

X.3 AND X.28 PROTOCOLS FOR TERMINAL NODE CONTROLLERS

Douglas Lockhart, VE7APU
9531 Odlin Road
Richmond, B.C. V6X 1E1
(604) 278-5601

Abstract

This paper proposes the adoption of an extended version of CCITT recommendations X.3 and X.28 for use in Amateur radio Terminal Node Controllers. The various X.3 parameters and X.28 commands and service signals (messages) are outlined and the extensions in place in the V-2 software implementation on the VADCG (Vancouver Amateur Digital Communication Group) TNC are discussed.

Terminology

This paper is semi-tutorial in nature and is intended for Amateur packet radio operators. Some of the terms used in the official X.28 protocol document have been translated into terms that hopefully may be more familiar to the readers. For example, the word 'TNC' is used instead of 'PAD' and the word 'terminal' is used instead of 'DTE' and in actual usage may be a microcomputer running host programs or a terminal emulation program. The terms I am using come from a variety of disciplines. My apologies if this makes it more difficult to understand rather than less so.

The Problem

In 1979, the VADCG TNC was the only one available and the software I had written for it only allowed two commands - control-Y caused a disconnect and a call sign followed by control-X caused a connection to be made. The very limited number of commands and the availability of only one version of software made it very easy for the new user to learn to use even though it was not very flexible.

The situation at the beginning of 1985 is quite different. There are now about 6 different TNCs available and three different protocols available, most with several different flavours or versions. Each combination of TNC and software has its own unique set of user commands. The more recent versions have a large and very flexible set of commands. In order to use the TNC efficiently, the user must learn his TNC's commands. With 40 or more commands frequently available, this becomes time-consuming. Furthermore, when the user tries to use the same commands on a different TNC or on the same TNC using a different link-level protocol, the commands are likely not to work and another set of commands must be learned. This learning process may be slowed by lack of advice from users with different command sets.

But perhaps the most serious problem that the lack of standardization of the command set causes is that application programs written for host computers will not be transportable. A special version for each case will have to be written. This situation has the potential of getting worse as new levels of protocols are developed, new TNCs become available and as new programmers begin to write software for TNCs.

The Solution

The commercial world ran into these same problems in interfacing ASCII terminals to the Packet Switched Public Data Networks (PSPDN) and developed a set of standards to solve their problems. The CCITT adopted Recommendations X.3 and X.28 in 1977. Recommendation X.28 defines the commands the terminal user sends to the PAD (Packet Assembler/Disassembler) and the service signals (messages) sent from the PAD to the terminal user. The PAD performs a similar function to the TNC (Terminal Node Controller) in the Amateur environment. Recommendation X.3 defines a set of parameters in the TNC which tailor its control of the terminal. In the ISO protocol model, X.3 and X.28 are a Level 6 or Presentation level protocol. To quote the OSI draft, "The purpose of the Presentation layer is to represent information to communicating application-entities in a way that preserves meaning while resolving syntax differences."

In August 1984 I received information on the CCITT X.3, X.28 protocols and realized that these standards could solve most of these problems if they could be implemented on all TNCs. On August 20, 1984 I wrote to the ARRL Ad Hoc Committee on Amateur Radio Digital Communication proposing the adoption of these standards. I also began recoding the V-2 software I was writing to use these standards. The proposal received a favourable response at the September, 1984 Committee meeting. The implementation described in this paper was completed in October, 1984. At the Committee meeting in November, 1984 I was asked to prepare a paper detailing my recommendations for a protocol based on the CCITT X.3 and X.28 protocols to be used in the Amateur packet radio environment. This is that paper.

The environment that the X.3/X.28 protocols were intended to work in is very similar to that found in most Amateur packet radio stations. However there are some minor differences which necessitate some deviation from the official recommendation. X.3/X.28 was intended to interface ASCII terminals and because of this, it only expected 7-bit ASCII characters to be used. In the Amateur environment, it is sometimes useful to transmit 8-bit binary data such as in object code. Some modification and extension of these protocols has been made in this recommendation to accommodate this type of usage.

Additionally, the terminal user in the commercial environment had no control over the tailoring of the operation of the data link level software. This recommendation includes extensions to standardize some of this control the Amateur user is already familiar with. While an attempt has been made to keep the X.28 commands and messages as similar to the official recommendation as possible, several have had to be changed to be more meaningful in the Amateur packet radio environment. These changes are similar to the type of changes made by commercial users when adapting these protocols to their own environments.

