Overview of the TEXNET datagram protocol

T.C. McDermott, N5EG
Texas Packet Radio Society, Inc.
PO Box 831566
Richardson, Texas 75083

Abstract

This article is an overview of the TEXNET datagram network protocol. It is not intended as a specification. It deals with the protocol within the network (intra-network). It does not deal with layer 3 protocol issues between the user and the network (inter-network).

Introduction

The actual data-transfer and routing portion of a network protocol is the simpler aspect of the design. More difficult problems are encountered in administration and maintenance of a network. The majority of the effort in developing TEXNET centered on solving these network administration and initialization problems.

Definitions

These definitions are important:

USER - a user of the network. Normally thisuser utilizes a 2-meter radio and a TNC.

NODE - a TENXET network node containing 2-meter 1200-baud and 450 9600-baud radios, modems, and a Node-Control-Processor (NCP)

PHYSICAL CHANNEL - a serial channel implemented in hardware. For example: the 450 9600-baud radio is connected to a physical channel, the 2-meter 1200-baud radio is connected to a different physical channel. Each node has at least 1 physical channel, and may have up to 3 physical channels.

LOGICAL CHANNEL - An entry in a table defining an Ax.25 connection. Each physicai channel may have several logical channels active simultaneously. A logical channel may be either a user channel or a network-trunk channel. Each physical channel may have any number of logical channels, but a software-defined maximum of 20 is the practical limit.

LOGICAL CHANNEL NUMBER - a l-byte dynamic number associated with a particular AX.25 connection at a node. As connections come and go at a node, the numbers will be reassigned. May be either a network trunk or a user connection.

NODE NUMBER - a l-byte static number that defines a given node. The correlation between ASCII node names and node numbers is determined by the nework administrator when the EPROM is programmed for a given network node.

Protocol basics

TEXNET is a datagram-based protocol. BY this is meant that the two terminating nodes (the nodes at the entry and exit points of a particular circuit) maintain tables that specify how each particular user is connected. Intermediate nodes within the network do not have knowledge of any users (except those users that terminate at that intermediate node). The purpose of an intermediate node is to perform transit-routing. Thus each node must contain a data-blase that represents the entire network. Initializing and developing this database is a challenging task.

Layer 2

The ARRL specification "AX.25" is the standard for all layer 2 connections. There are two types of L2 circuits, user circuits, which are either version 1 or version 2, with reserved bits unused, and network trunk circuits, which are always version 2, but which are distinguished from user trunks by having reserved bits 4 and 5 in the AX.25 header (which are normally set to one when inactive) used. Bit 4 is used to indicate user (inactive, or 1) versus

network (active, or 0). It is required to use the bits in the header field because the PID field, which would normally be used for this function, does not arrive until an I-frame is sent, and indication of status is required prior to this time. Bit 5 is used for disconnect timer control.

The T3 time-out timer of AX.25 is redefined as follows:

Bit 5 = 1 (inactive) = T3 is the inactivity disconnect timer. No traffic on the logical channel for T3 seconds causes a disconnect.

Bit 5 = 0 (active) = T3 is the link keep-alive timer. This is the normal Ax.25 version 2 definition. Network trunks utilize this to provide the permanent virtual circuit function. This bit could also be utilized by a user to inhibit disconnect timing for trouble-shooting purposes. Normally the user should not utilize this function.

These bits were intentionally made hard-to-reach for 'normal' users so they would not be tempted to activate them.

Adherence to the 128-octet frame length limit is maintained. TEXNET will perform packet fragmentation if necessary to comply with this limit. This is done in order to maximize the number of frame-buffers available for processing. A large pool of frame buffers can enhance network performance under certain circumstances.

The sender and receiver of all packets is identified by the amateur-callsign of the node(s) and or user involved, per FCC requirements. No callsign reassignment is performed. TEXNET utilizes the *** LINKED message to cause bulletin boards to work properly.

Laver 3

The TEXNET layer 3 protocol consists of 5 bytes that head every network-trunk packet. Additionally, certain network administration packets may contain further information in the 'data' field which follows the network header. All network packets are sent as I-frames, with the exception of one network initialization command, which is sent utilizing the UI-frame.

