A NEW VANCOUVER PROTOCOL

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

Abstract

This paper describes a new datalink protocol which is being developed by the author in Vancouver and being tested in Toronto. It is designed to replace the previous link level protocol commonly known as the 'Vancouver protocol' and it addresses all the major limitations of that protocol.

Historical Background

In the summer of 1979 in Vancouver I had a protocol in operation in the VADCG TNC which is not well known. It was a protocol designed to talk to a 'Station Node' or central controller which dynamically assigned addresses to each TNC which connected to it. All TNCs communicated to each other through the connection services provided by this station node. The channel was shared using carrier sense multiple access (CSMA) techniques similar to most Amateur packet operation today. Although this system worked quite well and in many respects was more advanced than anything currently in use today we had two major problems which have resulted in it not being used today.

Firstly, the station node used the facilities of my S-100 CP/M system which was being used to develop both the station node and TNC programs. The VADCG did not have the funds or equipment to dedicate to the station node and I was not ready to donate my only computer to the cause. The group tried to assemble equipment which could be dedicated to the station node but were unsuccessful mainly due to funding problems.

Secondly, in the fall of 1979, I was asked by a group in Hamilton, Ontario to write a protocol for them which would allow communication directly from one TNC to another without the need of a station node intermediary. This request was made so that they could use and test the TNCs before they assembled the funds to build their own station node and then switch over to the Station node based software. They had a similar problem to that of the Vancouver group - lack of funds.

And so, by request, I modified the protocol used to communicate to the station node by eliminating the dynamic address assignment system and substituting a fixed address system and made a minimal amount of changes required to provide the requested facilities. Well, as things turned out this 'temporary' protocol became known as the 'Vancouver' protocol and became the 'standard' in North America for a few years. It is still a standard in common use in many areas today. We never have gotten back to the older station node software although it is still in our plans to do so.

When I wrote it, I never intended the program to become a standard for packet radio communications, only for testing TNCs - which it certainly did. It received widespread distribution mainly because it was the only thing available and also because it was capable of doing much more than just test TNCs. In spite of its limitations, it still meets the needs of most users at the present time. This article is written to address those limitations by implementing a new link level protocol which I believe can be used as a base on which to build network and transport level protocols.

A meeting of U.S. (only) packet radio groups was held in October, 1982 which resulted in the adoption of a protocol commonly known as 'AX-25' by most U.S. packet radio groups. (I would have liked an invitation.) AX-25 addresses most of the limitations of the Vancouver protocol but is not a true link level protocol since it concerns itself with several network level functions as well. It is also undergoing changes at the present time and has its own set of problems and limitations. It is my preference to have a protocol which makes a clear separation of link level and network level functions as per the IS0 model. As far as standardisation is concerned, when signals with other protocols arrive in the area we hope to be able to provide a gateway interface to handle them.

For purposes of discussion I will refer to the orginal Vancouver protocol as the 'Vancouver protocol' and the new protocol as 'Vancouver Version 2 protocol' or 'V-2' for short.

INTRODUCTION TO V-2

The V-2 protocol is intended to be an efficient datalink level protocol specifically for the Amateur Radio environment but potentially usable in other environments as well. Both full and half-duplex modes are defined. After link establishment, V-2 has only 5 octets of overhead per frame in addition to the standard 3 or 4 octets required for framing. It is strictly a datalink protocol and does not provide any network level functions. The network level protocol will be provided in a Network header which appears only in Information-transfer frames.

All control information for the link and only control information for the link is carried in a Link Header and Link Trailer which appear at the beginning and end of every frame respectively. Only the information not in the link control fields need be passed to higher levels of protocol. This separation of both function and headers allows software for layers to be developed in modular form and permits modification of the code for one layer to have minimal affect on other layers.

This protocol makes no claim or intent at comparison or simulation of the CCITT X-25 standard although it is the author's opinion that V-2 can be compared in the same way as AX-25 can to X-25 with as much similarity or more.

Frame Structure

All transmissions are sent in frames similar to IBM's SDLC and the CCITT's HDLC as well as to AX-25 and the older Vancouver protocol. The frames are encoded in (and decoded from) NRZI (Non Return to Zero Inverted) mode by the HDLC protocol controller chip. The general field format of a V-2 frame is as follows:

SYNC	FLAG	ADDR	CONTROL	INFORMATION	FCS	FLAG

Field Descriptions

Sync field

This field is used for preframe synchronisation. It is either 16 bits of zeroes or sufficient flags for the receiving side to establish bit synchronisation. This field is basically under control of the HDLC protocol controller chip and is transparent to the software.

Flag fields

All frames begin and end with a flag which is the bit sequence 01111110. When sending multiple frames one after the other, the trailing flag of one frame may be used as the leading flag of the next frame. The HDLC protocol controller chip uses a bit stuffing and deleting technique to ensure that a flag sequence can only appear at the beginning and end of a frame. Flags are automatically inserted by the HDLC protocol controller chip on transmit and are likewise automatically deleted from the data stream before passing to the software. i.e. the flags are transparent to the software.

Address field

This is a four byte address used to identify the link. The old protocol used a one byte address. How it is formed is discussed later. This field is generated and checked by software.

Control field

This is a one byte field. It is used to identify the type of frame and provides information for supervision of the flow of data over the link. This field is generated and processed by software. The format of this byte is discussed later.

Information field

This variable length field is only present in certain types of frames. It must be an integral number of octets. Although the hardware allows a maximum length of over 65,000 bytes, it is recommended that a maximum length of only 200 bytes or less should ever be created for transmission. On the other hand, the system should be capable of handling frames of longer lengths say up to 250 bytes when received from other nodes. This allows room for higher level overhead should it be necessary. If unusually long frames are to be sent, higher level protocols should agree on the size in advance.

FCS (Frame Check Sequence)

This is a two byte field generated as per IS0 standard 3309. This field is automatically generated, checked and removed by the HDLC protocol controller chip so I will not discuss it further. It is used to verify the accuracy of the rest of the data in the frame.