Recommendation X.28

The first problem encountered in establishing communication between the terminal and TNC is to initialize the TNC to the proper data rate and data format of the terminal. A sequence of characters are sent by the terminal to the TNC until the TNC recognizes the speed and format of the terminal and responds with an initial message. This function is commonly known as 'AutoBaud' or 'Autospeed' or 'Adaptive speed'. Unfortunately X.28 does not define any character sequence for implementing this function and in practice there are several different methods employed in the data communications industry. The current V-2 implementation on the VADCG TNC uses alternating '.' and ',' (period and comma) for detection of Baud rate and format. This system is superior to single character AutoBaud implementations because it can detect the type of parity and number of data bits used in addition to the Baud rate. In some applications it may be desirable to have the TNC always come up at a fixed Baud rate and data format and in this case the autoBaud function may be unnecessary.

After communication has been established between the terminal and TNC, both data for the link and X.28 commands for the TNC may be sent by the terminal user to the TNC. Likewise, messages generated by the TNC and data from the link can now be passed to the terminal at this time. Since both data and commands are being multiplexed over the TNC/terminal connection, the question arises as to how to differentiate the two types of data. The TNC can operate in two modes: while in 'command mode' characters received from the terminal are interpreted as commands and while in 'data transfer' mode the characters are interpreted as data intended to be passed on the link. The TNC will go from data transfer mode to command mode when it receives the X.3 command escape character defined by X.3 parameter #1. This command escape character may be set to any value by issuing an X.28 command to modify the setting of an X.3 parameter.

The problem of transmitting the command escape character as data is solved by transmitting two escape characters in succession from the terminal. This sequence is recognized by the TNC and causes the TNC to pass a single escape character to the link. By using this 'byte stuffing' technique it is possible to transmit any combination of characters as data to the link. This technique is convenient for terminal users and for file transfer of ASCII files but it is not full data transparency. For transmission of binary files by host computers whose file transfer program cannot handle this 'byte stuffing' technique, another system must be used. X.3 parameters may be set so that the TNC does not recognize any characters as having special meaning - including the command escape character. (See X.3 parameters 3, 12, 13 and 15) Now we have full data transparency.

But we have a new problem. After we have transferred our transparent data, how do we get back into command mode? We can't use a command escape character to do it so it has to be done another way. The setting of X.3 parameter 7 to a value of 8 causes the TNC to go into command mode when a break signal is generated by the terminal. Some terminals and computers are not capable of generating a break signal but can get back into command mode by resetting the TNC causing it to return to the initial default values of the X.3 parameters.

Resetting of the TNC seems like an extreme measure and other alternatives for users incapable of generating a break signal should be found. This is an area where the standard may need to be extended to meet the special needs of the Amateur Radio environment. My suggestion is to use a

special sequence of a number of characters followed by sufficient time to cause the idle timer to expire. (See X.3 parameter 4) At expiry of the timer, the TNC could check the preceding characters and enter the command mode if a match is found. This method should be virtually foolproof because a computer would never allow this much wait time while sending a transparent file. This is an area of further study.

Commands

X.28 commands are not case-sensitive. They can be typed in either upper or lower case or mixed case with the same results. Commands are terminated by carriage return <cr> or by '+'. Officially, X.28 has defined the following 8 commands:

```
SET <parameter list>
    To change X.3 parameter values
PAR? <parameter list>
    To display the current value of specified X.3
    parameters.
SET? <parameter list>
    Sets and then displays the current value of
    specified X.3 parameter values
PROF <identifier>
    To set the X.3 parameters to a standard set of
    values
STAT - Request status of TNC
CLR - To terminate a link connection
RESET - To reset a link connection
INT - To transmit an interrupt packet
```

This is a fairly short list of commands. This is because the function of most of the TNC commands we are familiar with is done by modification of the X.3 parameters.

There are a number of problems with using this command set in the Amateur packet radio environment which I will discuss in the following paragraphs.

X.28 does not define a command used to establish a connection because the format of such commands is very environment-specific. In the V-2 implementation the following command is used to establish a link-level connection:

```
CONNECT <call-sign><cr>
```

The call-sign is entered as an ASCII field in either upper or lower case. It is converted to upper-case and padded on the right with ASCII blanks. In the V-2 implementation this field is 7 characters long with the 7th character identifying an extension to the call sign. In implementations for other protocols, this field may be only 6 characters long. Additional information may need to be specified when establishing a connection when digipeating is used and when network level protocols are developed for Amateur packet radio. This information may optionally follow the call sign in the 'CONNECT' command but this is a subject for future consideration for incorporation into a standard. With a network level it may be desirable to use the 'CONNECT' command to establish a network level connection and use 'LINK' to establish a link level connection independently.

The X.28 'CLR' (clear) command is used to terminate a connection. Unfortunately, this command does not correspond to current Amateur packet radio conventions where the 'DISCONNECT' command is usually used. I propose we use the 'DISCONNECT' command for this purpose. When network level protocols are developed, the 'DISCONNECT' command can be used to terminate a network level connection and the 'UNLINK' command can be used to terminate a link level connection.

