Terry Fox, WB4JFI
Vice-President, AMRAD
7877 Hampton Village Pass
Annandale, VA 22003

Purpose

This paper is presented to document some of the major changes proposed over the last four years of operation for the AX.25 Level 2 Version 2 Protocol. These changes have been collected by this author from various sources, and were recommended by a working group of the ARRL Digital Committee which met in July of 1988.

Background

The Amateur Radio Link Layer packet standard known as AX.25 Level 2 has come along way since it's creation during the summer of 1982. At that time, six Amateurs discuss a replacement for the original Vancouver Protocol created by Douglas Lockhart, VE7APU. Since this author had already written several of the proposed ideas in the AMRAD Newsletter, a draft of the new AX.25 Level 2 was begun.

This draft was almost completed when Tom Clark, then President of AMSAT, called a meeting in October of 1982 to establish a standard Link Laver Protocol to be used on AMSAT Satellites. At that meeting, several protocols were discussed, with the result being that a slightly modified version of the AX.25 Level 2 Draft being adopted. The major alteration was to include fields for more than one digipeater. This draft was then used by TNC-1 Terminal Node Controllers. The AX.25 draft quickly became THE standard for packet radio due to it's implementation by TAPR. It should be noted that the TNC-1 was capable of both AX.25 and the original Vancouver Protocol. The AX.25 Level 2 Protocol Specification was published in March of 1983 in the Second ARRL Amateur Radio Computer Networking Conference Proceedings by this author.

The AX.25 Level 2 protocol was heavily based on the commercial X.25 Protocol Specification, with some revisions. One of these revisions was the removal of certain types of frames (S-Frames) as commands for link status and maintenance. Instead, Information Frames were used, along with a heavier emphasis on timers. This was done to simplify the protocol implementation, but as it turned out, this short-cut caused more problems than it relieved.

It became apparent that some of the changes, such as the removal of command S-Frames, were not working as well as had been expected. About the same time the ARRL and it's Digital Committee expressed a interest to officially adopt the AX.25 Level 2 protocol. These factors led to the creation of a new version of the protocol, which more closely adhered to both the CCITT X.25 and AT&T BX.25 standards. The original specification was expanded and refined to become AX.25 Level 2 Version 2.0, which was adopted by the ARRL Digital Committee and the ARRL Board in October of 1984. The original AX.25 was labeled Version 1.0 to indicate it's operational differences.

Version 2.0 of AX.25 Level 2 has been in use for four years now, with estimates of between 50 and 80 thousand devices using it. During this time, some additional problems with it have been encountered. The remainder of this paper discusses alterations to the AX.25 Level 2 Version 2.0 Protocol Specification.

Backward Compatibility Issue

As just mentioned, there are an estimated 50 to 80 THOUSAND devices using the AX.25 Level 2 Version 2.0 protocol. A fundamental issue that must be resolved is what happens to these users when alterations to the protocol are made. In Amateur Radio, there is no method of requiring this installed user base to migrate to a new protocol quickly. If changes are made that cause incompatibility between users, complaints are sure to follow. An example of this is found with the TAPR TNC-1. Due to an oversight in the software design, TNC-1s cannot be used with Version 2.0, even as digipeaters. This is because the TNC-1 code accidentally tests bits that were clearly marked as reserved in the protocol spec, which were later defined in Version 2.0. To this day, people complain about this, even though it is an implementation problem, NOT a protocol issue.

Some of the more agressive protocol changers do not believe that backward compatibility to previous versions should be an issue. They feel changes for the. better should be made automatically without regard to on-air compatibility. In fact, they argue, if the newer version causes enough on-air problems and crashes, the rest of the community must follow just to stay on the air. This author vigorously disagrees with that method of subterfuge. Another potential result of this scheme is the turning-off of both network resources and Amateurs themselves.

Due to the large number of devices using Version 2.0, this author feels that backwards-compatible solutions should be implemented whenever possible. This guideline should be first and foremost. If a solution cannot be reached which is backward-compatible, it should be flagged and carefully weighed before implementation, since that implementation would immediately cause at a minimum incompatibility, and possibly complete and systematic crashes.

It should be noted that the above relates primarily to communications channels where large numbers of Version 2.0 users are found, and is NOT meant to preclude either experimentation or modifications to enhance specialized channels, such as high-speed network backbones. Since there will be a limited number of users on such a channel, they should be allowed to run any agreed-upon changes or enhancements which are within the governing body's rules and regulations.