Frame Types

There are three types of frame formats: unnumbered, supervisory, or information transfer format. The frame type is indicated by the value of the low order bit or bits in the control field as seen in storage. (Note that this article refers to the fields as they are seen in storage - not as how they are transmitted by the protocol controller chip) If the low order bit (bit 0) is 0 then it is an information transfer frame. If both bit 0 and bit 1 are 1 then it is an unnumbered frame. Finally, if bit 0 is 1 and bit 1 is 0 then it is a supervisory frame.

Unnumbered-format frames (U-frames) are used during link establishment and termination. They are also used to transfer data when the data is not to be checked as to its location in a sequence of frames. There are a possible 32 types of U-frames but only three of these are used in this protocol.

Supervisory-format frames (S-frames) are used to assist in the transfer of information in that they are used to confirm preceding frames carrying information. The frames of' the supervisory format do not carry information themselves. These frames carry an Nr count which is used to confirm received frames. They also convey ready or busy conditions which assist in coordinating flow control across the link and they are also used to report frame numbering errors (indicating that a numbered information frame was received out of its proper sequence). There are a possibility of 4 types of S-frames but only 3 are defined in this protocol.

Information-transfer format frames (I-frames) actually transfer the data from higher protocol levels across the link. The control field contains send and receive counts (Ns and Nr), which are used to ensure that these frames are received in their proper order (Ns) and to confirm accepted information frames (Nr). The Ns count indicates the number of the information frame within the sequence of information frames transmitted. The Nr count transmitted in a frame is the number (Ns) of the information frame that the node transmitting the Nr expects to receive next.

Frame numbering

A node transmitting numbered I-frames counts each such frame and sends the count with the frame. This count is a sequence number known as Ns. This sequence number is checked by the receiver's datalink layer for missing or duplicated frames.

A node receiving I-frames accepts each numbered frame that it receives (that is error-free and in-sequence) and advances its receive count for each such frame. The receiver count is called Nr. If the received frame is error-free, a receiving station's Nr count is the same as the NS count that it will receive in the next numbered information frame; that is, a count of one greater than the Ns count of the last frame received. The receiver confirms accepted numbered I-frames by returning its Nr count to the transmitting node.

The Nr count at the receiving station advances when a frame is checked and found to be error-free and in sequence: Nr then becomes the count of the "next-expected" frame and should agree with the next incoming Ns count. If the incoming Ns does not agree with Nr, the frame is out of sequence and Nr does not advance. Out-of-sequence frames are not accepted. The receiver does, however, accept the incoming Nr count (for confirmation purposes) if the out-of-sequence frame is otherwise error free.

The counting capacity for Nr and Ns is 8, using the digits 0 through 7. These counts wrap around; that is, 7 is sequentially followed by 0. up to seven unconfirmed, numbered information frames may be outstanding (transmitted but not confirmed) at the transmitter. All unconfirmed frames must be retained by the transmitter, because it may be necessary to retransmit some or all of them if transmission errors or buffering constraints occur. The reported Nr count is the number of the next frame that the receiver expects to receive, so if, at a checkpoint, it is not the same as the transmitter's next frame (Ns) number, some of the frames already sent must be retransmitted.

The Nr and Ns counts on both ends of the data1 ink are initialized to zero during the link establishment phase.

The Poll bit (P-bit) and receive timeout.

The Poll bit is bit 4 in the control field. It is used to force a response from the receiving node when set to 1 (on). Whenever a frame is transmitted with the P-bit on a receive timeout is started by the originating node. This receive timeout (T1) is in the order of 1-3 seconds. This timeout is cancelled by reception of any frame from the other side of the link. If the timeout expires before the expected frame is received then the transmitting node takes corrective action. The number of successive timeouts is counted and reset when a frame is successfully received. If the number of successive receive timeouts exceeds a predetermined level (N1), the next higher level of protocol is notified. The higher level may take other action such as terminating the link.

Note that the receive timeout only proceeds when the data link channel is clear. The timeout is suspended when the data link channel is occupied. On VHF, the datalink channel may be occupied by carriers, voice transmissions, QRM, etc. in addition to the expected data signals. For this reason, it is strongly recommended that CD (Carrier Detect) be tested for receive timeouts rather than DCD (Data Carrier Detect). All VADCG board installations that I know of in Canada are using the squelch line from the VHF transceiver rather than the Data Carrier Detect line from the modem. Some can select both. If the DCD line is used, excessive timeouts will occur and it will be difficult or impossible for the TNC program to determine proper action. In addition, the TNC will transmit on top of other stations using the channel which can lead to unpleasant verbal exchanges. Remember that no Amateur frequencies are exclusively reserved for this type of packet activity.

NODE NAMING AND ADDRESSING

Before describing link addressing it is necessary to understand the network node addressing and naming system in which this protocol is designed to operate. Also, some terms need to be defined.

Node - In this article the word, 'Node' refers to any entity in the communications system which originates or receives frames.

Node Names

Associated with each node in the network is a 7 character upper case string of ASCII characters. The first six characters of which are the Amateur Radio call sign padded by ASCII blanks (20 Hexadecimal) on the right. The final character is a suffix used to discriminate between multiple nodes which have the same call sign. This suffix will normally be an ASCII blank (20 hexadecimal) and any additional occurrences of the same call sign will be identified by the ASCII numbers 1,2,3, etc. Each node in the network thus has a unique call sign.

For example:
KA6M*** (The * represents a blank)

Or if KA6M has another node operating:
KA6M**1

Node Address

Also associated with every node in the network is a two byte (16-bit) binary address. This address can be derived from the node name by use of a modified cyclic redundancy (CRC) checking algorithm. The algorithm operates on the 7-byte node name handled as a binary number and generates a 2 byte number.

The algorithm does not generate node addresses with FF (hexadecimal) in the first byte because these are reserved for special broadcast functions. Please see Appendix A for details of the algorithm and a sample implementation for an Intel 8080 microprocessor.

The special destination node address of FFFF (Hexadecimal) is used as a general broadcast address and destination addresses beginning with FF are reserved for selective broadcasts and special purposes.

Link Address

The link address field in the frame is four bytes long and is generated by concatenating the 2-byte node addresses of the nodes at either end of the link. Each link actually has two addresses, identifying the two directions that data can flow accross the link. For example, if A and B represent the node addresses of linked nodes then AB and BA are the two addresses of the link between the pair. Address AB represents the link address for data originated by B and destined for A while BA represents the link address for data originated by A and destined for B. The destination node address always comes first.