The X.28 'INT' command has no counterpart in the Amateur packet radio environment at the present time. There is no such thing as an 'interrupt

packet' in current link level protocols. When higher level protocols are implemented, there may be a need for this command. This is a subject for future consideration.

The 'SET' command should be implemented exactly as described in the X.28 recommendation. As an example, to set X.3 parameter 5 to 3 and parameter 21 to 128 type the following:

```
SET 5:3 21:128<cr>
```

The parameter numbers and values are entered as pairs of decimal numbers. Any blank or non-numeric character can be used as a separator between the numbers. The above could be entered as follows:

```
set 5 3 21 128<cr>
```

The X.28 'PAR?' command parameters is a list of X.3 parameter numbers whose values are to be displayed. The parameter numbers are entered in decimal.

The X.28 'SET?' parameters are the same as for the 'SET' command above. The function of the 'SET?' command is basically similar to that of the 'SET' command followed by the 'PAR?' command and because of this redundancy it has not been supported in the current V-2 implementation. However, since it may be more convenient, this command should be considered optional.

The X.28 'PROF' command sets the X.3 parameters to a particular set of values or 'profile.' The identifier in the command is a decimal number which identifies the particular profile desired. X.28 defines two profiles - simple and transparent but many networks offer other profiles tailored to the characteristics of a particular terminal or application.

The transparent profile is used when the TNC needs to be "invisible" to the user. Commands can not be used, no messages are generated by the TNC and editing and echo are not supported.

The simple profile supports commands and uses a set of X.3 function which require a minimum of terminal features. Messages are generated by the TNC and echo and flow control are enabled. Data is forwarded whenever a control character is received by the TNC.

The 'initial profile' is the set of X.3 parameters that are in effect when the TNC is initialized or powered up. It is usually similar to the simple profile. TNCs that support non-volatile RAM (NOVRAM) can have the ability to tailor their initial profile while others can only support a fixed initial profile burned into EPROM. The 'PROF' command is particularly useful in the latter case because it can usually reduce the amount of time required to tailor the X.3 parameters after powering on the TNC. It can also be useful when the TNC is sometimes switched to a different function where substantially different characteristics are required. This paper recommends that the 'PROF' command should be considered optional but desirable. It is not supported in the current implementation of V-2 but will be implemented in a future release.

The 'STAT' command takes no parameters but causes the TNC to display the current connect status of the TNC. This command is not presently implemented on V-2 since the connect status of the TNC is indicated by two other methods. However, this command should be implemented since TNC software capable of multiple links and connections will become available and the connection status of a TNC may be difficult to determine without this command.

The 'RESET' command currently is not supported because the current link level protocols do not support a reset packet. This command should be

implemented when new protocols are developed which support this function.

You will note the very small number of commands needed by X.28 to control a TNC. Most functions are performed by the setting of X.3 parameters using the 'SET' command. The reduction in the number of messages and commands can reduce the learning period for a non-English speaking user and possibly reduce the time for programmers to recode the software for use with a different language.

In general, functions which control an on/off type of condition or control a value that can be specified in a single byte can be controlled by X.3 parameters. For example, the TNC either echoes the characters from the terminal or it doesn't. Another example is the setting of the number of retries attempted on the link.

On the other hand, functions which cause immediate action or require the entry of multiple characters are unsuitable for incorporation into X.3 parameters and so require specific commands. An example of this is the 'CONNECT' command which requires the provision of a call sign and causes an immediate change in the operation of the TNC.

One function common to all TNC command sets is the ability to set the station call sign and so should be incorporated into a standard. I recommend the following command for this function:

```
MYCALL <call-sign>
```

This format is similar to that of the 'CONNECT' command described above. At the present time the V-2 implementation uses the 'CALLSIGN' command to perform this function but the 'MYCALL' command mnemonic is more familiar to users of TNCs and so I will be changing it in future releases.

The commands described in this section are not meant to be the only commands that should belong in a TNC. They are intended to be a set of commands that should be standardized in every TNC because of a common need for these functions. If a specific function is only used in one implementation there is no need for standardization. However, if a function is common to more than one implementation, then there should be an attempt made at providing a standard way to control that function. No doubt, as Amateur packet radio develops, new commands will have to be incorporated into this recommendation.

To summarize this section, this paper recommends the incorporation of the following six commands into every TNC.

SET, PAR?, CONNECT, DISCONNECT, MYCALL, STAT.
The PROF and SET? commands are optional.

Messages

X.28 messages are frequently called service signals. The public packet-switched networks do not usually implement the official X.28 message set precisely as it is defined. This is because the messages are not human engineered and in many cases may not be suitable for a specific environment. This is also the case in the Amateur Radio environment. However a number of the messages can be used. The following list is a subset of the official X.28 message set that I am recommending be used exactly as defined.

Linefeed

Acknowledgment of a command.