In the following discussions, two distinct types of modifications will be discussed. Those that would be backwards-compatible to Version 2.0 could be considered for either a Version 2.1 release or a Version 3.0 release, while modifications that would NOT be backwards-compatible should be considered for a Version 3.0 release ONLY. This numbering scheme follows this author's recommendation that minor revisions are noted by the number after the period, while major alterations are noted by the number preceeding the period.

Addressing Issues

One of the prime motivating forces behind the move to update AX.25 Level 2 is its limitations in the area of addressing. When this author originally proposed the use of an extended address field that included Amateur callsigns, it was felt that six characters would be of sufficient length to contain the callsign, with a seventh tacked on to allow each Amateur to maintain more than one station. It now appears that both prefixes and suffixes should be sent in addition to the callsign. These additions often cannot fit within the six octets allowed under Version 2.0.

Further, since Version 2.0 specifically calls for sub-fields of six characters plus SSID, extending the individual callsign address sub-fields is precluded.

After much individual research and the suggestions of many others, two methods of allowing adding the additional information have been recommended. The first method has been designed to be backward-compatible, but is rather inelegant. The second method is designed to cure regulatory-related problems outside the United States in addition to the above issue, but is NOT backward-compatible.

Backward-Compatible Addressing Extension

The first method of adding more addressing information to AX.25 Level 2 Version 2.0 is admittedly very much a kludge. It will not be immediately obvious to older equipment exactly what is going on, but it will function properly with the older versions, assuming they followed the previous protocol standard, unlike the TNC-1 as described above. This unwitting inter-operation with the older standard makes it backward-compatible with them.

Simply put, method one defines additional address sub-fields called extension sub-fields which, if present, convey the additional addressing information required. They are placed after the destination and source addresses, but before any repeater address sub-fields. As with the other address sub-fields, these extension sub-fields must be seven bytes long. Both the placement and encoding of the SSID in the extension sub-fields are a subterfuge to imply to an older version device that the extensions are digital repeater addresses, allowing the older version to ignore the extensions presence. Since Version 2.0 allows at most eight digital repeaters, any extension sub-fields must be subtracted from the number of allowed digital repeaters to keep the maximum number of "repeater" sub-fields at eight or less.

Address Extension Information Encoding

The additional information is encoded in the same manner as the other address information. It should be bit-shifted ASCII, stuffed with trailing ASCII spaces as required to six characters plus SSID. The only other requirement is that if the additional information is a prefix to a callsign, the slash (/) character is placed after the prefix. If the additional information is at the end of the callsign, the slash is placed before the postfix. For example:

Amateur Callsign	Normal Address	Extension Field 1	Extension Field 2
WB4JFI/KT-1	WB4JFI-1	/KT	(none)
VP2M/WB4JFI-1	WB4JFI-1	VP2M/	(none)
VP2M/WB4JFI/KT-1	WB4JFI-1	VP2M/	/KT

As implied above, more than one extension may be required, and may be used. If both pre- and postfixes are required and are under six bytes in total length (included a shared slash), they may be combined in a single extension field.'

The SSID number of the first address extension sub-field for each address (callsign) will be set to zero, the second to one, and so on as necessary. Not only will this aid in "glueing" the address back together, but will also indicate when one extension block ends and another begins.

The equivalent of the H-bit (bit 7) in the SSID octet of all extension sub-fields should be set to one at all times by a Version 2.1 device, indicating to a previous version device that this is a repeater field that has been repeated. This will allow a previous version device which is a destination to conclude that all repeaters (including the extensions) have repeated the received frame, and it may process the frame.

Extension Information Indication

Any address that has extension information will indicate this by resetting bit 6 (hereafter called the A-bit, for Address extension) of its SSID octet to zero (0). This bit has previously been reserved and should have been set to one (1) as indicated by both Versions 1.0 and 2.0 of the AX.25 protocol specification.

The extension sub-address fields should also have the A-bit set to zero to simplify the comparison of extension sub-fields with repeater sub-fields in Version 2.1 devices.

Extension Information Placement

In order to fake-out earlier versions of the protocol, the extension information cannot simply follow its base address. The only place this new information can be placed which will work with older versions is between the source address and the first of any repeater addresses. If placed there with the H-bit set to one, older versions will assume the field is for a repeater that has already repeated the frame.

The order of appearance of these extension sub-fields is the same as the main address sub-fields; any destination extensions come first, followed by any source extensions, then any repeater extensions in the same order as the repeater sub-fields themselves.

Examples of Address Extension

The following example will aid in indicating the proper operation of the proposed address extension recommendation.

