Annex H. Implementation Guide and Notes
(information only)

This Annex contains rules and guidelines for adaptive speed control and other implementation issues based on NC3A experience with earlier systems. The best and most complete source of information on implementation topics is NC3A TM-937 “Open Systems for Radio Communications: A Subnet Architecture for Data Transmission over HF Radio”.

H.1 Flow Control

Flow control imposed by a client on the Subnetwork could cause the receiving queues of the HF Node to be filled, which in turn could cause older queued data for other clients to be discarded or result in a temporary pause in accepting and acknowledging error free PDUs. This situation is not acceptable since a client, of even a low Rank, could, in principle, cause a deterioration of the service provided to the other clients connected to the Node. Implementers are encouraged to implement one of the following approaches:

- minimize or eliminate the use of flow control on the subnetwork-to-client (i.e., receive) interface;
- employ an implementation-dependent queuing strategy that allocates buffers separately for each client;
- allocate sufficient buffer space that overflow has a low-probability of occurance.

H.2 Reasons for Data Transfer PDUs (D_PDUs) with Different Rules

Annex C to this document defines a number of different D_PDUs which may be used to transfer data: normal data transfer (D_PDU types 0-3), expedited data (D_PDU types 4 and 5), management message (D_PDU type 6), and non-ARQ (types 7, 8, and 15). This section reviews the use of the different D_PDUs and the reasons for having them.

H.2.1 Normal Data Transfer

The normal data transfer D_PDUs are intended in most cases for use in the transfer of encapsulated U_PDUs. It is difficult to efficiently handle high priority data in the context of this type of D_PDU alone. When a PDU from a higher layer reaches the data transfer sublayer, it is segmented into a number of D_PDUs that are assigned sequence numbers. While it may be possible to “unqueue” or cancel parts of higher level PDUs, this will either result in the loss of data (cancellation) or delays in transmitting the high priority traffic. Thus, the additional types of D_PDUs have been introduced to efficiently accommodate high-priority traffic.

H.2.2 Expedited Data

Expedited data D_PDUs are intended for use to support subnetwork peer-to-peer communications (primarily making and breaking physical links), and exceptionally, to provide a path for U_PDUs of the highest priority which bypasses all existing queues. An example of the first would be if it were desired to immediately break a link that has long queues of traffic pending. If this “break” C_PDU was handled as a normal data PDU, it might (depending on the system implementation) have fairly long delays (minutes) before it could be transmitted. With the expedited data mechanism, the C_PDU bypasses all the normal data queues and is transmitted at the beginning of the next transmission interval.

Expedited data bypasses all existing normal data queues (which are maintained during the handling of the expedited data). A number of expedited data D_PDUs (corresponding to, and not more than, a single expedited data C_PDU) may be transmitted in a single transmission. A stop and wait protocol is used for the acknowledgement of (the group of) expedited data D_PDUs. This service is intended for occasional use to transmit small amounts of data. Frequent use, or use with large U_PDUs, will degrade system performance, and therefore the use of the expedited data modes for client traffic is strongly discouraged.

H.2.3 Management

A third type of service is provided exclusively for system management functions, for example, to co-ordinate the adaptive change of the HF modem data rate. This function, and others like it, requires a service with the smallest possible delay and with maximum robustness. This is provided by the management data D_PDU. Management D_PDUs bypass all pending data D_PDUs (when the system enters the management mode, both normal and expedited data queues are put on hold).

Transmission of the management D_PDU type follows a D_PDU-by-D_PDU stop-and-wait protocol, with D_PDUs repeated as necessary to fill the HF modem interleave buffers and provide maximum robustness and efficiency. Only a single Management (Type 6) D_PDU may be outstanding (unacknowledged) at any moment; the D_PDU is repeated until acknowledged (or the link fails because of a time out).

H.2.4 Non-ARQ

Non-ARQ, or unacknowledged, D_PDU types are provided in order to allow the transfer of data which does not require acknowledgement or cannot reasonably be acknowledged. This is useful for “broadcast” or “multicast” modes of operation, in which nodes are not allowed to transmit because of EMCON restrictions or where D_DPUs are addressed to multiple nodes. It is also required to support functions that occur when the node is not in an ARQ-processing state, such as the peer-to-peer communication by the Channel Access Sublayer when establishing a connection.