Use of Selective Receive

This link protocol has been designed to make good use of the selective receive function available on most HDLC/SDLC protocol controllers such as the Intel 8273. The 8273 can be set to pass to the software only frames which have a specific combination of bits in the first byte after the leading flag. In this protocol, this byte is the first byte of the destination node's address. If a node does selective receive only on the first byte of its address, it can eliminate 99.6% (on average) of the link traffic from having to be read into memory buffers and analyzed by software and yet still be able to do multiple link establishments, terminations, etc.

There are a number of modes of operation possible with this protocol using selective receive options as follows:

1. Selective receive only on the first byte of the node's address for using multiple links as above. This would normally be used in anything but the unlinked state but would be used in the unlinked state if it was not desired to do any of the activities in items 2 and 4 below such as a node with a CBBS host.

2. Selective receive only on address FF (Hexadecimal) to monitor broadcast type transmissions and not establish links.

3. Selective receive on address FF and also the first byte of the node's address. (2 selective receive addresses are supported on the 8273) Links can be set up and broadcast messages can be received. Useful in the unlinked state.

4. Receive on all addresses. This is not a selective receive but it is useful for evesdropping on transmissions intended for others or monitoring activity on the channel to put it more politely.

Link States

Data links are set up and discontinued as required by the nodes in the network. They are temporary. The normal life history of a datalink usually progresses through three basic phases or states namely: establishment, information transfer and termination. Although technically not a link state, the situation where a node has no datalinks in any of the above phases is called the 'unlinked' state.

INFORMATION-TRANSFER FRAMES

I (Information) frames are numbered. The Ns count provides for numbering the frame being sent and the Nr provides acknowledgement for the I frames received. When duplex information exchange is in continual process! each node reports its current Ns and/or Nr counts in each I or S frame exchanged. I-frames are only originated by nodes that are in the information-transfer phase.

The expected acknowledgement is an S or I format frame whose Nr count confirms correctly received frames. (S-frames may be interspersed with I format frames, as needed.) An I-frame always has an I-field, in fact, an I-format frame is considered invalid if it does not contain an I-field in this protocol. See figure 1 for the layout of the I-frame control field.

Control Field Type	Control Field Bits 765	4	321	0
I-frame	Nr	P	Ns	0

Where:
Ns is the send sequence number.
Nr is the receive sequence number.
P is the Poll bit.
Figure 1. I-Frame Control field

I-Field Content

The contents of the I-field in an I-frame are not the concern of the datalink protocol. They are determined by higher levels of protocol. The design of the V-2 network level has not been completed and will be the subject of another paper. However, the work on the network level has progressed sufficiently that some advice can be given to network level implementors and testers that will allow different network headers to be identified and to co-exist on the same channel. It is for this purpose that the following Network level information is included here.

The first sub-field in the I-field is called the Network Header. It is a variable length field composed of an even number of bytes. The low order four bits of the first byte indicate the number of words (1 word = 2 bytes) long the the Network Header is. The high order bit indicates if there is another header following the Network Header and the three remaining bits are used to identify different types of Network Headers that may have the same length.

The bit layout of the first byte in the Network Header is as follows:

```
Bitnumber  765432 10
           F T T T L L L L
```

Where: F indicates another header follows if 1
T T T indicates the Network Header type
L L L L indicates the length (in words) of the Network Header.

Even if only the link level protocol is in use the following two-byte dummy Network Header should be included as the first subfield in the I-field:

0100 (Hexadecimal)

This will indicate that the header is 2 bytes long and that no other header follows this one.

SUPERVISORY FRAMES

There are three types of supervisory frames defined by this protocol:

RR Receive Ready
RNR Receive Not Ready
REJ Reject

An S (Supervisory) frame must not contain an I-field. It is considered invalid if it does. See figure 2 for the layout of the control field of an S-frame. S-frames are only originated by nodes on links which are in the information-transfer phase.

Control Field Type	Control Field Bits 765	4	3 2 1 0
RR	Nr	P	0 0 0 1
RNR	Nr	P	0 1 0 1
REJ	Nr	P	1 0 0 1

Figure 2. S-frame Control field.

RR (Receive Ready)

Indicates that the originating node is ready to receive I-frames and acknowledges receipt of numbered I-frames through Nr-I. It must be sent to clear a previous not ready condition (RNR). It can also be sent with the P-bit on to elicit a response from the node at the other end of the link. It is sent with the P-bit off in response to a poll by the other side when no I-frames can be sent.

RNR (Receive Not Ready)

Indicates that the originating node is temporarily not ready to receive I-frames and acknowledges receipt of numbered I-frames through Nr-1. It is usually sent when buffering limitations and other internal restraints in the node are encountered. It can be sent with the P-bit on to elicit a response from the node at the other end of the link. It is sent with the P-bit off in response to a poll by the other side when no I-frames can be sent. This not ready condition must be cleared by the sending of an RR or REJ frame. Note that the node should still be capable of handling S and U frames even in the not ready condition.

REJ (Reject)

Acknowledges receipt of numbered I-frames through Nr-1 and indicates that the originating node is ready to receive I-frames. It requests the retransmission of numbered I-frames starting at the Nr contained in the REJ frame. Its purpose is to speed up recovery of dropped frames when operating full duplex. Note that this frame is only used with full duplex links and should never be sent on half duplex channels. Only nodes intending to operate full duplex need incorporate support for this type of frame. It is optional. REJ frames may be interspersed in the sequence of transmitted frames on full duplex links. The REJ condition is cleared when the requested frame has been correctly received.

UNNUMBERED FRAMES

There are three types of unnumbered format frames supported by this protocol:

XID Exchange station identification
DISC Disconnect
UI   Unnumbered information (formerly NSI)

Unnumbered frames are not sequence-checked and do not use Nr or Ns. See figure 3 for the layout of the control field for these frames. Although I am using names and control field formats common to HDLC/SDLC nomenclature, the actual usage of the unnumbered frames in this protocol is different. I tried to use the HDLC name with the closest approximation to what this protocol is doing. I hope this does not cause any confusion. My other alternative was to use the undefined HDLC U-frames and give them my own names.