The following example indicates how a single address sub-field may be extended. In it, the destination field has a post-fix modifier. The frame is a UI command frame from WB4JFI-1 to K8MMO/KT-0. Note how the A-bit is set to zero both in the destination sub-field AND the address extension sub-field. Note also how the address extension sub-field has an SSID of 0000 indicating it is the first (and in this case only) extension sub-field.

```
!   DA    !   SA    !  Ext. 1 ! CTL !
! K8MMO 0 ! WB4JFI1 ! /KT   0 ! UI  !
```

The actual bit-encoding of these fields is as follows:

	A1	K	1001 0110	96
D	A2	8	0111 0000	70
e	A3	M	1001 1010	9A
s	A4	M	1001 1010	9A
t	A5	0	1001 1110	9E
.	A6	SP	0100 0000	40
	A7	SSID	1010 0000	E0
	--	--	CARS SID-	--
S	A8	W	1010 1110	AE
o	A9	B	1000 0100	84
u	A10	4	0110 1000	68
r	A11	J	1001 0100	94
c	A12	F	1000 1100	8C
e	A13	I	1001 0010	92
	A14	SSID	0110 0010	E2
-	--	--	CARS SID-	--
E	A15	/	0101 1110	5E
x	A16	K	1001 0110	96
t	A17	T	1010 1000	A8
e	A18	SP	0100 0000	40
n	A19	SP	0100 0000	40
.	A20	SP	0100 0000	40
	A21	SSID	1010 0001	A1
	--	--	---- ----	--
	CTL	UI	0000 0011	03

If there are real repeater fields in addition to the extension information, the address field will look as follows:

Uplink to the Destination Station

```
!   DA    !   SA    !  Ext. 1 !  RPTR 1 ! RPTR 2!
! K8MMO 0 ! WB4JFI1 ! /KT   0 ! WB3KDU5 ! WB4JFI5!
```

Link Back to the Original Station

```
!   DA    !   SA    ! RPTR 2  ! RPTR 1  ! Ext. 1 !
! WB4JFI1 ! K8MMO 0 ! WB4JFI5 ! WB3KDU5 ! /KT   0!
```

If the link-back frame is passed through older version digipeaters, the H-bit in the Extension sub-field and A-bits in both fields may not be set to indicate an extension is present. Fortunately, since the newer device is now **receiving** the frame it knows there may be extension information which may look like repeater sub-fields. It can compare the frame's repeater addresses (or count the number of H-bits set compared to repeater fields used), to find out if all repeaters addresses have been used.

Address **Extension** Operation

If an older-version device starts the link, it will not create these extension **sub-fields**. Therefore, frames it generates, and those frames that are passed through repeaters will not have problems. If a new-version device is at the destination, it may respond with a **frame** that does have extension information. **The** extension sub-field will have the H-bit set, so the first station (and any repeaters) will **assume** it is just a has-been-repeated field. **If the** first station does not modify its path table entry, both stations will operate as first started, with the extension information being carried as "excess baggage". If the first station does **modify** its Path table to include the extension information as repeaters, the extensions will be placed at the end of the address field, allowing-any actual repeaters to repeat the frame properly. Once the frame has finished being repeated, just the extension information is left with the H-bit not set. The second station can evaluate the received frame and detect that all repeaters are done.

If the first station originating the connection is a newer protocol device attempting a connection to an older protocol device, operation is similar to the above. The extension information is placed between the Source and any repeater address sub-fields by the originator. Since the H-bit has been set on the extensions, if a repeater is involved, it will skip the extensions to the first actual repeater address without the H-bit set and test for its address. It will repeat the frame if its address is found, **setting** the H-bit when sending the frame out. This will continue until all repeater fields have been used, and the frame arrives at the destination (hopefully). The older device will reverse the order of repeater sub-fields. and send an acknowledgement **back.** The state **of the** A-bit in all fields is unreliable at this point, making it useless for further **processing. In** addition, the H-bit will be off on all repeater AND extension sub-fields. Since the extension **sub-fields** are now at the end of the address field. any true repeaters in the path will repeat the frame (and set the corresponding H-bit) properly. Once the frame has cleared all-the repeaters and arrived back at the first station, there are still repeaterlike **sub-fields** with the H-bit NOT set. **The new** version device must look at these to-see **if they** are true repeater addresses or **just** extension information.

New Address Framing Technique

The second method is the **subject of a** separate paper found elsewhere **in these proceedings** and will not be discussed "in depth here. It involves separating the addressing issues from the rest of the protocol, and totally redefining the address portion of Level 2 frames.