H.3 Other topics

Several short topics are addressed in the subsections below.

H.3.1 EOT Calculation

The definition of the End-of-Transmission (EOT) field in D_PDUs gives a maximum transmission interval of 127.5 seconds, or just over two minutes. When computing the value of the EOT, the results of the calculation must be rounded up to avoid collisions between node transmissions on half-duplex/single-frequency channels.

H.3.2 Error Detection for D_PDU Headers

Because the D_PDU header is generally shorter than the data, errors are more likely in the data part of a D_PDU than the header. Protecting the header with its own CRC allows the possibility to detect and use uncorrupted header information even if the data part of a D_PDU contains errors. For this reason, the added overhead of the CRC-on-header field was deemed warranted in the design of STANAG 5066.

H.3.3 DROP PDU Processing

If a D_PDU is to be sent with the DROP PDU bit set, it is inefficient to send the data portion of the D_PDU. However, all D_PDUs that make up the PDU to be dropped must be sent to maintain window synchronisation. A D_PDU that is received with the DROP PDU flag set must still be acknowledged.

H.3.4 Size of User Data Field

The 10-bit field SIZE OF USER DATA indicates the size of the information field in bytes. Its value does not include the size of the CRC-ON-SEGMENTED-C_PDU field. Note that full-size D_PDU transmissions can involve significant transmission time, as, for example, 2^{10}=1023 bytes=8184 bits=109 seconds at 75 bps. A balance should be struck between protocol efficiency, which would encourage the use of large D_DPUs, and access and turnaround time, which would encourage small D_DPUs. An additional balancing factor in protocol efficiency must be weighed as well, since high-noise or fading environments will encourage the use of small (normally 100 to 400 byte) D_DPUs because of the increased likelihood of error (and therefore decreased throughput) with large D_PDUs. Subnetwork implementations that allow adaptation of the maximum D_PDU size to promote protocol efficiency in varying channel conditions are not necessarily precluded so long as they do not violate the mandatory requirements in this STANAG or prevent interoperation with subnetwork implementations that do not support such adaptation algorithms. This issue is addressed further in Section H.7

H.3.4 Forwarding of D_PDUs to the Channel Access Sublayer

To allow effective monitoring of channel activity by the Channel Access Sublayer, the Data Transfer Sublayer should deliver all identifiable *D_PDUs* (i.e. D_PDUs in which no errors were detected in the header) to the Channel Access Sublayer, even when they are *not addressed* to the receiving node. The Channel Access Sublayer filters the received C_PDUs according to address and type and, as appropriate, locally processes the C_PDU or passes the C_PDU to the Subnet Interface Sublayer.

In the non-ARQ modes of operation of the Data Transfer Sublayer, all identifiable *C_PDUs* (i.e. C_PDUs in which the header portion of one or more associated D_PDU's) was received without error are delivered to the Channel Access Sublayer. This procedure was adopted so that:

i) the Channel Access Sublayer could effectively monitor the channel activity, and
ii) a client could specify that only error free or, alternatively, that all identified S_PDUs be delivered.

The second option may be useful if the client handles printable text and/or implements additional error control functionality.

H.4 Synchronisation of the ARQ Machine

Establishment of a "connection" requires the initial synchronisation of the peer protocol ARQ machines in the Data Transfer Sublayer. This process can be viewed as automatic in the sense that the peer ARQ machines associated with a new connection (i.e. one for which an ARQ machine must be established) will be automatically reset, and valid numbers established for the upper and lower window edges at transmitter and receiver. The on-going assumption that the ARQ machines associated with a revived data state connection (i.e. one for which an existing ARQ machine is re-activated) remain in synchronization over long periods of time may not be strong, however, because of node failure, undetected message error, or other anomaly. For these reasons, the procedures for synchronization of the ARQ machine on a reliable link were deemed sufficiently important for correct link operation to be defined as mandatory requirements for subnetwork operation. These synchronization procedures involve two steps:

1) on-going tests of ARQ machine operation through validation of the upper- and lower-window edge values in D_PDUs exchanged over the link (defined in Annex C section C.6.3) and
2) definition of the negotiated FULL-RESET ARQ resynchronization protocol using the RESET/WIN-RESYNC (Type 3) D_PDU (defined in Annex C section C.6.5).