*

Command mode prompt signal.

xxx

Indicates that the line delete function is completed.

ERROR

Command error.

PAR <n:n>

Response to SET and SET? commands. The n's indicate the parameter number and value in decimal respectively.

PAR <n:INV>

Response to an invalid parameter setting in a SET or SET? command. n is the parameter number.

The following list contains messages which are not exactly as defined by the official X.28 recommendation but which I recommend be used in the Amateur packet radio environment.

LINKED <call-sign>

Indicates that a data link connection has just become established or a response to the STAT command when a data link connection is established.

UNLINKED

Response to the STAT command when a data link connection is not established.

LINKING <call-sign>

Response to the STAT command when a data link connection is being attempted.

UNLINKING <call-sign>

Response to the STAT command when a data link connection is being broken.

UNLINKED <call-sign> <reason>

Indication that a data-link connection has been broken or could not be established. The reason is given after the call-sign and can be one of the following:

BUSY - The indicated station is busy.

REM - The remote data link partner terminated the link.

REMERR - The remote station detected a protocol error.

NORM - Local DISCONNECT command.

NORESP - The station did not respond.

PROT - Protocol mismatch.

ERROR - Terminated by link protocol error.

TIMOUT - Excessive timeouts on link.

INV - Requested facility not supported.

Some of the reasons refer to facilities not supported in all link-level protocols. In this case, these reasons need not be used. However, when these facilities are available, they should be implemented as shown. Other reasons for link termination will be the subject of future standardization attempts.

It should be noted that the words, "link, linked, unlinked, etc." are used instead of the words, "connect, connected, disconnected, etc." even though the latter terms are in more common use at the present time. This was done to avoid future confusion because of terminology. In the ISO model, "connections" may be established at all levels. There are link level connections, network level connections, transport level connections, session level connections, etc. In the future, TNCs will likely have software to support multiple protocol levels, not just the link level as at present. TNCs capable of supporting multiple network level connections and multiple link level connections simultaneously will likely appear. When this happens, the term "connection" used without qualifiers will be ambiguous and cause some confusion. The term, "link" refers only to the data link layer and its use should reduce this potential problem. I believe that the term, "connection" without qualifiers should only be used to refer to a network level connection.

X.28 does not define the initialization message which appears when the TNC is initialized and I am making no recommendation in this area.

Recommendation X.3

In 1984, the official X.3 recommendation defined 22 parameters used to tailor the way the TNC controls the interface between the TNC and terminal or host computer. Unlike the X.28 recommendation which needed substantial modification to adapt it to the Amateur packet radio environment, these X.3 parameters can be implemented almost exactly as described in the official recommendation. For the Amateur packet radio environment, this paper recommends some extensions to this set to provide additional tailoring of the terminal interface, tailoring of the operation of the link interface and to control operation of the network level if implemented.

In October of 1984, the V-2 implementation on the VADCG TNC supported an additional 14 parameters beyond the 22 used by the official recommendation to provide this additional control. The parameters supported in this implementation are summarized in Table 1. The official X.3 parameter values which are not supported are indicated and extensions to the standard are also indicated. All parameters beyond 22 are extensions. The default values in the distributed software are also indicated. These defaults should not be considered as part of this recommendation but they do serve as an example of an 'initial standard profile'.

In this implementation the X.3 parameters have been set up as a contiguous string of bytes in memory. The first byte being the value of parameter 1 and it is recommended that the parameters be implemented in this way. Each byte has a set of legal values defined by this recommendation. The software should not allow non-legal values to be set into these parameters.

Parameter 1 defines the character sent to switch the TNC from data transfer mode into command mode. While the official standard only permitted values of 0 and 32-326 to be specified, I have extended this range to allow all 8-bit byte values. This was done because TNCs are frequently used with host computers rather than ASCII terminals and so are capable of transmitting 80bit characters. A value of 0 set in parameter 1 prevents escape from data transfer mode. by the transmission of a command escape character. Only one command can be issued from the terminal per escape. The TNC reverts back to command mode after each command is issued. Reception of the escape character also caused any accumulated data received from the terminal to be forwarded as a packet.

Parameter 2 controls whether the TNC will echo characters received from the terminal.

Parameter 3 defines characters which the TNC recognizes as a signal to forward any accumulated data received from the terminal and transmit it as a packet on the link. See parameters 1, 4 and 33 for other conditions which cause accumulated data to be forwarded. Note that values in parameter 3 can be added together to get some forwarding combinations that are not standard in the official recommendation. For example, a value of 10 can be specified to cause forwarding on carriage return (CR), DEL, CAN and DC2.

Parameter 4 defines a time interval of terminal inactivity which is used to forward any accumulated data as a packet. The timer is started each time a character is received by the TNC. If it expires, the data is forwarded. This forwarding method is disabled if the parameter is set to 0.