One of it's advantages is that the Amateur address portion of the frame is totally separate from the X.25 protocol machine, with counters and pointers relating to the address information maintained within the address portion of the frame, rather than implied, which is the **case** with AX.25 Version 2.0. This is meant to simplify software implementation of the protocol.

Another advantage is the addition of a hashed version of the next destination address as the first byte of the address field. This allows the implementation of selective addressing which is built into most synchronous hardware chips, reducing processor overhead.

In addition, Address sub-fields may be of variable length, and may include address mnemonics in addition to Amateur callsigns.

The main disadvantage of this new addressing **technique** is that **it** totally alters the addressing of **AX.25,** rendering it **absolutely** incompatible with ALL earlier versions. **This** removes it from the backwards-compatible **category.** In fact, since the alterations are so drastic, it may cause catastrophic failures of earlier versions just by being heard on a channel, especially to systems whose software is not designed to be fault-tolerant.

Another disadvantage of this new address scheme is that the HDLC address extension bit is NOT implemented, which means it technically violates the HDLC framing standards. The end of the address section is indicated by counters and pointers rather than by the use of the E-bit. The non-use of the E-bit means, among other things, that Protocol Analyzers can no longer be used to troubleshoot and analyze Links and software implementations.

The use of counters and pointers to indicate important places in the address field is meant to simplify software processing of the frame. This simplification is gained at the cost of additional channel overhead. Each of these counters and pointers which must be transmitted ties up the RF channel for that much longer. It is a small point, but does add up eventually.

In addition, there are other **pieces** of information which add to channel overhead, such as the hashed next-destination address field, an address checksum and more **protocol identifiers.** The total of additional information required is **a** minimum of 10 bytes above the AX.25 Version 2.0, Keep in mind that this is 10 bytes more in each transmitted frame.

It is obvious that in the above proposal, concern for **processing** power is higher than concern over channel overhead. Both of these are important issues, so any trade-off between them should be judged **carefully**. Generally speaking, as the channel speed increases, small **additional** overhead becomes less important while processing speed becomes more important. The inverse is true as channel speed is **decreased.**

While both of these address extension systems will convey the information required by the regulating bodies, the second method is not only more flexible, it is also easier to understand and implement. Is this trade-off worth the small additional channel overhead? Are the advantages also worth obsoleting all older versions of the protocol, at least on the channel it will be used on? Will both versions be required to be available simultaneously? Time will tell.

State Description Logic (SDL) Diagrams

In the back of the AX.25 Level 2 Version 2.0 Document are three state tables, which are meant to describe the operation of the protocol based on various external actions taking place. There has been some concern as to whether these tables are actually part of the document or just an implementation guide. They were included to easily indicate to implementers how the protocol should operate, and therefore are part of the protocol description.

One of the problems with these state tables are that they cannot easil indicate the various steps taken when an external event happens. There is only enough room to indicate if a frame is to be sent as a result of the event, and any state transition made as a result of that event. If there is more to be done, the "flatness" of the tables precludes **description** there, requiring the document reader to search the actual text.

There has been another method of describing the actions taken by protocols that is gaining in popularity. This method is called State **Description** Logic, or SDL diagrams. Most of the newer protocol documents developed by the **CCITT** are using SDL diagrams to describe protocols. **An** SDL diagram looks much like a software program flowchart, with slightly different symbols. These SDL diagrams are MUCH easier to read, follow, and implement from. There is an effort to document the AX.25 Level 2 Version 2.0/2.1 in SDL diagrams. If they are available in time, the next printing of the AX.25 Level 2 Document may include the SDL diagrams in place of the State Tables. The main reason for any delay is that the SDL diagrams are quite a bit more complicated, and must be checked very thoroughly with the text to make sure they agree with each other.

State Table Changes

Given the preceeding information, State Table changes may be a moot point. They are included here for completeness.

The main State Table change is the removal of States 14 and 16. After review, no one has ever found how a protocol machine could remain in either of these states for any length of time. Both have to do with the local station being busy, but having sent a REJ frame. The sending of the REJ frame itself indicates a non-busy condition by requesting a re-transmission.

Unumbered Information Operation

There has been some discussion regarding the proper use of the Unnumbered Information, or UI frame. This is especially true when discussing the use of the Poll bit associated with UI frames. Some feel that it is possible to maintain a separate "UI Mini-State-Machine" by the use of Poll and Final bits associated ONLY with UI frames. After careful review, it was deemed that in order for these P/F bits not to interfere with the normal protocol machine P/F operation, more processing overhead would be required than could be justified. Therefore, UI frames are left as Commands, with no Poll bits used.