Each network packet can be viewed as a series of bytes, starting with the first data byte within the I-frame. In order to minimize the number of data bytes of overhead, each node within the network is assigned a unique l-byte value as it's node number. Further, each user connection is assigned, dynamically, a l-byte value at each node. Thus 2-bytes specify both the node and the exact user connection involved. This allows 4-bytes to specify both ends of the connection. Information about both ends is included in order to allow error-recovery by intermediate nodes.

Network header block (NHB)

1st byte	NHB.RNN	= remote node number
	NHB.RLC	= remote logical channel number
	NHB.LNN	- local node number
	NHB.LLC	= local logical channel number
5th byte	NHB.NCF	- network control field
6-nth	data	= user data or network data (determined by control field)

The valid range of l-byte numbers is l-255. Zero is reserved to indicate an invalid node or channel.

The following list specifies the currently defined control field values. Values in the range O-127 are commands, and those in the range 128-255 are error messages (returned to the user as Network Information Code, NIC, plus the error number).

Decimal value	Definition
0 = NODEIN	Node initialization
1 = STATRQ	Current statistics request
2 - STATRS	Statistics response
3 - NODEID	Node identification + real time clock and date initialization
4 = POINTQ	Point update
5 - POINTR	Point response to update
6 = YSTATR	Yesterday's statistics request
16 = CONREQ	Circuit connect request
17 = CONACK	Circuit connect acknowledge
18 = DISREQ	Circuit disconnect request
19 = FLOWON	Set logical channel to not busy
20 = FLOWOF	Set logical channel to busy
21 = USERIN	Contains user data
32 = MYNAME	Distant node telling us his ASCII name
33 = TIMEUP	Update remote node's real time clock and date manually

Certain of the above commands have additional data included within the data field after the first 5 header bytes. These are defined subsequently.

Control fields of 48 to 63 are broadcast-type packets. Broadcast packets are sent to the entire network via controlled 'flooding'. Each broadcast type datagram has a broadcast stack following the 5 header bytes.

48 - ALRTON	Alert status ON
49 - ALRTOF	Alert status OFF
50 = ROUTLD	Delete a node from the routing table
51 = IMHERE	New node announcing it's presence to the entire network

Network circuits

A network circuit is a communication path between two users via a network. In TEXNET, only the two terminating nodes have knowledge of the user circuit. These two terminating nodes communicate with one another to set-up, maintain, and tear-down the circuit.

Circuit set-up/teardown

Control fields 16-21 are the commands necessary to establish and delete user connections through the network. Each terminating node maintains several flags about the state of each user logical channel. These flags, and the command fields regulate the connection. The flags are:

1 - user is in command mode
2 - incoming connection request
3 - outgoing connection request
4 - circuit is in information transfer mode
5 - local logical channel is busy
6 - remote logical channel is busy
7 - remote user is an authorized administrator

The terminating nodes exchange the fields 16-21 to control both set-up/teardown, and end-to-end flow control. Flow control is activated between user logical channels at the respective terminating nodes. AX.25 busy is entered when either local or remote busy is encountered on a circuit. This prevents over-filling the network frame buffers within each network node when the sending user is transmitting faster than the receiving user can accomodate. It does not flow-off other users at those nodes.

Connect request data field

The connect request data field contains the callsign of the requesting user, the callsign of the desired target user, and the digipeater string (if any) to be utilized at the remote network node.

Packet fragmentation

Since the maximum packet length is 128 bytes, and the network header takes 5 bytes, if the incoming user data packet is between 123-128 bytes in length, it is fragmented into two packets. The packet IS NOT re-constructed at the terminating node, but rather is sent as two packets to the remote user.

Network initialization

TEXNET is a network with no fixed routing assignments. This current version is a significant improvement over the previous version where the entire static network map was stored within the EPROM at every node. This necessitated changing every EPROM in the network whenever the topology changed. The automatic routing system developed is described below.

The purpose of the automatic routing system is four-fold::

1. Notify all possible neighbors whenever a new node is turned on - even if we do not know who those neighbors are.

2. Initiate AX.25 trunk connections between all neighbors, and associate each logical channel with each neighbor.

3. The new node must 'announce' it's presence to the entire network.

4. Routing table entries between the new node and every other node in the network must be generated. Furthermore, the optimum path must be selected for use as the primary route (where two or more routes exist), and an alternate route must be selected, to be utilized if the primary route fails, assuming that there is redundancy in the paths. Further, it is possible the paths on several different physical channels (different radios) may exist simultaneously, therefore priority must be resolved.

