Revisiting the TNC firmware

Joachim Elen, ON1DDS
Unie der Belgische Amateurzenders – Radioclub Sint-Truiden (UBA-RST)
Naamsesteenweg 333, B3800 Sint-Truiden, België
e-mail: on1dds@qsl.net

Keywords
TNC, AX.25, SLIP, PPP

Abstract

This paper describes the implementation of a bridge in a TNC. By letting a TNC make the translation between AX.25 and a popular link layer protocol, countless new possibilities arise. Supporting SLIP or PPP would allow us to transparently attach our TNC to any device with a serial port, from personal computer to mobile handset. Users without specialized knowledge can start using complex network protocols like TCP/IP over radiofrequencies as they do on the Internet. This way, new networking technologies can be adopted or developed by radio amateurs.

Introduction

Except in gateways to the Internet, our European packet network generally does not support an advanced network protocol. Today, in 2001, we're still maintaining bulletin board systems. By nature, BBS's are driven manually, eliminating the necessity for further research on high-speed packet radio.

Packet radio might revive through a transparent introduction of modern network protocols like TCP/IP, Novell IPX, AppleTalk and others. SV2AGW[1] demonstrated this principle with his software. In this paper another approach is demonstrated.

Implementation

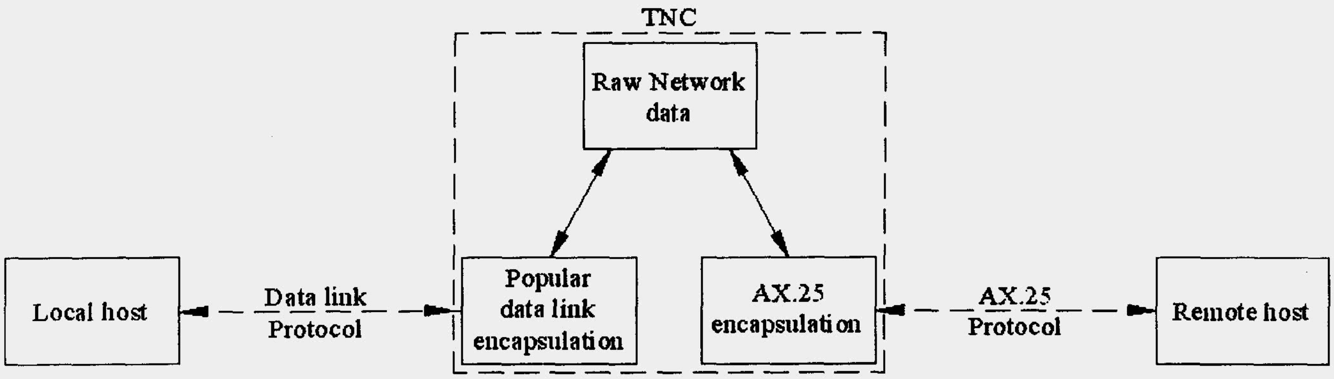
One major problem we have always been confronted with was the lack of hardware compatibility with any industrial communication standard. If we want to take advantage of the evolution in the commercial world, we have to commit ourselves to these standards and provide our users a 100% compatibility with them. We can revalue our TNC by making it work as a telephone modem and maintain AX.25 as our link protocol. The TNC can operate as a bridge - a device to translate AX.25 to another link protocol like PPP[2] or SLIP[3]. This would make the TNC a transparent device in either direction. Only this way we can guarantee (AX.25) net-connectivity for handheld devices and operating systems that don't support AX.25. This requires new developments on TNC firmware.

Command mode

Traditional modems generally use AT commands and S-registers. The same standard can be used for amateur radio purposes. Amateur specific parameters can be easily implemented as extended AT instructions and the S-registers can hold AX.25 specific flags and parameters. To go ONLINE, the traditional telephone number can be altered into the destination call sign with optional via's.

ONLINE mode

Once online, the TNC will communicate PPP or SLIP with the computer and AX.25 on the frequency. The bridge functionality that takes care of the translation between both the protocols is easy to understand and is depicted in the following diagram:



The AX.25 frame exchange itself can be accomplished in either VC or UI mode, depending on the personal preferences of the one implementing the functionality. Each mode has its drawbacks and advantages.

<u>AX.25 Unnumbered information</u>

The implementation of UI is relatively easy. A UI connection relies on a link quality better than 90%. Since missing frames have to be detected by a higher level time-out mechanism. By default, popular operating systems implement these timers with exponentially growing retry times (often not providing configuration options. Baud rates of less than 4800 baud with busy frequencies affect both these timers and the link quality (collisions). This leads to infinite delay times and timed out applications. Nevertheless, a good quality and a higher baud rate overcomes this problem and can guarantee acceptable data transmission.

<u>Virtual circuit</u>

The implementation of VC is much more complex. A VC connection does not demand links of such quality because AX.25 provides a reliable connection autonomously. The drawbacks are similar to the ones in UI mode. Not only low baud rates and busy frequencies, but also the transmission of acknowledge frames affects the higher level timers. Additionally, AX.25 forces retransmission of lost frames – again requiring bandwidth. These retransmissions inavoidably cause a distorted round-trip time calculation, ending up in a unnecessary datagram retransmission. On the other hand, if the TNC could discard datagram duplicates, this weird behavior could be better controlled. Usage of VJ[4] compressed TCP frames decreases bandwidth tremendously. I have tried to implement VJ compression in a UI environment, but it was useless because the link quality has to approach the 100%.