Automatic Re-connect Elimination

Perhaps the largest complaint heard regarding AX.25 Level 2 performance is when a "disconnected" connection keeps coming back. THere are cases when a station has requested a disconnect, gone into timer recovery due to the disc frame getting lost, and then receives I frames from the other station. Eventually the first station will consider itself disconnected, and send DM frames. The second station still has data pending, so it will re-establish the connection and pass the data. This is NOT a bug, rather it is a deliberate attempt to make AX.25 re-establish failed links. For our shared RF medium, it now appears this tactic is too aggressive. The new recommendation will specify that the link is not re-established, instead an error message is passed to the higher layer protocol or program, which will decide what action the AX.25 Level 2 machine should take.

Maximum Packet Size

There have been many discussions regarding the maximum allowable frame size. Some people feel the 256 byte limit of data in Information frames is too small. They site two reasons for an increase in size. The first is that some Higher-Layer protocols require substantial overhead (such as TCP/IP), and this overhead must be added to the real user data, subtracting from the total transmittable user data per frame. The other reason is that higher speed channels should be able to transfer larger amounts of data per frame, increasing the overhead-to-user-data ratio, thereby making the channel more efficient. Information field sizes of 1024, 2048, or larger have been suggested.

Making an ad-hoc change to the maximum frame size may have serious repercussions, however. Older devices amy have hard limits to the maximum allowable frame size, reducing the chance of backward compatibility. An even greater potential problem is if implementations do not check for an allowable maximum frame size and crash if substantially large frames are received. Older devices have a limited amount of RAM memory, and substantially larger frames might also tie up too much memory, another potential for crashing implementations.

In order to make sure older eqipment doesn't crash due to excessive frame sizes, there needs to be more study done on this issue. Otherwise, drastic results may occur.

The result of discussions within the Digital Committee is an agreement that the 256 byte limit be maintained at this time, with an escape clause that Source, Destination, and Intermediate repeaters may use agreed-upon larger frame sizes on particular link connections and channels. Implementers should be aware of this, and make sure maximum allowable frame size is checked.

Maximum Window Size of One

Phil Karn has been suggesting that present AX.25 connections that send more than a single frame per RF transmission are actually using the channel inefficiently. He feels there should be only one I frame outstanding at a time per direction per link, creating his Ack-Ack protocol. Even if there is some accuracy in his arguement, altering the protocol to make this operation mandatory would be short-sighted. The decision was to leave it alone at this time.

Non-Use of Polling

The original AX.25 Level 2 Version 1 specification did not use Supervisory frames as commands, only responses. Their introduction as commands was the significant alteration that caused Version 2.0. Phil Karn now suggests that we go back to the original system, using I frames for retransmission recovery, but only if the lost I frame was "small". After some discussion regarding "small" vs "large", and implementation requirements, it was decided to keep the Version 2.0 scheme for now. It should also be noted that this system follows the traditional X.25 approach.

Forced Disconnect

Franklin Antonio has raised an issue regarding the Disconnect Request state, S4. Presently, while in this state, if a Local Stop Command is received (from the higher-layer), no action is taken. He suggests a transition to the Disconnected state, S1, and the discontinuation of sending Disc frames. This appears to be reasonable behavior, and was recommended.

Stop Timers During Channel Activity

The main reason for the use of timers during AX.25 Level 2 links is to make sure the link is still valid during slack transmission periods and for error recovery. In half-duplex Amateur Radio use, errors are normally introduced whenever two or more stations transmit at the same time, interfering with each other. The timers presently run whether the channel is busy or not. Some Amateurs argue that whenever the channel is busy, a station cannot transmit anyway, and the busy period should be removed from the delay period. This may also add to the randomization of the delays before station retransmissions.

Stopping the timers appears to have some advantage, but its implementation may cost more in processor overhead than is gained. This issue has been reserved for further study before a decision is made one way or the other.

Ack Prioritizing

Another suggestion that has been made is to make sure acknowledgements have a higher priority than other frames during connections. It is suggested that this will cause fewer retransmissions, since the shorter acks can be clobbered by longer data frames, causing error recovery procedures to be hastily implemented.

After reviewing the various requirements to implement this, especially when repeaters are involved, it was felt that implementing Ack Prioritization could become extremely complicated. More study is needed regarding this subject, as it may still have advantages.

Remote AX.25 Level 2 Parameter Control

There have been a few Amateurs that have indicated a need to be able to remotely access and possibly alter various Level 2 parameters. This has been met with quite a bit of resistance, primarily because of the potential for link damage by others, either accidentally or on purpose. If remote parameter setting is to be performed, it should be done within the confines of a higher protocol, preferrably with some authentication.