UI - frames

The UI-frames are used by a new node (one that has just completed a processor reset sequence) to alert it's presence to all nodes within hearing range. These frames are an 'invitation' to a neighbor node to start a network trunk back to this new node. The UI-frame is sent 'retry' number of times, to assure that all other network nodes within hearing distance will have a good chance of receiving it. This is transmitted on all physical channels unless the node database has a particular physical channel disabled from network trunk initialization.

The format of this UI-frame is defined as follows:

PID - 0C3 Hex
1st byte - Our network number
2nd - Our node number
3-n bytes - Our neighbor exclusion table

'Our network number' is a l-byte 'network identifier'. This allows two (or more) different TEXNET networks to co-exist on the same frequency without logical interconnection. This may be useful for network testing. Additionally, this can be very useful if it is desired to interconnect two TEXNET networks with a gateway station. The gateway can communicate to both networks on the **same** frequency, but the two networks will not logically interconnect. This would be

useful, for example, in constructing a single satellite gateway to two different geographically adjacent networks. 'Our node number' is our assigned l-byte number, and the neighbor exclusion table is a listing of those other nodes which we disallow direct network trunk connections to. This is useful for locking certain paths out (restricting the network topology) during network testing, or eliminating known marginal paths from trunk initialization. Receipt of this UI-frame will cause the receiving node to 'start' (SABM) an AX.25 circuit back to the sender of this UI-frame.

Node identification command

When a connection trunk start is received, the receiving node sends the NODEID network command. This causes the receiving circuit to record this nodenumber into the logical channel table so that association between a logical channel and a particular neighbor node number is made.

Additionally, the NODEID contains the real-time clock and date from the node that received the UI-frame. This automatically keeps new (and restarted) network nodes operating at the correct time and date.

Routing path initialization

After the network trunks have been started between nodes, the node must announce it's presence to the network. It must convey to the network how it is connected to it's neighbors (what it's connectivity to the network is), it must convey it's ASCII node name, and it must convey it's one-byte node number.

The new node will now send the 'IMHERE' I-frame to all of it's neighbors that have started an AX.25 connection to it. Since this is a broacast type of packet, it will be sent to the entire network. Each node in the network will return the 'MYNAME' packet back to this node.

The 'IMHERE' packet data field (in fact, all broadcast packets) contains the following information:

0-4th byte: Normal 5-byte TEXNET header

5th byte: RTHOPS - route hop counter (incremented at each node while building the routing table)

6-20th bytes: RTSKDP - 'push-down stack' holding the previous transit node #s

21-27 byte: RTNONA - ASCII name of the subject node

The hop-counter is incremented by each node that re-broadcasts the packet. Each node also pushes its own node number onto the stack. If the stack contains our node number, then we discard the broadcast packet, since we've already seen it. If the broadcast packet does not contain our node number, we add it to the stack, increment the hop counter, and send this packet to all of our neighbors. This is how infinite routing loops are prevented in TEXNET.

Network response to initiating node

Each node that receives the broadcast 'IMHERE' packet adds the station to it's routing table, if it is not already there. If this station is already there, then the routing table entry is replaced with the new route only if the hop count is less than the current hop count in the table.

At this point, the node adds a physical channel 'hurdle' to the hop count. Routing received on a high-speed network trunk is preferentially treated. That is, if our node has both 9600-baud and 1200-baud trunks, and it receives routing information on each, the 1200-baud trunk will have an additional number of hops arbitrarily added to the hop count before the entry is stored in the routing table. This gives preference to the 9600-baud path, but still allows alternate routing via another physical channel. Also added to the routing table are the ASCII nodename, and it's nodenumber, and the logical and physical channel number that the route was received on (which becomes the 'return route' to that remote node).

The actual algorithm is somewhat more sophisticated. It tests for the case of a previous network node restarting, and prevents duplicate primary and secondary entries under these and other conditions. Receipt of an improved route causes it to become the new primary route, and the old primary to become the new secondary route.