Note that the previously described I and S-frame usage adheres fairly closely to the HDLC/SDLC standards but there are major divergences in the usage of the unnumbered format frames. This protocol is not intended to be a copy of any other protocol but has been designed pragmatically. When a piece of another protocol fits the need, it has been used but when the need is unique, then unique solutions are designed. This method reduces the amount of development effort. It also helps those who are familiar with other protocols to quickly develop a familiarity with this one. On the other hand, there may be some confusion when divergences with the other protocols occur.

Control Field Type	Control Field Bits 7 6 5	4	3 2 1 0
XID	1 0 1	P	1 1 1 1
DISC	0 1 0	P	0 0 1 1
UI	0 0 0	P	0 0 1 1

Where:
P is the Poll bit.

Figure 3. U-Frame Control field.

XID (Exchange station Identification)

The XID frame is only used during the link establishment phase of the protocol. I was thinking of calling it the 'LINK' frame rather than XID. Before I-frames can flow across the link, information is exchanged between both nodes in the XID frames. The opposite node's XID information is analyzed and a determination is made whether or not the link can be established. If both nodes like what they see, then the link is established and I and S-frames can flow across the link. On the other hand, if one node doesn't like what it sees, then it sends an XID frame to the other with a reason code filled in which indicates what it didn't like and the link is not established. XID frames are only transmitted on links in the establishment phase. The XID frame A, C and I fields are as follows:

AAAACSSSSSSSDDDDDDDPTR

Where:
AAAA is the address field.
C is the control field (XID).
SSSSSSS is the transmitting node's 7-byte name.
DDDDDDD is the destination node's 7-byte name.
P is the protocol level field.
T is the link type field.
R is the reason field.

Protocol level field (P-field)

This is a one byte field in the XID frame indicating the version of protocol being used by the originating node. By testing this field, the receiving node may determine if it can communicate with this type of protocol. At the present time, this field is 01 (Hexadecimal), indicating this is Level 0 of the protocol (bit 0 is on). This version number should be changed whenever a new level is produced which has features or changes which are not compatible with previous levels. This feature has been added in anticipation of future revisions and extensions of the protocol.

The receiving node checks for a match on this field. Note that a node may support multiple levels of protocol in order to communicate with nodes of different levels. For example, a primary station which supports multiple protocols may send an XID saying that it can communicate with levels 0 and 1 of this protocol by sending a P-field of 03 (Hexadecimal) with bits 0 and 1 on. The secondary determines if communication is possible by and'ing its own version(s) with that in the incoming P-field. If the result of this operation is non-zero, then the protocol levels are compatible and the secondary will respond with a P-field indicating which protocol it is using. If the result of this operation gives more than one bit on, then the secondary can choose which of the protocols it wishes to use.

For example:

1. Primary sends P-field of 03 indicating it can support levels 0 and 1.

2. Secondary logically 'ands' the primary's P-field with its own protocol version number of 01 (Only supports level 0). The result is 01.

3. Since the result was non-zero, the protocols are compatible and the Secondary can respond with a P-field value of 01.

On the other hand, if the result is zero, then there is a protocol incompatibility and the secondary will respond with the protocol mismatch bit (another P-bit) set in the Reason field (R-field). The link will not be established.

Link type field (T-field)

The T-field is a one byte field in the XID frame indicating the type of link being established. At present the only choice in link type is either full duplex or half duplex. The byte format is as follows:

```
Bit number  7 6 5 4 3 2 1 0
            X X X X X X X F
```

Where X indicates reserved bits - must be 0
F indicates full duplex if 1, half duplex if 0

This field is checked by the receiving node and if it is capable of handling the type of link indicated it will respond with a matching indication in its own T-field. But if, for example, the primary requests a full duplex link and the secondary only has support for half-duplex, the secondary will respond with the Link type mismatch bit (T-bit) set in the reason field (R-field).

Reason field (R-field)

The reason field is a one-byte field in the XID frame used by the secondary to report to the primary the reason why the link is not being established. If any bits are on in this field, the l-ink was not established. The R-field layout is as follows:

```
Bitnumber  1 0
    X X X X A R T P
```

Where:
X - reserved bit - must be 0.
A - Address duplication (See Restrictions)
R - Resource limitation
T - Link type field mismatch (see T-field desc.)
P - Protocol mismatch (see P-field description.)

The R-bit being on (1) means that the originating node has a temporary resource shortage which prevents the establishment of the link. The most likely reason being that it has a limitation on the number of simultaneous links that the node can support. Most nodes used in Amateur Radio so far can only support one link at a time, and so, if a node has this link already established when it receives an XID frame from a third party node, it should respond with the R-bit turned on.

Link Establishment

For purposes of the following discussion the node initiating the connection transmits first and is called the 'primary' node. The node being connected to is called the 'secondary' node. This distinction is made because the primary and secondary nodes use some of the fields in a slightly different way. This is a different use of these terms than is found in other protocol documents.

The following chart shows a normal link establishment between the primary node whose address is A and the secondary whose address is B.

```
BA,XID-P --->                 Primary transmits XID
              <--- AB,XID     Secondary transmits XID
BA,I-P ----->                 Primary transmits I
etc.
```

Conflict Resolution

Although unlikely, there is a possibility of the occurence of conflicting requests during the link establishment phase. This is not possible during other link phases. A conflict occurs if two nodes try to initiate a link with the other at the same time but with conflicting requirements. For example, one node wants to establish a half-duplex link while the other wants to establish a full-duplex link or the protocols they want to use are different. Even if both sides take the same recovery action this still does not solve the problem. One side must gain the upper hand. The resolution method chosen is that the node with the higher address gets its way and the node with the lower address must report to its higher level that a link was established but not with the desired specifications due to conflict.

Disconnect frame (DISC)

The DISC frame is only sent during the link termination phase and it is the only type of frame that can be sent on links in the termination phase. I think this frame would better be called 'Unlink' because that is a better description of the function it performs. Connections (in my opinion) are the concern of higher protocol levels - not the datalink level. The DISC frame is only sent after the link has been established and indicates that the originator will no longer send or receive I and S-frames on the link. It is sent with the Poll bit on by the primary node (The node initiating the link termination) and sent with the poll bit off by the receiving secondary node to indicate that it will no longer send I and S-frames on the link.

