network Console

Richard Black

9th March 1994

1 Introduction

With the advent of a real number of port controllers and switches being used as switches a new phenomenon was observed. This was the desire for the console output of all the ports in a switch to be collected together on the same serial line - that of the "master" port controller of the switch. This document describes the protocol and mechanisms of the "Network Console".

2 Protocol

It is the producer of console output's responsibility to find some recipient, connect to them and deliver the console output to them.

2.1 Finding a server

A Wanda machine looks in its environment for the variable `TTYSERVER`. If present the string is parsed for a dotted notation address and port. The network console code will attempt to connect to, and deliver all its console output to that MSNL SAP. The well known port number 0.0.0.225 has been allocated for network console, and most recipients listen on this port number. If the network console believes that it is running in a ROM kernel (specifically one with no host information), then when it receives a rarp reply, it will use the MSNL address of the sender of the rarp reply, and the default port 0.0.0.225 and use that as the SAP of the recipient.

2.2 Data format

A network console block consists of a single integer in the first four octets which is, in network standard form, the number of characters of console output present in that block. It is followed by that number of characters. No sequence number or checksum is present, as the switch fabric is reliable.

3 Persistence

If a port crashes in such a way that the networking code is rendered inoperable, the diagnostic information about that crash cannot be sent over the network console connection. To avoid loosing this console information, the network console code attempts to keep a persistent buffer of pending output across kernel instances.

This buffer is kept in the last page of RAM, which is acquired specifically for this purpose from the VM system. A checksum is kept of the control fields in this buffer so that when the next network console starts it can check the validity of the buffer and forward any pending output from the previous instance.

It is interesting to note that the DRAMs decay sufficiently slowly that it is usually possible to recover the console log of a machine even if it has been switched off and on again, provided the power outage is only a few seconds.

At the time of writing, the boot roms in the FPC3 systems implement this protocol, but do not implement persistence. This usually means that any buffered output from a kernel will be forwarded and printed at the next occasion that a new kernel is booted, and not when the next console output is received from that machine. This can confuse the naïve user.

4 Servers

Network console daemon servers exist for both Wanda and Unix. The Wanda one is preferred, and is normally used in conjunction with `xscons` which presents the console output of the various ports in separate widgets.

4.1 Unix Network Console Daemon

A Unix Network Console Daemon was written to allow redirecting of the output of the network console output to a Unix process that has a MSNL listening process at one end and does TCP listening at the other end. Thus `xwcons` will be able to connect directly to the daemon (setup the machine.Console entry in the trader to be doing TCP connection).

The netcond daemon reads a configuration file to determine which Wanda network console client to serve. The format of the configuration file assume each line to indicate a machine in which this server is serving. The daemon will then check with trader to confirm that this information is indeed correct and obtains the port number for the TCP listen.

When a netcon client connects to the netcon server, it will register this entry with the netcon server and try to do a join of this netcon client and an existing xwcons TCP connection.

Correspondingly, when a TCP listened socket returns from select, we do an accept, register this new entries and try to join this with the corresponding netcon client.

Whenever any end of a join is broken, we destroy the join and free the registration associate with the socket.

The usage of the corresponding Unix tool is:

```
netcond [-d] [-l log_file] [-c config_file]
```

where -d specifies the debug mode, -p specified the alternative MSNL port other than the MSNL-PORT_NETCON defined, -l specifies alternative log file location (default /tmp/netcond.log) and -c specifies configuration file location (default to look at /etc/netcon.conf follow by ./netcon.conf).

Remember that the MSNL connection between the Wanda machine and the Unix daemon will be unreliable.