The AX.25 Amateur Packet Radio Link-Layer Protocol

Unnumbered and Supervisory Frames

Flag	Address	Control	FCS	Flag
8 Bits	112 - 560 Bits	8 Bits	16 Bits	8 Bits
01111110	Callsigns & SSIDs of Destination, Source and Optionally, Digipeaters	Frame Type	Calculated Value	01111110

Unnumbered Frames

There are six types of unnumbered frames:

SABM Set Asynchronous Balanced Mode --Initiates a connection between two packet-radio stations.

DISC Disconnect -- Terminates a connection between two Packet-Radio stations.

UA Unnumbered Acknowledge --Receives and accepts the SABM and DISC frames described above.

DM Disconnected Mode -- If the packet station is busy and unable to accept a connection at the moment, it rejects the SABM frame by transmitting the DM frame.

FRMR Frame Reject -- Indicates that the source station is unable to process a frame and that the error is such that resending the frame will not correct the problem. This frame is rarely used.

UI Unnumbered Information -- Allows data to be sent from a source station without a connection to the destination station.

Supervisory Frames

Supervisory frames are used to control the communications link:

RNR Receive Not Ready --Indicates that the destination station is not able to accept any more Information frames because of a temporary "busy" condition.

RR Receive Ready -- Indicates that the destination station is able to accept more Information frames.

REJ Reject -- Used by the destination station to request a retransmission when an out-of-sequence frame is received.

Information Frames

Flag	Address	Control	PID	Information	FCS	Flag
8 Bits	112 -560 Bits	8 Bits	8 Bits	N X 8 Bits	16 Bits	8 Bits
01111110	Callsigns & SSIDs of Destination, Source and Optionally, Digipeaters	Frame Type	Layer 3 Prot. Type	User Data	Calculated Value	01111110

Flag_Field

Since amateur packet radio is a bit-oriented protocol, the only way to tell when one frame is over and another is starting for sure is to delimit each frame with a certain bit sequence both at the beginning and the end. This is the job of the flag field. A flag consists of a zero followed by six ones followed by another zero, or 01111110 (7E hex). Due to the bit stuffing mentioned above, the only time this sequence is allowed is at the beginning and end of a legitimate frame.

Address_Field

The address field is used to identify both where the frame came from and what the destination of it is. In the CCITT recommendation X.25, this field is only one octet long. This permits at most 256 users per level 2 channel, and since some bits of this field were used for other purposes, the real number of users were about thirty per level 2 channel. Both the HDLC and ADCCP recommendations allowed the address field to be extended, so we decided to extend the address field per their recommendations in the amateur version of X.25 to include the callsigns of both the destination and source amateur radio stations. The method used to extend the address field will be described shortly.

Control_Field

The control field is used to identify the type of frame and control several attributes of the level 2 connection. It is one octet in length, and its encoding will be discussed in a following section.

PID-Field

The Protocol Identifier (PID) field is used only in information frames, and identifies what kind of layer 3 protocol, if any, is in use. Its encoding is as follows:

```
M      L
S      S
B      B
xx00xxxx  Reserved at the moment.
xx01yyyy  AX.25 layer 3 implemented.
xx10yyyy  AX.25 layer 3 implemented.
11110000  No layer 3 implemented.
11111111  Escape character.  Next byte contains more PID information.
```

Where:

1. An x indicates a "don't care" bit.
2. A y indicates all combinations used.

Information_Field

The information field is used to convey the actual user data from one end of the link to the other. I fields are allowed in only three tyes of frames, the I frame, the UI frame, and the FRMR frame. The I field can be up to 256 octets long, and should be an even multiple of octets long. Any information in the I field should be passed along the link totally transparently, except for any zero-bit insertion necessary to prevent flags from accidentally appearing in the I field.

Frame_Check_Sequence

The frame-check sequence is a sixteen-bit number calculated by both the sender and receiver of a frame. It is used to make sure that the frame was not corrupted by the medium used to get the frame from the sender to the receiver. It is calculated in accordance with ISO 3309 (HDLC) recommendations.

Bit_Stuffing

In order to assure that the flag sequence mentioned above doesn't accidentially appear anywhere else in a frame, as the frame is being sent it should be monitored, and if more than five contiguous ones are detected, a zero bit should be added between the fifth and sixth ones, eliminating the possibility of a flag appearing in the frame other than where it belongs. The receiver of five ones, a zero, and more ones should automatically eliminate the inserted zero before passing the data on.

The KISS TNC: A simple Host-to-TNC communications protocol

Mike Chepponis, K3MC
Phil Karn, KA9Q

Presented at the ARRL 6th Computer Networking Conference, Redondo Beach CA, 1987.
Translated to HTML by KA9Q, January 1997.

ABSTRACT

The KISS ("Keep It Simple, Stupid") TNC provides direct computer to TNC communication using a simple protocol described here. Many TNCs now implement it, including the TAPR TNC-1 and TNC-2 (and their clones), the venerable VADCG TNC, the AEA PK-232/PK-87 and all TNCs in the Kantronics line. KISS has quickly become the protocol of choice for TCP/IP operation and multi-connect BBS software.

1. Introduction

Standard TNC software was written with human users in mind; unfortunately, commands and responses well suited for human use are ill-adapted for host computer use, and vice versa. This is especially true for multi-user servers such as bulletin boards which must multiplex data from several network connections across a single host/TNC link. In addition, experimentation with new link level protocols is greatly hampered because there may very well be no way at all to generate or receive frames in the desired format without reprogramming the TNC.

The KISS TNC solves these problems by eliminating as much as possible from the TNC software, giving the attached host complete control over and access to the contents of the HDLC frames transmitted and received over the air. This is central to the KISS philosophy: the host software should have control over all TNC functions at the lowest possible level.

The AX.25 protocol is removed entirely from the TNC, as are all command interpreters and the like. The TNC simply converts between synchronous HDLC, spoken on the full- or half-duplex radio channel, and a special asynchronous, full duplex frame format spoken on the host/TNC link. Every frame received on the HDLC link is passed intact to the host once it has been translated to the asynchronous format; likewise, asynchronous frames from the host are transmitted on the radio channel once they have been converted to HDLC format.

Of course, this means that the bulk of AX.25 (or another protocol) must now be implemented on the host system. This is acceptable, however, considering the greatly increased flexibility and reduced overall complexity that comes from allowing the protocol to reside on the same machine with the applications to which it is closely coupled.

It should be stressed that the KISS TNC was intended only as a stopgap. Ideally, host computers would have HDLC interfaces of their own, making separate TNCs unnecessary.[15] Unfortunately, HDLC interfaces are rare, although they are starting to appear for the IBM PC. The KISS TNC therefore becomes the "next best thing" to a real HDLC interface, since the host computer only needs an ordinary asynchronous interface.