Routing

Each routing table entry contains the primary and secondary (if secondary exists) routes to a particular node. This table indexes the desired destination of the packet to a particular logical and physical channel that leads to that destination. When a packet is received, the field 'REMOTE NODE NUMBER' is examined. If this field is not equal to our node number, then the packet must be transit routed toward the destination. The routing table supplies the information on which physical and logical channel the packet should be sent. No data is altered within the packet.

If however, RNN = our node number, then the packet is for our node. We do not forward the packet, but rather parse the control field, and act appropriately.

Network administration

A number of network commands are used for administration (such as the IMHERE and MYNAME types). These packets are exchanged between nodes, and are independent datagrams between nodes. They are defined as either command with required response, command without required response, or as response, as follows:

TYPE	COMMAND W/RESP	COMMAND WO/RESP	RESPONSE
0 - NODEIN		X	
1 - STATRQ	X		
2 - STATRS			X
3 - NODEID		X	
4 - POINTQ	X		
5 - POINTR			X
6 - YSTATR	X		
16 - CONREQ	X		
17 - CONACK			X
18 - DISREQ		X	
19 - FLOWON		X	
20 - FLOWOF		X	
21 - USERIN	this is user information		
32 - MYNAME			X
33 - TIMEUP		X	
48 - ALRTON		X	
49 - ALRTOF		X	
50 - ROUTDL		X	
51 - IMHERE	X		

Identification

Periodically, the network will transmit a UI-frame with the normal layer 2 (PID = FO) PID field. This packet consists of the layer 2 amateur callsign of the node in the SENDER field, the text 'ID' in the RECEIVER field, and the ASCII name of the node in the data field. This identifies the node per FCC requirements.

Response fields

A simple algorithm is used to determine how to construct the response datagram to any command. The Network Header block bytes are reversed. This causes the local and remote node numbers to change place, and the local and remote logical channel numbers to change place, then the appropriate control code is inserted. In this way, the response is returned to the correct requestor.

Response field data

Each response field may have response data associated with it. The following table lists the response data.

TYPE DATA FIELD CONTENTS

2 - STATRS Compressed 21-byte binary statistics block. This block is expanded into English by the terminating node when delivered to the user.

5 - POINTR Binary block containing the status value of the input-point word (up to 13 input points may be monitored).

Error recovers

In case an error is detected by an intermediate or terminating node, that node performs a similar network header block reversal, but then overwrites it's node number into the local field, and 0 into the logical channel field, and adds the appropriate Network Information Code (NIC) - the error code. In this way the user is informed about the error, and the name of the intermediate node that detected the error. Control fields 128-255 are reserved for the various NIC codes.

Isolation of errors to a particular node is very useful in fault sectionalization. That is, identifying either a particular defective node, or a particular marginal trunk path.

Transport layer (layer 4)

No transport layer protocol is included within the terminating nodes, or within the TEXNET definition. Consider the following scenario: User 1 is sucessfully transfering data through the network to user 2. Channel congestion on the user 1 to TEXNET frequency (the 2-meter user input channel) causes the user 1 to TEXNET link to reset (SABM), possibly resulting in the loss of data frames. A transport function entirely within the network could not correct such loss of frames, and could not correct a complete loss of connection to the network. The connection would either proceed with data errors, or would terminate abnormally.

Instead, the IS0 reference model defines that the transport layer is conducted between the end-users. In this way loss of data or of connection anywhere within the communication path is detected and corrected by the layer 4 (see, for example CCITT X.224 class 1). The IS0 reference model assures that end-to-end data integrity is maintained.

Practical experience with TEXNET has shown that we have not had a need for a transport-like function inside the network to improve the quality of network service. All known instances of data errors have occured between the individual users and the network, not within the network. We have instead devoted limited EPROM space to additional features, such as the network bulletin board, the conference bridge, and the weather-service interface, which are heavily used by our community.

Author's opinion

The definition and adoption of an inter-network, or user-to-network standard (layer 3)within the amateur community is of critical importance. The definition of the intra-network protocol (such as this document) is of secondary concern. It should not matter to any user whether the underlying network is TEXNET or any other network, the interface should be the same regardless. Future developments, such as transport layers, and more, are critically. dependent upon this standard. Ideally this standard would encompass networks, bulletin boards, and future services.

Use of tranport-like functions within the network can improve the quality of service delivered by the network if needed, but do not assure end-to-end integrity. Reference to transport, or layer 4 should be restricted to a protocol which is end-user to end-user in nature.