The format of the DISC frame is the same as that of the XID frame described above except that the usage of the P, T and R fields is different. In the DISC frame the Reason field (R-field) indicates the reason for the disconnect when sent with the Poll bit on by the primary node. If the R-field has no bits turned on this is a normal link termination but if a bit or bits are on in the R-field, then this link is being terminated because of an error. The R-field bits are defined as follows:

```
Bitnumber  7 6 5 4 3 2 1 0
           0 0 0 0 Z Y X W
```

Where:

1. If W is on, the originating node has received an invalid or non-implemented control field.
2. If X is on, the originating node has received a frame with a prohibited I-field.
3. If Y is on, the originating node has received an I-frame with a length greater than the maximum the node can support.
4. If Z is on, the originating node has received a frame with an incongruous Nr.

If any of the bits are on in the R-field, the P-field contains the control field of the frame received which caused the error condition and the T-field contains the following information:

```
Bit number 7 6 5   4   3 2 1   0
             Vr    0    Vs     0
```

Where Vr is the next expected Ns value to be received and Vs was the next Ns value to be sent at the time the error was detected.

The P,T and R fields are undefined (and should be ignored) in a DISC frame with the Poll bit off. Similarily, the P and T-fields are undefined in a DISC frame with a value of 0 in the R-field.

Even though the node names are unnecessary from a technical point of view in the DISC frame, they are included in the Amateur Radio environment to give information to other nodes monitoring the channel as to who the two nodes are who have terminated the link between them. The names are also sent out when the link is being established as well. This is somewhat similar to the Amateur practise of identification at the beginning and end of each QSO.

Unnumbered Information Frame (UI)

The UI-frame is included in this protocol as a method of transmitting information when a link has not been established or for bypassing the normal handling of I-frames when a link is established. One example of such use would be a repeater identifying itself every few minutes - a type of broadcast message. It is included in the protocol because of a need in the Amateur Radio environment for such a function.

Since UI-frames are unnumbered, they should not be acknowledged whether or not they are transmitted with the Poll bit on or off. If one is lost, there is no way of recovering it. No error reporting is done for UI frames and the UI frame should only be passed on to a higher level when the data link is not established, not being established and not being terminated. For use as a type of general broadcast, the link address field in a W-frame should use the special general broadcast address of FFFF (Hexadecimal) as the first two bytes of the link address. If the broadcast is only intended for a specific area then the first byte should be FF and the second byte should be an area or group number. And if it is intended for a specific node then that node's address can be used in the first two bytes.

Responses to polls

When a node receives a frame with a P-bit on, it must respond. The type of response varies depending on the type of frame that contained the P-bit. The response frame does not have the P-bit on except in the case of the I-frame response where the P-bit is allowed. The responses possible for each type of frame containing a P-bit are shown in Figure 4 following:

Polling Frame Type	Responding frame(s)
I	I or RR or RNR or REJ*
RR	I or RR or RNR
RNR	RR or RNR
XID	XID
DISC	DISC
UI	none
REJ*	I

* = REJ only used in full-duplex
Figure 4. Poll Responses

HALF-DUPLEX PROCEDURES

Half-duplex links assume that only one side transmits at a time. For this reason they can be used on a single two-way communications channel. Half duplex protocols can also be used on independent transmission channels as well but full-duplex would be a more efficient method when these facilities are available. Channel sharing with multiple nodes require half duplex protocols. V-2 provides a half-duplex multipoint protocol. This protocol can be used in a point-to-point environment as well with very little reduction in efficiency compared with a protocol only designed for point-to-point work.

Information transfer

When a node has one or more I-frames to send, it sends them in sequence and in conformity with the requirements described in "Frame numbering" previously. Since I-frames always require a response, the P-bit is turned on in the last frame of the transmitted sequence of I-frames. The exception to this rule occurs when it is necessary to transmit an S-frame with the P-bit on in the same transmission as well. either case, a receive timeout is started at this time as previously described in the section, "The Poll bit and receive timeout". The node then turns the link around and listens for a response. Note that the reception of an I-frame with or without the P-bit on demands a response.

If the receive timeout expires meaning nothing was received, the node will not retransmit the previous frames but transmit an RR or RNR frame with the poll bit on and again wait for a response. This operation repeats itself until something is received from the other side. (The higher level of protocol is notified after every Nl consecutive timeouts.)

When a response is received, the node starts transmitting I-frames again,, beginning with the first unacknowledged I-frame.

Not ready condition handling

When a node becomes not ready to receive I-frames on a link it will send an RNR frame with the poll bit on at the earliest opportunity. If permitted by the other side, I-frames may be sent along with the transmission of RNR but only the RNR frame should have the poll bit on in this case and it should be sent last in the transmission. The node then listens on the link.

If the receive timeout expires, only the RNR frame with the poll bit on is retransmitted. If an I-frame is heard the RNR frame with poll bit is retransmitted. (Note - the node may accept or ignore the I-frame at its discretion even though it is trying to tell the other side to stop sending them.) If an RR or RNR frame with the P-bit on is heard, the node should respond with I-frames followed by an RNR frame with the P-bit on if it can send I-frames or otherwise by transmitting two RNR frames, one with the poll bit on and the other with the poll bit off and start a receive timeout. (Note that two are required, one to respond to the poll from the other side and one to do a poll for the proper response from the other side. Finally, if an RNR or RR frame with the P-bit off is received this will be accepted by the node as an indication that the other side is aware of the not ready situation and the node can go back to sending I-frames if it has any to send and is permitted to send them. Most of these actions can be deduced from Fig. 4 above.

Note that no node is permitted to send I-frames after receiving RNR and before receiving RR.

Ready condition handling

When a node becomes ready to receive I-frames on a link it will send an RR frame with the poll bit on at the earliest opportunity. If permitted by the other side, I-frames may be sent along with the transmission of RR but only the RR frame should have the p011 bit on in this case. The node then listens on the link.