Parameter 5 defines the type of flow control indication is to be used by the TNC to regulate the flow of data from the terminal. In addition to the X-ON/X-OFF flow control method defined in the

official X.3 recommendation, the TNC can also control the RTS (Request to Send) line on the terminal interface to indicate whether it is ready to accept data or not. An active RTS line indicates that the TNC is ready to accept data. This extension was necessary to support the transfer of non-ASCII data (such as object code) by host computers to the TNC while still retaining flow control capability. The X-ON/X-OFF protocol is not capable of doing this.

Parameter 6 controls the use of X.28 messages by the TNC. These messages can be disabled as is frequently necessary when the TNC is connected to a host computer rather than a terminal or when the TNC is to remain transparent to the user. The network dependent format service signals are not supported because this requires a protocol layer that is not used in Amateur TNCs at the present time. It is a subject of future consideration when the protocols are in place that can support this function.

Parameter 7 determines the action of the TNC when it receives a break signal from the terminal. The only values that are supported by the V-2 implementation are 0 and 8. Value 8 provides another way to escape to command mode for terminals which support the break signal. It may be particularly useful for host computers wishing to switch from a transparent mode of operation after transferring a file. Values 1,2,4,5, and 21 are not supported because there are no interrupt packets, reset packets or any way to indicate the detection of a break in a packet using the current Amateur link level protocols. Their use should be reconsidered when higher level protocols which support these functions become available. Although not implemented at present, value 16 which supports the ability to flush TNC data for the terminal is useful.

Parameter 8 can be set so that the TNC will discard all data for the terminal. The only use for this feature at the present time is for testing purposes. When session level protocols are implemented in Amateur packet networks, this feature will become useful. It is easy to implement.

Parameter 9 controls the number of padding characters (ASCII nulls) which are sent to the terminal after every carriage return character. This is frequently required by hard copy terminals which require extra time to perform a carriage return. Some video terminals and terminal emulation programs require extra time as well. In the V-2 implementation the padding characters are actually sent but this function may be implemented by adding the equivalent amount of time fill as well. The official X.3 recommendation only allowed up to 7 padding characters but the V-2 implementation allows up to 255 padding characters or equivalent time fill.

Parameter 10 controls the point where long lines are broken or folded so that they display as multiple lines. This has its main use with RTTY terminals to prevent the last characters in a long line from overprinting at the end of the line. Some video terminals do not support line folding and may need to use this feature.

Parameter 11 indicates the speed of the terminal. Take special note that this parameter cannot be changed by the X.28 SET or SET? commands. It is a read only parameter that is set by the TNC initialization process or by the AutoBaud routine. Its only use at present is to determine what speed the terminal is running at if this is not known. It will have other uses when session level protocols are used in Amateur packet networks. The values indicated as not being supported are Baud rates that the V-2 AutoBaud routine does not support. Baud rates not in the list which are imple-

mented should use values above 18 and will be the subject of future standardization efforts.

Parameter 12 defines the type of flow control signals which the TNC will act upon to control the flow of data from the TNC to the terminal. The functions provided by this parameter have been extended beyond the official recommendation in order to be able to pass binary data to the terminal or host computer. An additional flow control method using the CTS (Clear to send) line from the terminal has been provided. If value 2 is set, the TNC will only send data to the terminal when the CTS line is active. Note that X-ON/X-OFF flow control cannot be used while transferring binary data. When X-ON/X-OFF flow control is specified here, the X-ON and X-OFF (DC1 and DC3) characters are not forwarded.

Parameter 13 controls the insertion of line feeds by the TNC after carriage returns either going to or coming from the terminal and after carriage returns being echoed back to the terminal. Any combination of these insertions may be selected.

Parameter 14 controls the number of padding characters (ASCII nulls) to be inserted after every line feed being sent to the terminal. This feature is used with terminals which take a long time to do a line feed operation. This is somewhat similar to parameter 9 and equivalent time fill can be used instead of sending padding characters.

Parameter 15 controls whether editing operations can be performed on data accumulated by the TNC in data transfer mode but which have not been forwarded yet. Editing is always available in command mode. The characters acted upon for the editing functions are defined in parameters 16, 17 and 18. Editing is normally used with terminals rather than host computers. The V-2 implementation does not turn off idle time forwarding automatically when editing is specified as per the official X-3 recommendation. Care should be taken with the specification of parameter 3 to ensure that data is not forwarded before there is a chance to edit it.

Parameter 16 defines the editing character the TNC acts upon to delete the previous entered character from the unforwarded accumulated data. The V-2 implementation uses the range 0-255 rather than 0-127 which is the official X.3 recommendation.

Parameter 17 defines the editing character the TNC acts upon to delete the unforwarded accumulated data. The V-2 implementation uses the range 0-255 rather than 0-127 which is the official X.3 recommendation.