Implementation Issues

There have been several questions asked regarding how to properly implement AX.25 Level 2 Version 2. There have also been some "bugs" introduced on the air due to its improper implementation. After discussion of a few of these implementation problems, it was decided to include an Appendix to the AX.25 Level 2 document discussing implementation issues. In addition, the inclusion of the SDL diagrams discussed above should help resolve future questions. A few of the implementation problems are discussed below.

Queued Text When S1 or S4 Entered

Franklin Antonio points out that some implementations (particularly TNC-1 and TNC-2) send any queued text left upon disconnection as Information in UI frames AFTER the disconnection. This should NOT be done. Either the link should be re-established, or the information should be discarded (the decision on which action to take is now left to the upper-layer). This is an implementation error.

RNR and Memory Usage Problems

Some Amateurs indicate that there is a problem with some devices which use RAM memory to store, or log, received data or messages. After a certain amount of operation, these devices can become full of data. If a connection is established while one of these devices is already full, that device will allow the connection, but then indicate its lack of resources by sending RNR frames until the user frees some memory. This should not happen. If a device cannot support a connection, it should indicate that by rejecting the connection request (with a DM response). This is clearly an implementation issue that is beyond the protocol document to describe. The document is meant to describe how the protocol machine operates, not all possible implementation issues, such as mail storage in the same device as the protocol machine.

Bells, Clear Screens, Other Binary Data

A problem that is becoming more an issue every day is that of binary data being passed to the user terminal from the TNC device. Quite often, a user who is monitoring a packet channel suddenly hears bells, the screen clears, and other strange things start happening. This is often because more real network protocols are showing up on the air. These network protocols use binary data in their control fields (which are located in the information field of the AX.25 Level 2 frame), which can cause a terminal to go crazy if it receives the binary data.

When the AX.25 Level 2 protocol was designed, a Protocol IDentifier (PID), was added to indicate what type of higher level protocol if any was being used. At that time, a PID of FO hex was issued to be used whenever no upper layer protocol is being used. Since then, additional PIDs have been assigned to Network Layer protocols. At the time it was hoped that if a device (or software between the user and the device) saw a PID other than FO, that device could selectively not allow the data to the terminal (or computer screen). Since this was not clearly stated as a purpose in the protocol document, it appears that was not implemented. Future software designers should allow for this option, reducing the number of problems showing up on the user screen. This will become even more important as more network protocols show up on the air.

Level 1 and Level 1/2 Interface Issues

Many of the proposed changes received are not directly related to AX.25 Level 2 operations. Specifically, adjustment to L2 timers and channel access operations should be considered outside the scope of the AX.25 Level 2 protocol documentation, since different channel access protocols require different settings and adjustments. There needs to be more work done in this area, possibly as a separate AX.25 Level 1 document. The following is a list of ideas and suggestions that should be considered for a Level 1 document. As an alternative or interim solution, either an addendum to the AX.25 Level 2 document, or a separate working paper could be created to indicate the suggested operations. Another reason for this separation is that most of these adjustments are to fine-tune half-duplex CSMA or CSMA-CD operation. In full-duplex, slotting, or other channel access methods, these parameters do not necessarily come into play, and do not need to be specified or adjusted.

If there is separate document created for these Level 1 issues, wording should be added to the AX.25 Level 2 document indicating which parameters may be altered as a result of another Level's operation.

Persistence

Back in the old days of Vancouver boards running either V1 or AX.25, the software would not automatically and aggressively retry, but would wait before retransmitting. Unfortunately, the TNC-2 software did not implement this, and as a result there has been a beating-of-the-chest regarding schemes to take care of this "sudden" problem. Recently, Phil Karn and others have rediscovered this situation and has suggested we modify the Level 2 specification to include p-persistence.

Adding requirements to the AX.25 Level 2 specification regarding persistence would be a mistake. We will not ALWAYS be using half-duplex CSMA as the channel access protocol.

In our half-duplex environments, p-persistence should be used, with the value of the persistence being set depending on channel operation. The actual value and rate of any alterations made to the persistence are subject to further study.

Retransmission Backoff

Another Level 1 issue has to do with the adjustment of the retransmission timer, T1. Originally, the Vancouver TNC also added some delay each time a retransmission was attempted. Exponential backoff was insisted upon by Phil Karn last year, he has since agreed that exponential backoff may be too aggressive. Tom Moulton has indicated that some have found that on marginal links a simple arithmetic backoff may operate much more efficiently than an exponential backoff. Frnklin Antonio also states some reservation regarding exponential backoff, and mentions arithmetic backoff as an alternative. After some discussion regarding actual implementation of retransmission backoff, most agreed that second-order polynomial backoff might be a good compromise. This may still be subject to further experimentation.