Note that in addition to the negotiated FULL-RESET procedure of Annex C, the destination node can independently re-synchronise its receive LWE and UWE pointers to the indicated TX FRAME SEQ # *UWE* pointer of the originating node. This is equivalent to signaling a group ACK to all frames transmitted by the originating node. All synchronization options result in some loss of data although, in general, a negotiated re-synchronisation results in the loss of a smaller number of frames.

H.5 Use of Rank and Priority Arguments (Subnet Interface Sublayer)

The rank of subnet clients is used to determine the allocation of subnet resources. As an example, if a new client attempts to come on-line (bind to a node) but not enough resources are available, the Subnetwork Interface Sublayer is permitted to unilaterally declare a client with lower rank off-line in order to release resources for the higher-ranked client.

Rank is also used to determine management privileges. The node shall not accept command-type S_MANAGEMENT_MSG_REQUEST primitives from a client with rank less than 15. The node shall accept request-type S_MANAGEMENT_MSG_REQUEST primitives (which cannot change the configuration of the node or subnetwork) from any client.

Priority can take a value in the range 0-15. The node “does its best” to service high priority U_PDUs before lower priority U_PDUs which are queued in the system. This means that the node is not required to *guarantee* that the higher priority U_PDUs will overtake all queued lower priority U_PDUs (depending on the implementation of the node, it may not be possible for a higher priority U_PDU to overtake a queued lower priority U_PDU which has entered an advanced stage of processing).

Client rank and the priority of data submitted by that client are not necessarily dependent.

H.6 Implementation notes for Data Rate Change Procedure

Although many modems that support the STANAG 4285 waveform are implemented so that the transmit and receive data rates must be the same at any instant, this is not an absolute limitation.

Some modems may be implemented so that they can use different transmit and receive data rates. Furthermore, the subnetwork, including the modem, may be implemented in such a way as to circumvent the problem if it exists, i.e., fast remote control of the modem (in a half-duplex system), or multiple modems (in a half or full duplex system). However, the DRC procedures defined in Annex C Section C.6.4.3 (Additional Scenarios) will support the case in which the system is limited by this constraint.

If a node uses D_PDU error statistics to make data rate adaptation decisions, the decision to change data rate should be made only based on a transmission containing DATA-ONLY or DATA_ACK D_PDUs; or after a transmission made up of only ACK_ONLY D_PDUs is received with errors. These conditions are imposed because it is difficult to make reliable DRC decisions based on analysis of short signals. While these conditions are not required for interoperability, they are required for reliable system operation. Experience with algorithms based on counting D_PDUs with errors shows that short transmissions are usually either received completely, with no errors, or not received at all (this is caused by the steep BER vs. E_b/N_o curves for the STANAG 4285 and MIL-110A modems). This can cause the data rate on an acknowledgement link, carrying only short transmissions, to increase steadily until, after one increase too many, the acknowledgements are no longer received. This is also the reason for transmitting ACK-ONLY D_PDUs multiple times when they are the only D_PDU type transmitted. This rule will generally lead to the ACK-ONLY D_PDUs being sent at a lower data rate, when data is flowing in (mainly) one direction on a link. In order to avoid the situation where the data transmissions are at 75 bps and the ACKs are at 2400 bps, the following rule should be followed: one end of the link shall transmit at not less than 1/8 the data rate of the other end.

If a node is only receiving acknowledgements, the transmit data rate for acks should be less than the transmit data rate for data but not less than 1/8 of the transmit data rate for data. The result of this rule will be that adaptive changes to data rate on one direction of the link will in some cases “pull” the data rate on the other direction of the link. This recommendation does not apply to a circuit that includes a shore answering frequency that must operate at a fixed data rate.

H.7 Optimum D_PDU Size

A number of studies have examined the impact of the size of D_PDUs (or HF frames) on the throughput of the ARQ protocol. Perhaps the most relevant of these studies to performance in the context of STANAG 5066 has been a study which combined on-the-air trials with OPNET modeling, done by DRA Portsdown. Other studies have been done by SHAPE Technical Center (STC) and by Rohde and Schwartz.