If the receive timeout expires, only the RR frame with the poll bit on is retransmitted. If an RR or RNR is heard with the poll bit on the node should respond by either transmitting I-frames followed by an RR with the poll bit on or otherwise by transmitting two RR frames, one with the poll bit off and the other with the poll bit on and start a receive timeout. If an RNR or RR frame with the P-bit off or an I-frame is received this will be accepted by the node as an indication that the other side is aware of the ready situation and the node can go back to sending I-frames if it has any to send and is permitted to send them.

FULL-DUPLEX PROCEDURES

A full duplex link requires two independent channels for the transfer of data. The full duplex protocol is designed so that it is possible for both sides to continuously transmit data to the other side simultaneously. One channel is used for transmissions by node A for example and the other channel is used for transmissions by B. Because of the potentially continuous nature of these transmissions, it is usual that each node never listens on its own transmitting channel. However, a node may delay its transmissions if it has some way of knowing that its transmitting channel is unuseable.

This version of V-2 only defines a point-to-point full-duplex protocol. This has uses in 'backbone' linking for an Amateur Radio digital communications network and for satellite work. It is an option which does not have to be implemented in all nodes. Extensions to V-2 for multipoint full duplex operation are being planned but are not within the scope of this paper.

Information transfer

In full-duplex each node listens even when transmitting. Frame numbering and acknowledgment is similar to half-duplex except that acknowledgments are done 'on the fly' even while the other side is transmitting.

A node sends I-frames in sequence until it either has no more frames to send or there are 7 frames outstanding. At this point it will put the P-bit on in the final frame to request acknowledgement from the other side and begin a receive timeout. In order to improve throughput on links with a long turnaround delay - such as satellites, it is permissible to put the poll bit on in a frame before 7 frames are outstanding and still continue to transmit up to the maximum of 7 unacknowledged frames. This technique is called 'pacing'.

If the receive timeout expires meaning nothing was received, the node will not retransmit the previous frames but transmit an RR or RNR frame with the poll bit on and again wait for a response. This operation repeats itself until something is received from the other side. (The higher level of protocol is notified after every N1 consecutive timeouts.)

When a response is received, the node starts transmitting I-frames again, beginning with the first unacknowledged I-frame.

If a node receiving sequenced I-frames encounters a frame not in sequence it should send a

REJ frame at the earliest opportunity it gets. The node receiving the REJ should go back to transmitting the last unacknowledged frame and continue from there. The node should not transmit any more REJ frames until the reject condition is cleared by the receipt of the numbered I-frame indicated in the REJ frame.

Not ready condition handling

Not ready conditions are handled similarly to that described in the half-duplex environment except that the channel doesn't have to be turned around.

Ready condition handling

Ready conditions are handled basically the same as described in the half-duplex procedures.

THE BIG PICTURE

The **datalink** layer of protocol described herein is only one part of a complete communications protocol. As per the suggested IS0 protocol layering structures, this protocol only interfaces with the two adjacent layers in the node in which it is running and with the **datalink** layer on the other side of the link.

The **datalink** layer receives data and orders **from** the Network layer immediately above it. The network layer is a 'higher' level of protocol **and** sort of acts like the **datalink** layer's **'boss'. The datalink** layer acts on these commands and reports on its progress and status as the job is done. When unusual conditions occur which the **datalink** layer **can't** handle it goes to its **'boss'** for advice.

The **datalink** layer passes data and commands to the next lower protocol layer - the Physical layer which in the Amateur environment is usually the HDLC Controller chip. The physical layer acts on the commands from the **datalink** layer and returns data and status information to the **datalink** layer by means of status lines and interrupts.

The **datalink** layer also communicates to the **datalink** layer on the other side of the link.

Even though it is only the third of these interfaces which is the subject of this paper, it should be realized that in a real implementation of a **datalink** protocol there are actually three different protocols involved: i.e. the three communication interfaces. The type of protocols used for adjacent level communication are very system dependent but I will discuss in a general **way,** the type of information that is exchanged between the Network and **Datalink** layers.

Interface with the Network **layer**

Commands to the **datalink** layer and data to be passed **accross** the link are passed from the network layer to the **datalink** layer. The basic commands needed are:

1. Link - This is a command which orders the **datalink** layer to establish a link. The data passed with this command indicates what type of link (either Full or Half Duplex) is desired and the node name of the node to be linked to.

2. Unlink - This is a command which orders the **datalink** layer to terminate a link.

Status information and information received from the link in I-frames are passed to the Network layer from the **datalink** layer. The following status information should be passed:

1. When a **datalink** becomes established. The type of link and node name should be passed as well.

2. When a **datalink** is terminated. The reason code and node name should be passed as well.

3. Whenever a predetermined number of consecutive timeouts **(N1)** on a link has occurred.

4. When an invalid frame is received on a link.

The above lists are not necessarily complete but should serve to give the reader a good idea of the type of information needed. Each implementation will have its own method of handling these interfaces and it is not the intent of this paper to give detailed information as to how the information is to be passed.

SAMPLE EXCHANGE

The following example shows a V-2 exchange with no errors or exceptional situations encountered It is beyond the scope of this paper to show an example of all possible types of **exchanges.** It is hoped that the information in this paper will be sufficient for the reader to determine what the exchanges would be when receive timeouts, lost frames, invalid **frames,** and colliding frames are encountered. The **A,C** and I-fields are shown separated by commas. The address field is displayed in hexadecimal. The control field is represented as follows:

T (Ns) P (Nr)

Where T is the frame type acronym and P is the poll bit. The (Ns) and (Nr) are only shown for frames which contain them and the P is only shown if the poll bit is on **(1).**

The Network header sub-field in the I-field in I frames is shown in Hexadecimal characters and the rest of the I-field is shown in ASCII characters. The subfields in XID and DISC frames are shown separated by commas, the node names being in ASCII characters and the **P,** T and R fields in Hexadecimal. Time progresses downward in the following charts.

The communication is between two nodes. Here is the pertinent information about the two nodes.