T1 regarding Round-Trip Timing

Yet another modification to the T1 timer Phil has proposed is to have it's value adjusted based on an average of the round-trip time for information packets sent and acks received. Once again, the old Vancouver AX.25 code had a variation on this, in that it automatically altered the value of T1, based partially on the number of digipeaters used by the link. Phil suggests that a continual monitoring of round-trip timing be used, and a smoothed version of this value be used to adjust T1. Retries should not be included in adjusting T1, as that may throw off the actual round-trip timing.

Carrier Sensing

Franklin Antonio, N6NKF, has suggested that RF carrier sensing be used in half-duplex operation. He points out that while most all AX.25 software presently implements this for half-duplex operation, it is not actually written in any specification. This is yet another item that belongs in another Protocol document, related to channel access methods. This is especially true since carrier sensing really applies primarily to CSMA channels. Full-duplex, slotting, and Aloha may not need to do this sensing.

Keyup Delay

Most TNCs have a variable delay between when the RF transmitter is first turned on, and when data actually starts being sent, to allow for the transmitter to stabilize. This keyup delay is often called TXDELAY, and Franklin suggests that it be required to be user-settable, and he also suggests that if possible it be implemented (at least partially) in hardware, due to the wide variation of values based on packet speed. While its specification is needed, Keyup Delay is another issue that belongs in a Level 1 document.

When to Stop Transmitting

Another timer issue Franklin has brought up is a transmit turn-off delay (TX-Tail). He points out that some packet hardware can have many bytes of buffering, and although the host CPU may think a packet is done, the actual data may not have cleared the hardware yet. If the CPU then turns off the transmitter, the last packet will not have been completed, and the whole packet is therefore corrupted. He further points out that present implementations do not adjust this timing, but rather make it arbitrarily long to be sure the whole packet clears, regardless of channel speed. He suggests this timer also be user-settable.

Others indicate different methods can be used to insure the end of the packet has actually been sent. If a particular chip has a three-character buffer, the start of another frame plus three characters can be sent to the chip, then an abort command is issued to the chip, which will abort the short "timer frame", yet allow enough time for the last frame to be completely sent. This removes the necessity of having yet another special user-settable value which the user will not know how to set most likely. Implementers should take note of this trick in order to reduce channel overhead.

Additional AX.25 Level 2 Issues

In addition to the previous items, there are a few more that have come up whenever alterations to the AX.25 Level 2 protocol is discussed. These have not been thoroughly digested yet, and may be subjects for further enhancements at a later time. Some of them are listed below.

Parameter Negotiation

Several methods of negotiating different AX.25 operating parameters at connection setup have been discussed. The parameters most frequently talked about are packet data, and window size. Alteration of the default values have been suggested either by adding an information field to the SABM frame, or by using the XID frame (not presently allowed in X.25).

Either method violates the X.25 protocol standard, which does not seem to allow for such an option. Using I fields in connection requests (SABM frames) was done under the original Vancouver protocol. This seems to violate some basic frame rules, including Unnumbered and Supervisory frames not containing data (except for FRMR frames). The general feeling seems to be to not use this method.

Using XID frames to transfer special requests also has problems. Do the XID frames come before or after the connection is established? It has been suggested that an XID command frame with the Poll bit set be used to convey a special request, and an XID response frame with the Final bit set be used to indicate the response to the request. This seems more managable, and should be researched further.

Larger Window Sizes

When the Amateur community starts using satellites and high speed links for backbones, it may be beneficial to use larger frames and more of them. Our present limit of window size is seven, due to the three bit frame numbering plan implemented. Commercial satellite users have found that this limitation is too small when round-trip transmissions are in the order of a quarter-second for a geosynchronous satellite. The use of larger windows, by extending the numbering plan to seven bits, has allowed more effective use of satellites.

Extending the frame numbering to seven bits would allow up to 127 frames outstanding at a time. This would require a second control field byte. Use of the extended numbering should be considered in future AX.25 releases.

Selective Reject

While not allowed under X.25, other HDLC Level 2 protocols do allow the use of selective reject frames. These are used when a station receives a multiple frame transmission, with a bad one in the middle. The selective reject frame indicates not only that a bad frame was received, but gives the number of that frame, indicating it only needs that frame to complete the group. While the use of selective reject is not particularly time-saving with smaller windows, if the above extended windowing suggestion is implemented, some form of selective reject should also be implemented.