Parameter 18 defines the editing character the TNC acts upon to redisplay the unforwarded accumulated data. This is mainly used to redisplay edited lines on printing terminals which do not have the ability to erase characters. The V-2 implementation uses the range 0-255 rather than 0-127.

Parameter 19 selects the type of signals that will be returned to the terminal when handling editing characters. The V-2 implementation allows for tailoring these signals for either printing or video display terminals. For example: for the delete previous character operation the TNC will return the character "/" for a printing terminal or the character sequence "<backspace>, <space>, <backspace>" for a video display terminal.

Parameter 20 is used to specify which characters will not be echoed to the terminal when echo is enabled by parameter 2. Note that the flow control characters X-OFF and X-ON are never echoed. Note that the values in this parameter can be added

together to get character sets which are not in the official recommendation. For example, a value of 6 may be specified which prevents echo of **LF**, **VT**, **HT** and **FF**.

Parameter 21 specifies whether the TNC will generate the parity bit in the data sent to the terminal and whether parity will be checked on data from the terminal. This parameter is not used in the V-2 implementation. The **AutoBaud** routine in V-2 generates a like parity to that supplied by the terminal but no action is taken on parity errors in data received from the terminal. The official X.3 specification seems to be vague as to what action is to be taken on parity errors or what type of parity (even, odd, mark, space, etc.) will be set by this parameter. However, I recommend implementation of this parameter when its function is clear and useful.

Parameter 22 specifies the number of line feed characters that the TNC will send to the terminal before stopping the sending of additional characters. The sending will recommence when an X-ON character is received from the terminal. This function allows the terminal user to read all the information before it is scrolled off the screen. The X.28 message "**PAGE**" is sent to the terminal to indicate the reason the data transfer to the terminal has been stopped.

Parameter 23 defines the number of additional characters that the TNC can receive from the terminal when the flow control method specified by parameter 5 acts to stop data from coming in from the terminal. This parameter and all remaining parameters described here are extensions to the official X.3 recommendation. This parameter was added to support a number of host microcomputers which do not act immediately on flow control signals. Some microcomputers only check flow control signals after the transmission of each line which could result in lost data if extra buffering capability was not available at the time the TNC acted to suspend the data flow.

Parameters 24, 25, 26, 27, 28 and 29 are extensions to allow for tailoring of the operation of the data link on the TNC.

Parameter 24 defines the number of consecutive link timeouts (no response that will be tolerated before "giving up"). This value is used only when a 'CONNECT' or 'DISCONNECT' command is in progress. When a link is established, the value in parameter 25 is used.

Parameter 25 defines the maximum number of consecutive link timeouts that will be tolerated on a data-link connection before "giving up". This value is only used when the link is established. See parameter 24 for the value used when the link is being established or terminated.

Parameter 26 defines the amount of time the TNC will wait for a response from a data link partner. If this time expires before receiving the expected response the TNC takes remedial action which is usually a transmission polling the partner for another response. The V-2 implementation on the VADCG TNC suspends this timer when the link is busy (a signal is detected on the channel) so that timeouts are not caused by use of the communications channel by other stations.

Parameter 27 allows for improved usage of the TNC in situations where it is not possible to reliably determine whether the link channel is occupied or not. This is frequently the case with noisy links such as are encountered with satellite and HF channels. When a value of 1 is specified, the TNC ignores the status of the carrier detect line and always assumes that the channel is clear.

Parameter 28 defines the amount of time the **RTS line** will be active before the data is transmitted on the data lines. This is used to support transmitters and modems which are slow in turning on. This time delay is also used to allow sufficient time for receivers to unquench so that the first part of the data frame is received without error. Note that the data will not be transmitted until the CTS line is active too.

Parameter 29 can be used to prevent links by other stations from being established. When the value is set to 1, the V-2 protocol responds to attempts to establish a data-link connection by indicating a 'busy' condition.

Parameter 30 is intended for future link control functions.

Parameter 31 controls whether the TNC will use the **AutoBaud** feature when re-initialized or use preset values for speed and format of the terminal interface. In TNCs which do not support non-volatile RAM, the value set by the terminal user will only be effective until the TNC is powered off.

Parameter 32 controls which packets received by the TNC will be passed to the terminal. If set to 0, the TNC acts as a filter which prevents data in packets not intended for this station from being passed to the terminal. Other values to provide different filtering actions should be the subject of future standardisation efforts.

Parameter 33 specifies the maximum number of characters received from the terminal which can be accumulated before forwarding the data in a packet. This determines the maximum number of bytes in the data field of a packet. The actual maximum frame length transmitted must include the overhead of the various protocol layers in use as well as this value and the maximum value that can be specified may vary with different protocol implementations. Care should be taken when specifying this parameter so that frames which are greater than the link partner's ability to handle are not generated. I recommend that all implementations be capable of handling values of at least 128 characters.