Routing

A telephone modem is limited to providing a single point-to-point connection. A TNC however is a point-to-multipoint device. One might think that the nature of SLIP or PPP prevents the TNC from making multipoint connections. This is not true. Keeping a limited ARP table – either static or dynamic – in TNC memory and directing each outgoing frame depending on its ARP resolution, allows us to set up a direct link to each host within reach. Networking systems like TCP/IP however compel client/server types of operation. Therefore, routing systems are only required when next to the default server one wants to connect a colleague radio amateur running services on its host or when multiple servers can be reached simultaneously.

MCB152[6] – an implementation

To check the true value of our philosophy, Walter Machiels, ON4AWM constructed the MCB152 TNC. This is a modular TNC, initially built for test purposes. The MCB152 is an 80c152jb based microcontroller board with 64 Kbytes of CODE memory and 64 Kbytes of DATA memory. The 80c152jb is an 8031HB with integrated SCC (Serial Communication Channel). A Baycom[5] USCC modem is to be plugged on the MCB152 and connected to the transceiver. The MCB152 is designed as a development board and has an EPROM containing a firmware loader. The upload is done using a copy command to the computer's serial port, but might, alternatively, be burned in EPROM. The Firmware was written by ON1DDS.

The current SLIP firmware implements an AT command interpreter with additional commands for packet radio purposes. The string in the following example sets TX-delay=15, slot time=7, TX-tail=0, persistence=63 and source call sign= "ON1DDS".
AT &TXD=15 &SLOT=7 &TXT=0 &P=63 C"ON1DDS"

To go online, one can use a command like AT DT ON0BAF.10@ON0EUL

The current firmware version processes AX.25 UI frames only, but a full AX.25 v2.2 implementation is under development. We currently use a 9600 baud DK9RR-FSK modem, but tests with a 153600 baud DF9IC/DG3RBU-FSK modem (at the highest baud rate) have proved to be functioning as well.

We have tested the MCB152 on most Microsoft Windows versions, Mac OS and Linux. None of them caused problems. Installation of a standard modem driver, normally delivered with the OS, is the only requirement. Dial in and ... enjoy TCP/IP packet radio using your favourite Internet software.

Unfortunately its not that simple, because this way of working requires an up and running TCP/IP server which, to prevent TCP/IP users from being isolated from the rest of the packet network must provide data exchange with other AX.25 users. We could use a NOS-like environment, but that would not be in line with our way of thinking – being as compatible as possible with the industrial standards. Therefore, Gert Leunen, ON1BLU has set up a TCP/IP server[7], running native Linux. A report on this project "TCP/IP and radio amateurism. - A UBA-RST TCP/IP Taskforce project" is presented elsewhere in this proceeding.

For backward compatibility, the AT @K command is implemented. This command switches the TNC to KISS mode. SV2AGW added this initialization string for the MCB152 in his software as well.

Conclusion

Although SLIP and PPP TNCs are no longer compatible with the traditional AX.25 in text mode, the advantages are obvious. Immediate compatibility with traditional telephone modems puts our TNC back in line with current Internet devices. Radio amateurs should be provided with the same environment as they are used to at home and at work. This is true for both hobbyists which prefer to surf the packet network for information as for professionals setting up a server. Hobbyists can focus on application programming, without worrying about compatibility issues any longer. Once again there is a need for developing high speed modems and dedicated transceivers. The need for good servers arises. These servers are 100% compatible with Internet servers, a good reason for young people to become radio amateurs. They can do tests that will not be allowed by providers on the real Internet and learn about upcoming technologies during their spare time. And last but not least, average packet radio users can take advantage of those developments without having to worry about the technical background.

UBA-RST TCP/IP TaskForce:

- **Gert Leunen (ON1BLU)** - *TCP/IP server setup, Belgian IP/DNS coordination*
  Homepage: http://www.qsl.net/on1blu
  AMPR-mail: on1blu@on0baf.baf.be.ampr.org
  Internet E-mail: on1blu@qsl.net

- **Joachim Elen (ON1DDS)** - *Firmware development*
  Homepage: http://www.8052.com/users/joachim.elen
  AMPR-mail: on1dds@on0baf.baf.be.ampr.org
  Internet E-mail: on1dds@qsl.net

- **Walter Machiels (ON4AWM)** - *Hardware development*
  Homepage: http://home.worldonline.be/~vda10786)
  AMPR-mail: on4awm@on0baf.baf.be.ampr.org
  Internet E-mail: walter.machiels@pandora.be

References

[1] ROSSOPOULOS, G. SV2AGW
homepage: http://www.raag.org/sv2agw

[2] PERKINS, D. *Point-to-Point Protocol: A proposal for multi-protocol transmission of datagrams over point-to-point links*. ARPANET Working Group Requests for Comments, DDN Network Information center, SRI International, Menlo Park, CA, Nov. 1989. RFC-1134

[3] ROMKEY, J. *A nonstandard for transmission of IP datagrams over serial lines: SLIP*. ARPANET Working Group Requests for Comments, DDN Network Information Center, SRI International, Menlo Park, CA, June 1988. RFC-1055

[4] V. JACOBSON. *Compressing TCP/IP headers for low-speed serial links*. ARPANET Working Group Request for Comments, Real Time Systems Group, Lawrence Berkeley Laboratory, Berkeley, CA, Feb. 1990. RFC-1144.

[5] BAYCOM, Bavarian Packet Radio Group
homepage: http://www.baycom.org

[6] The MCB152 project
homepage: http://www.caseconsole.com/mcb152

[7] LEUNEN, G. ON1BLU. *The linux TCP/IP server configuration page*
http://www.qsl.net/on1blu – Linux server section