There is a possibility that many frames are missed, with selective rejects issued for each of those frames. This could actually perform slower than if all outstanding frames were retransmitted. Actual implementation should take this into account, relying on some mean value to determine which action to take.

Multi-Reject

Presently, once a Reject frame has been sent, another cannot be issued until the error condition has been cleared, Suggestions have been made that multiple rejects be allowed to be sent. simplifying error recovery. The exact operation of this with older devices is in question, and will also be subject to further study.

Data Compression

Franklin Antonio has also suggested that standard data compression techniques be implemented over AX.2.5 Level 2 links, to shorten transmission times. While this is a good idea, and should be further researched, it probably doesn't belong in the actual protocol specification.

Assignment of PIDS Based on Manufacturer

GLB Electronics requested a couple of years ago that certain Protocol IDentifiers be reserved and assigned to implementers. After much discussion over several meetings, this issue is still unresolved. The leaning of many at the moment is that this could be dangerous, potentially triggering user wars between system types. Many remain unconvinced that the benefits would outweigh the potential for harm.

Conclusion

This paper outlines most of the issues brought up regarding AX.25 Level 2 Version 2.0 since its adoption in October, 1984. There may be some additional ones that have been missed, which will be picked up at future and Conferences.

There have also been several suggestions and corrections to the text of the AX.25 Level 2 Version 2.0 document, which were left out for the sake of brevity. Most of these corrections have been indicated to the ARRL Digital Committee.

The next step is for this author to add the suggested changes to the AX.25 document and distribute the changed version to the Digital Committee, where it will be held under further review. At that time the SDL diagrams will also be added to the document.

After passing that review step, the Digital Committee will approve a final new version of the protocol, then have it printed and distributed.

Those with any comments, suggestions, or complaints should send them to this author at the above address. They will be passed to the Digital Committee in addition to being placed in the permanent AX.25 Documentation file kept by the author,which is the main basis for further AX.25 Level 2 modifications.

References

Lockhart, D. "The Vancouver Amateur Digital Communications Group", Protocol Column, AMRAD Newsletter, AMRAD, June 1980

Fox, T. "AX.25 Amateur Packet-Radio Link-Layer Protocol Version 2.0 October 1984", ARRL, 1984

Fox, T. "Notes of East Coast Packet Radio Standards Meeting", Unpublished, January 23, 1982

Fox, T. "Level 2 Protocol Proposal", AMRAD Newsletter, March 1982

Fox, T. "Protocol Specification for Level 2 (link level) Version 1.0", Unpublished Draft, July 4, 1982

Fox, T. "AX.25 Level 2 Protocol", Second ARRL Amateur Radio Coromputer Networking Conference, ARRL, 1983

Karn, P."Proposed Changes to AX.25 Level 2", Correspondence, June 1987

Karn, P. and Lloyd, B. "Link Level Protocols Revisited", Fifth ARRL Amateur Radio Computer Networking Conference, ARRL, 1986

Antonito, F. "AX.25 Proposed Changes", Correspondence, 1988

G3VPF "Suggested AX.25 Specification Changes", The South-West AX.25 Group, 1988

Gustafson, E. "Proposals for Updating AX.25 Level 2 Protocol Specification", Correspondence, 1988

Anderson, P. "AX25L2V2", Correspondence, 1987

Johnson, L. "Thoughts on AX.25 Level 2 Version 2.1", Correspondence, 1988

CCITT "Recommendation X.25", CCITT, 1976, Ammended 1976, 1980, 1984

IS0 "Data Communications - Description of the 1984 X.25 LAPB - Compatible DTE Data Link Procedures", IS0 Document 3179, ISO, 1984

Bell System "Operations Systems Network Communications Protocol Specification BX.25 Issue 2", American Telephone and Telegraph Company, 1980

Bell System "Operations Systems Network Communications Protocol Specification BX.25 Issue 3", American Telephone and Telegraph Company, 1983

IS0 "Data Communications - High Level Data Link Control Procedures - Frame Structure", IS0 Document 3309-1976 (E), ISO, 1976

NBS "Interface Between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) For Operation With Packet-Switched Data Communications Networks", Federal Standard 1041, Federal Information Processing Standards (FIPS) Publication 1041, National Bureau of Standards, 1983

BBN "Defense Data Network X.25 Host Interface Specification', Defense Communications Agency, 1983

GSA "Proposed American National Standard for Advanced Data Communication Control Procedures (ADCCP)", GSA, 1977