The first work in this area was done by STC in 1992 and documented in “Laboratory and Field Tests of the High Frequency OSI Data Link Protocol”, STC TN-506, August 1993. Some of the major findings in this report are[1]

1. the optimum frame size varies with the data rate and with channel conditions (fading as well as SNR)
2. throughput is not strongly sensitive to frame size

[1] One result from this report which is not relevant to the subject at hand but is more relevant to adaptive data rate control, is the very small improvement in throughput gained from switching from 1200 to 600 bps (see figures 24 and 25 of NC3A TM-937).

3. adaptive control of data rate with a compromise frame size of 200 bytes gives throughput very nearly identical to that realised with the “optimum” frame size for each data rate.

These results formed the basis of STC’s, and later NC3A’s, work with adaptive data rate HF ARQ protocols from 1992 until the present.

More recently, the DRA trials focused on the effect of varying error rate at 1200 bps. The DRA study has confirmed that throughput at a given BER is not strongly dependent on the frame size. At the higher BERs, there is a throughput maximum for frame sizes between 100 and 200 bytes. As the BER decreases, the frame size for maximum throughput also increases but very slowly, and the curve becomes even flatter.

The DRA throughput data are also valid for non-ARQ use of STANAG 5066 D_PDUs. The DRA data show the error free data bytes received, rather than the overall throughput; the small amount of overhead due to the short ARQ transmissions on a half-duplex link is not included. Thus, the 200 byte frame size is also a “good compromise” for use with non-ARQ 5066 transmission.

The DRA trials have confirmed (at least for 1200 bps) the following:
1. that the throughput is not strongly sensitive to frame size;
2. that 200 bytes is a good a ‘compromise’ selection for frame size.

If there is a desire to vary frame size in some way, STANAG 5066 supports the use of variable frame sizes. While the data available to date suggest that the benefit may be marginal, it would be possible, for example, to associate a certain frame size with each data rate.

The variable frame size scheme mentioned in the preceding paragraph brings with it a number of implementation issues (as does any variable frame size). One of these issues is the fact that, when a data rate change is made, there will generally be some number of D_PDUs queued for transmission. It will, in general, be advisable to transmit these (at least up to the end of the next C_PDU) without changing the frame size and start using a new frame size when the next D_PDUs are created. This avoids some difficult synchronisation issues, and the data available suggest that the performance penalty will be negligible.

H.8 Application Note: Use of STANAG 5066 in Broadcast Transmission Modes

Classical use of HF has allocated a single circuit (i.e, a frequency or set of frequencies) to each broadcast-traffic source. Typically, multiple frequencies are allocated to the circuit to increase the HF coverage area. When broadcasts from multiple sources are supported, the techniques typically used to share the circuit is frequency-division multiplexing (FDM) or time-division multiplexing (TDM), with FDM typically preferred because of the low throughput (nominally 75 bps) supportable on long-haul HF circuits. Even when higher throughputs are available, and TDM is a viable approach for transmission, the fixed-capacity allocations of TDM and FDM techniques can lead to inefficient circuit utilization and poor flexibility.

With its capabilities for higher throughput and group-addressing of U_PDUs, STANAG 5066 can support a broadcast of traffic from multiple sources. The paragraphs below review some of the implementation and efficiency issues that arise in using these capabilities of STANAG 5066. Some of the differences between an approach based on 5066 broadcast modes and the traditional, physically multiplexed (either FDM or TDM) multi-channel approaches are described.

It is assumed in the discussions below that an HF node which is handling multiple broadcast clients will be dedicated to this task. It would seem in general inappropriate (although perhaps not inconceivable) to have this node abandon multiple broadcasts in order to do some other task.

In order to support multiple simultaneous clients of the same type and SAP ID attached to the subnetwork a multi-user subnetwork client is required. This client will sit between a number of clients and the subnetwork interface and perform a number of functions, including multiplexing and segmentation. The arrangement is shown in Figure H-5.