Parameters 34 through 38 are intended to be used to tailor the operation of the network layer. It is too early in the development of Amateur packet radio protocols to standardize network layer functions. The current use of parameter 34 in the V-2 implementation is shown in Table 1 but it is not the intent of this paper to recommend standards in this area.

Parameter 39 determines how the TNC controls the Receive Line Signal Detect (Carrier Detect) line on the terminal interface which is pin 8 on the standard EIA RS-232 interface connector. If set to 0, this line is always active. If set to 1, this line is only active when a data-link connection has been established. This feature is useful when the TNC is connected to a host computer.

Parameter 40 controls the masking of data bits in the characters being sent to and received from the terminal. If this value is 255, the TNC will pass 8-bit characters to and from the terminal. If the value is 327, the high order bit will be masked off (set to 0). Any bits can be masked off. The masking values are easier to understand when converted to binary.

Recommendation X.29

Although not discussed previously, it seems appropriate that CCITT recommendation X.29 be mentioned because it is closely related to the X.3 protocol. X.29 is a protocol which allows a remote

TNC to both inspect and change the **X.3** parameter values. This allows an application program to coordinate its operation with that of the remote terminal. For example, the application program could disable echo by altering X.3 parameter 2 while a password was being entered. In the ISO model, X.29 would be called a Level 5 or Session level protocol because it provides the means to coordinate the operation of the remote Presentation level (**X.3/X.28**). A number of significant advantages of the X.3/X.28 protocols will not be realized until a Session level of some kind such as X.29 is implemented in Amateur packet networks but it is not within the scope of this paper to make recommendations on Session level protocols for Amateur use.

Implementations

The only full implementation of the X.3 and X.28 protocols on Amateur TNCs that I am aware of is with the V-2 protocol on the VADCG TNC or **Ashby** boards. The **Ashby** board requires a hardware modification for the software to work. I understand that Terry Fox, **WB4JFI** with **AMRAD** is working on implementing this protocol with AX.25 on the VADCG TNC with his daughter boardmodification. He has been sent a copy of this implementation. TAPR representatives have indicated that they will provide X.3/X.28 as an optional command set on the TAPR board at some future date. Phil Karn, **KA9Q**, who is developing packet software for the Xerox 820 board is also interested in using these protocols. From discussions I have had with representatives of many of the Amateur packet radio organizations, I feel there is widespread support for the development of standards based on the X.3 and X.28 protocols.

Coordination of implementation

Some problems may arise when implementing these protocols in other TNCs or with other link-level protocols. The following paragraphs are guidelines for implementors.

Not all TNCs will support all the functions defined in the X.3 parameters. An invalid parameter message should be returned when an attempt is made to set an unsupported value.

Functions which can be implemented in X.3 parameters but which are not defined in the current recommendation should be implemented in parameters above 40.

When these new parameters are used in an implementation an attempt should be made to coordinate these parameters with other implementations so that the same parameter is not used for different functions or that different parameters are used for the same function. On request, I have volunteered to act as a coordinator for this effort within the ARRL Ad Hoc Committee on Amateur Radio Digital Communication. The Committee will use its facilities to act as a clearing house for information on **X.3/X.28** implementations for individuals who are planning extensions to these recommendations. Implementors should write to the Committee or directly to me at the address at the beginning of this paper.

Many functions are not suitable for control by X.3 parameters and will require extensions of the X.28 command and message set. An attempt to coordinate these extensions should be made. Implementors should communicate details of these extensions to me or the Committee as described in the previous paragraph.

Summary

The protocol described in this paper has been tested by many Amateurs in a number of countries with good results and it is estimated that several hundred thousand users of public data networks use the X.3 and X.28 protocols daily. The temporary disruption caused during the conversion from the old command set to the **X.3/X.28** protocols is only a small inconvenience when compared to the advantages of having a standard set of commands and messages for all TNCs. This inconvenience becomes even less significant when one considers the potential advantages of the X.3 parameters when session level protocols such as X.29 are developed for Amateur packet networks.

The initial recommendations made here will likely be fine-tuned as a result of future discussions, feedback and coordination efforts but the author hopes that this paper will lead to the development of a widespread standard set of commands and responses for terminal users of Amateur packet radio equipment and to give programmers a standard interface to TNCs so that they may write applications that will work on all Amateur TNCs. The recommendations made in this paper do not restrict the commands and messages used by TNCs but do propose standard ways of controlling certain common functions. No doubt, as Amateur packet radio networks mature, other functions will need to be standardized and other presentation level protocols will be developed for special environments.

Reference

[1] Shanahan, Teresa A., "Packet Assembly/Dissassembly (PAD) CCITT Recommendations x.3, X.28 and X.29" Data Transfer newsletter, Transmission #9, April 1984, Omnicom Information Service.