Call sign	**VE7APU1**	**KA6M**
Node number	**627B**	68ED

Exchange **1.**

This example shows link establishment, information transfer and link termination. A complete QSO. Arrows to the right indicate transmissions from **VE7APU** and those to the left indicate transmissions from **KA6M.**

68ED627B,XID-P,VE7APU1,KA6M ,P=01,T=00,R=00 ---->

<------ 627B68ED,XID,KA6M ,VE7APU1,P=01,T=00,R=00

68ED627B,I(0)P(0),0100,Hello Hank **---------------->**

<-------------- 627B68ED,I(0)P(1),0100,Goodbye Doug

68ED627B,RR(1) ------------------------------------>

68ED627B,DISC-P,VE7APU1,KA6M ,P=00,T=00,R=00 --->

<----- 627B68ED,DISC,KA6M ,VE7APU1,P=00,T=00,R=00

The first line shows **VE7APU** has entered the link establishment phase at request of the Network level and is trying to initiate (Poll bit is on) a half duplex link (Link Type = 00) with **KA6M** using version 0 level protocol (Protocol = **01**). Because the Poll bit is on we know **VE7APU** has started a line timeout.

The second line shows **KA6M** has entered the link establishment phase and is responding (the Poll bit is off) positively (the Reason field = 00) using version 0 level protocol. When this frame is received, **VE7APU** will enter the information transfer phase and cancel the line timeout.

The third line shows **VE7APU** sending an I-frame with data to **KA6M** and demands a response from **KA6M** (the Poll bit is on). **VE7APU** starts a line timeout at this time. When **KA6M** receives this frame it will enter the information transfer phase.

The fourth line shows **KA6M** sending an I-frame with data and acknowledging correct reception of **VE7APU's** I-frame **(Nr=1).** It is demanding a response from **VE7APU** and starts a line timeout.

The fifth line shows **VE7APU** acknowledging **KA6M's** I-frame by sending a Receive-Ready (RR) frame with Nr set to **1.** It has cancelled its receive timeout. The Poll bit is not on so **VE7APU** is not demanding a response from **KA6M** and so does not start a line timeout. **VE7APU** could have sent an I-frame at this point if it had any information to send.

The sixth line shows **VE7APU** has entered the link termination phase on orders from higher levels and is sending a DISC frame to **KA6M** to request termination of the link. The termination has not been caused by an error condition because the Reason field is 00. VE7APU is demanding a response from KA6M and starts a receive timeout.

In the last line, KA6M enters the link termination phase when it receives the DISC frame and responds (Poll bit is off) with a DISC frame. After transmitting this frame, KA6M goes into the unlinked state.

Since there are no further transmissions on the link we know that VE7APU has received the DISC, cancelled its receive timeout, and gone into the unlinked state.

RESTRICTIONS AND LIMITATIONS

1. This protocol will not function correctly if a node in the information transfer phase can hear frames from another node which is also in the information transfer phase and is using the same four-byte link address. Note that this would require two pairs (4) of nodes active on the same channel each pair using the same area number and node number. One node from each pair would have to be linked with one node from the other pair. Without going into any mathematical analysis I will state that, the probability of this situation arising is extremely low. (Somewhat similar to the probability of the FCC or DOC issueing the same call to two different stations!).

2. Although multiple links are allowed, a link will not be established to two nodes with the same address at the same time. The reason the link is not established is passed to the other node and corrective action could be taken - perhaps by changing the call sign suffix.

3. Multiple links between the same nodes are not allowed. The need for this fieature is questionable as all necessary data traffic accross the link should be able to be multiplexed by the higher levels of protocol. Although not defined in this protocol, a node could masquerade as a different node by supporting multiple node addresses and names by changing the call sign suffix.

4. This protocol only supports datalinks between pairs of nodes. Thus it does not support conferencing or roundtable communication among a group of nodes. This function is normally provided by higher levels of protocol. The complications and problems encountered in providing data integrity with this service at the datalink level are severe and I do not know of any commercial or Amateur datalink protocols which do this. However, a form of roundtable operation could be implemented which does not guarantee data integrity. Each node in the group could operate in unlinked mode transmitting and receiving data in UI frames. The members of the group could agree to all use a broadcast address FFxx (where xx=a group number) in the destination part of the link address and then only pass information from frames which had a link address of this form. Members of the group should be in good communication with each other because there is no way of recovering data from lost frames when using this method of conferencing.

IMPROVEMENTS OVER THE OLD VANCOUVER PROTOCOL.

1. Both full and half-duplex links are provided for.

2. Multiple links are supported.

3. Some multiple protocol support.

4. Vastly increased number of link and node addresses.

5. No coordination of node addresses is required.

6. A rudimentary conferencing system is provided.

COMPARISON OF V-2 AND AX-25

I am including this section because I am sure that many people will make comparisons and also because AX-25 is the only other Amateur Radio packet protocol which has a published specification document.

This comparison is difficult to do because despite statements in the literature that AX-25 is a link level protocol, it is, in fact, a type of network level protocol. AX-25 routes packets through a series of links from a source node to a destination node. Although AX-25 provides end-to-end flow control, sequencing and data integrity, it provides no link level error recovery, retransmissions, acknowledgements, sequence checking, etc. for any of the links in the chain. Only in the special case where the source and destination are connected by a single link could AX-25 be construed to be a link level protocol. In this special case, the end points of the connection become identical to the end points of the link and so the end-to-end connection facilities cannot be distinguished from the link facilities. Some comparison of the two protocols can be done when considering this special case.

A document describing a Network level protocol for V-2 will be published later. Routing of packets through a network is one of the responsibilities of the Network level. Code for a Network level is being written at the present time for the VADCG TNC. Code for this link level protocol has already been implemented on the VADCG TNC.

1. V-2 has a reduced protocol overhead compared to AX-25. A 4-byte fixed length address field as opposed to a 14-byte or greater address field in AX-25.

2. V-2 has a facility to select full-duplex or half duplex protocols at link establishment.

3. V-2 has a facility to select different protocols at link establishment.

4. V-2 uses an addesssing and naming system for nodes. AX-25 makes no distinction between names and addresses.

5. AX-25 has a network routing system. V-2 link level does not.

6. V-2 does not bit shift call signs and other names when transmitted.

7. V-2 makes use of the selective receive function of the SDLC/HDLC protocol controller chips to automatically eliminate frames that do not need to be checked. AX-25 cannot do this. For example, in this area, all call signs start with the character 'V'. Using AX-25, this would mean that all frames transmitted in this area would have the same character after the initial flag in every frame. This effectively renders useless the selective receive function present on the protocol controller chips and forces the software to read into memory and examine every frame that is transmitted on the channel.