TABLE 1 - X.3 PARAMETER VALUES

* = default value
 # = not supported
 % = additional to X.3 standard

REFERENCE

1	Command escape	0 *27 1-26, 28-255	not possible ESC character optional values
2	Echo	0 *1	no echo echo
3	Data forwarding	0 1 *2 4 6 8 16 18 32 126	full packet only alphanumerics carriage return ESC, BEL, ENQ, ACK carriage return, ESC, BEL, ENQ, ACK DEL, CAN, DC2 ETX, EOT carriage return, EOT, ETX HT, LF, VT, FF all other characters below 32
4	Idle timer delay	0 *32 1-31, 33-255	no timer default value delay values in approximately tenths of a second.
5	Flow control to TNC	0 1 2 %*4	no flow control X-ON/X-OFF (data transfer) X-ON/X-OFF (data transfer and command) flow control with RTS line
6	Control of TNC service sigs	0 1 *5 #8-15	no service signals transmit service signals transmit service and prompt sigs network dependent format service sigs
7	operation on break	*0 #1 #2 #4 #5 8 #16 #21	no action interrupt packet reset packet indication of break service signal interrupt and indication of break escape from data transfer state discard output to terminal 1 + 4 + 16 combined
8	discard output	*0 1	normal data delivery discard output to terminal
9	Carriage return padding	*0 1-255	no padding number of nulls inserted after CR
10	Line folding	*0 1-255	no line folding number of characters per line

11	Binary speed	0	110 bit/s
		1	134.5 bit/s
		2	300 bit/s
		3	1200 bit/s
		4	600 bit/s
		5	75 bit/s
		6	150 bit/s
			1800 bit/s
		8	200 bit/s
		9	100 bit/s
		10	50 bit/s
		#11	75/1200 bit/s
		12	2400 bit/s
		13	4800 bit/s
		14	9600 bit/s
		15	19200 bit/s
		#16	48000 bit/s
		#17	56000 bit/s
		#18	64000 bit/s
12	Flow control to terminal	*0	no flow control
		1	X-ON/X-OFF flow control
		%2	flow control using CTS line
13	Line feed insertion	0	None
		1	after carriage return to terminal
		2	after carriage return from terminal
		4	after echoed carriage return
		5	values 1 + 4
		"6	values 2 + 4
			values 1 + 2 + 4 (data transfer only)
14	Line feed padding	*0	none
		1-255	number of nulls after line feed
15	Editing	0	off
		*1	on
16	Character delete	*8	BS (backspace+ character (control-H)
		o-7, 9-127	other characters
17	Line delete	*24	CAN character (control-X)
		0-23, 25-127	other characters
18	Line redisplay	*18	DC2 character (control-R)
		o-17, 19-127	other characters
19	Editing service signals	0	no editing service signals
		1	editing for printing terminals
		*2	editing for display terminals
		#8	editing using characters from range 32-126
20	Echo mask	*0	all characters echoed
			no echo of carriage return
		2	no echo of LF
		4	no echo of VT, HT, FF
		8	no echo of BEL, BS
		16	no echo of ESC, ENQ
		32	no echo of ACK, NAK, STX, SOH, EOT, ETB, ETX
		64	no echo of editing characters
		128	no echo of all characters in columns 1 & 2 plus DEL
#21	Parity treatment	#0	no parity detection or generation
		#1	parity checking
		#2	parity generation
		#3	value 1 + 2
22	Page wait	*0	no page wait
		1-255	number of line feed characters before waiting

%23 Buffer cushion	2-79, 81-254	*80	number of characters in cushion other values
%24 Linked timeouts	1-15, 17-255	*16	maximum consecutive timeouts linked other values
%25 Unlinked timeouts	1-4, 6-255	*5	maximum consecutive timeouts when not linked other values
%26 Line timeout	1-9, 11, 255	*10	approximately 5 seconds other values
%27 Duplex line control		*0 1 #2	half duplex line control always assume that channel is clear full duplex
%28 Clear to send delay	1-31, 33-255	*32	approximately 1 second other values
%29 Link control		*0 1	normal links to other nodes not permitted
%30 Unused link control parameter			
%31 Autobaud control		0 *1	use fixed UART defaults AutoBaud
%32 Received packet forwarding		0 *1	only pass packets when linked pass unlinked packets
%33 Maximum packet length	3-199, 201-250	*200	maximum output packet data bytes other values
%34 Network header 2nd byte	1-255	*0	NH second byte is 00 other values
%35 Unused network control parameter			
%36 Unused network control parameter			
%37 Unused network control parameter			
%38 Unused network control parameter			
%39 RLS (CD) line control		*0 1	always on indicates if link is established
%40 Data mask		*127 255 0-255	mask off high order bit pass all 8 bits in each byte range permitted