SUMMARY

The general specifications of the V-2 protocol have been presented in this paper. As this is the first draft of a new protocol, some details may have been omitted. This document will be revised as necessary, to expand on areas not clearly presented and to include changes in the protocol. Those interested in obtaining the latest version of this document should contact the author. Anyone having questions or comments should do the same, preferably by writing.

At the present time most of this protocol has been implemented in the VADCG TNC. The source code for this implementation is available on CP/M 8-inch diskettes. Anyone wishing to implement it on another system should contact the author directly so that the work may be coordinated.

The author feels that V-2 link level is an efficient protocol both in terms of channel utilization and software requirements and that it is particularly suited to the Amateur Radio environment. It provides almost all the functions required by the IS0 data link model without overstepping into functions in the domain of other layers. It is intended as a building block upon which higher levels of protocol can be independently developed as per the IS0 proposals.

REFERENCES

1. Rutledge, J. H. "OSI and SNA: A Perspective" IBM Washington Systems Center Technical Bulletin, April 1981, GG22-9225-00

2. Data Processing - Open Systems Interconnection - Basic Reference Model, International Standards Organization Draft Proposal 7498, December 1980

3. Fox, Terry "AX-25 Level 2 Protocol" Second ARRL Amateur Radio Computer Networking Conference proceedings, March 1983

4. Fox, Terry "Alterations To AX.25 Level 2 Document By ARRL Digital Comm." AMRAD newsletter, November-December 1983

5. "Using the 8273 SDLC/HDLC Protocol Controller" Intel Peripheral Design Handbook, August 1981

APPENDIX A. Node Address Calculation

The CRC-16 algorithm divides the 7-byte character string of the Node Name by the following generating polynomial:

$$x^{16} + x^{12} + x^5 + 1$$

This is the same polynomial used in calculating the FCS field by the various HDLC/SDLC protocol controller chips. In the calculation, integer quotient digits are ignored and the 16-bit remainder is checked to see that the first byte is not FF (Hexadecimal) and if it is, the bytes are reversed. If the first digit is still FF the value used is 0000. This playing around with the value is done to ensure that none of the special purpose addresses beginning with FF are generated.

Many of you out there may find this a little difficult to understand so I am including a sample assembly language program listing below which actually does the above generation for an 8080 microprocessor. It should not be difficult to implement in other processors.

Program listing

```
                    ; ROUTINE TO GENERATE A NODE ADDRESS FROM A NODE NAME
                    ; WHEN CALLED, IT USES THE NODE NAME AT LOCATION 'NODENM' AND
                    ; LEAVES ITS NODE ADDRESS AT 'CRC' UPON RETURN
                    ; IT DOES NOT GENERATE ADDRESSES WITH FF AS THE FIRST BYTE

0000                ADRCALC:ORG     0
0000 210000                 LX1     H,0      ; INITIALIZE CRC FIELD TO 0000
0003 225100                 SHLD    CRC
0006 215300                 LX1     H,NODENM        ; POINT TO 7-BYTE NODE NAME
0009 1607                   MVI     D,7      ; NUMBER OF CHARACTERS
000B 7E             LOOP:   MOV     A,M      ; GET A CHARACTER
000C CD2D00                 CALL    CALCCRC  ; GIVE IT TO CRC ROUTINE TO PROCESS
000F 23                     INX     H        ; POINT TO NEXT CHARACTER
0010 15                     DCR     D        ; HAVE FINISHED WITH ALL THE CHARACTERS?
0011 C20B00                 JNZ     LOOP     ; NO, GO BACK TO PROCESS ANOTHER
0014 2A5100                 LHLD    CRC      ; GET THE GENERATED CRC ROUTINE
0017 7D                     MOV     A,L      ; IS THE FIRST BYTE FF (HEXADECIMAL)?
0018 FEFF                   CPI     OFFH
001A CO                     RNZ              ; NO, RETURN WITH NODE ADDRESS IN CRC
001B 7C                     MOV     A,H      ; YES, BAD LUCK, REVERSE THE CRC BYTES
001C 6F                     MOV     L,A
001D 26FF                   MVI     H,OFFH
001F 225100                 SHLD    CRC
0022 7D                     MOV     A,L      ; NOW IS THE FIRST BYTE FF?
0023 FEFF                   CPI     OFFH
0025 CO                     RNZ              ; NO, RETURN WITH NODE ADDRESS IN CRC
0026 210000                 LX1     H,0      ; YES, EXCEPTIONALLY BAD LUCK
0029 225100                 SHLD    CRC      ; SET NODE ADDRESS TO 0000
002C C9                     RET              ; AND RETURN, JOB COMPLETE.

                    ; THIS IS THE ACTUAL CRC-16 CALCULATION ROUTINE
002D E5             CALCCRC: PUSH   H
002E 0608                   MVI     B,8
0030 4F                     MOV     C,A
0031 2A5100                 LHLD    CRC
0034 79             CALCCRCl: MOV   A,C
0035 07                     RLC
0036 4F                     MOV     C,A
0037 7D                     MOV     A,L
0038 17                     RAL
0039 6F                     MOV     L,A
003A 7C                     MOV     A,H
003B 17                     RAL
003C 67                     MOV     H,A
003D D24800                 JNC     CALCCRC2
0040 7c                     MOV     A,H
0041 EE10                   XRI     10H
0043 67                     MOV     H,A
0044 7D                     MOV     A,L
0045 EE21                   XRI     21H
0047 6F                     MOV     L,A
0048 05             CALCCRC2: DCR   B
0049 C23400                 JNZ     CALCCRC1
004C $5100                  SHLD    CRC
004F                        POP     H
0050 C9                     RET

0051 0000           CRC     DW      0        ; WHERE NODE ADDRESS IS PLACED
0053 5645374150NODENM DB    'VE7APU1'        ; NODE NAME

005A                        END
```