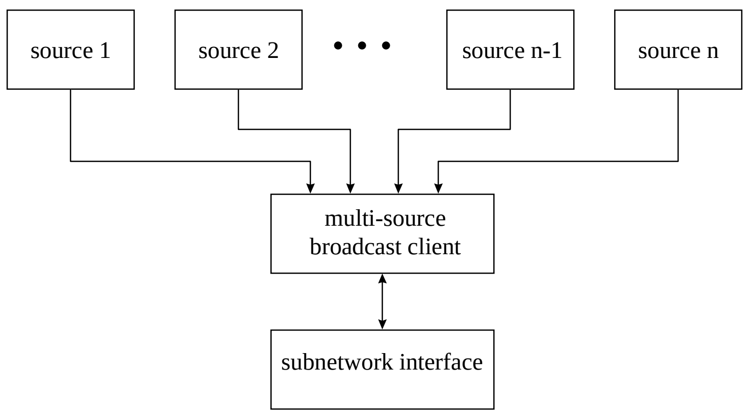

Figure H-5. Multi-source Broadcast (MSB) Client

Data will be accepted from the sources according to the interface spec of the source; the MSB client then forms a buffer between the standard interface to the 5066 subnetwork and the sources, which may be existing equipment.

Segmentation by the MSB client (or any other client) allows for more responsive and adaptive behaviour by the subnetwork. The client must segment large U_PDUs into segments (nominally 2 kb or smaller) and submit these segments in the appropriate primitive to the subnet interface sublayer. The reasoning behind the requirement for the client to segment large U_PDUs is provided below in conjunction with the summary of sublayer processing.

Insuring that each broadcast source gets appropriate access to the channel would seem to be a task appropriate to the channel access sublayer. However, since the MSB client is already responsible for queuing the data from the various sources, it knows the load that each source offers the subnetwork. The MSB client also knows the rank of the sources and the priority of the data. It therefore has all of the information required to allocate capacity to the various sources. Therefore, the MSB client should insure that U_PDUs or segments from multiple broadcast sources are passed down to the lower layers as is appropriate to

- the current (recent average) bandwidth of the source
- the rank of source
- the priority of the data from the source

At the receiving node, if a node is in more than one broadcast group, U_PDUs may (will!) arrive for various broadcast groups (perhaps composed by different broadcast compilers) and be delivered to the multi-source broadcast client. The MSB client will be able to identify the source of the arriving U_PDUs using the information in the S_UNIDATA_INDICATION primitives (sent to the client) which have the addressing information in them (S_PDUs do not). Exchanging this information between the Subnet Access Sublayer and Data Transfer Sublayer is an implementation issue (i.e., the definition of the internal primitives) which is not appropriate for a STANAG. Detailed guidance is available in NC3A TM-937.

Alternatively, the MSB client can use the Unreliable Datagram-Oriented Protocol (UDOP) defined in Annex F of this STANAG. UDOP supports source identification (through the connection identifier field of the UDOP PDU) and segmentation/re-assembly services (through the U_PDU segment identifier field of the UDOP). Yet another option would be the use of the Internet Protocol (IP) client type defined in Annex F of this STANAG in conjunction with the Internet User Datagram Protocol, since UDP/IP also supports source identification and segmentation and reassembly. Note that the subnetwork interface sublayer **does not segment** U_PDUs because re-assembly is not supported in the S_PDU format.

The channel access sublayer is simply a pipe for type 0 (data) S_PDUs, adding a few bits to convert them into type 0 (data) C_PDUs. The Data Transfer Sublayer will segment the arriving C_PDUs into D_PDUs and queue them for transmission. Note that, as noted in Annex H.7, the nominal optimum size of 200 bytes for HF transmission also applies to broadcast modes. So a C_PDU of 2 kb will be segmented into 10 D_PDUs, and queued for transmission (at 300 bps, these D_PDUs will take about 1 minute to transmit). The next C_PDU to arrive will be likewise segmented and queued.

With this summary of operation, the reason for limiting the size of U_PDUs, and thereby also S_PDUs, is clear. Very large U_PDUs, unsegmented by the user, can capture the channel. For example, if a 100 kb S_PDU (inside a C_PDU) were to arrive at the DT sublayer for ARQ transmission, then the DT sublayer would segment it into some 500 D_PDUs and queue them for transmission. If the transmit speed is 300 bps, it will be about 45 minutes before any other client has access to the channel. Depending on the implementation of the DT sublayer, this situation may or may not also arise if a large PDU arrives at the DT sublayer for non-ARQ service. Given that, even with data-rate adaptation, there is the potential for low-speed transmission over poor channels, the size of the maximum transmission unit (MTU) in the subnetwork should be made compatively small to avoid channel-capture. MTU sizes on the order of 1500 bytes (comparable to Ethernet) and smaller should be used.

The addressing structure adopted for non-ARQ (type 7) D_PDU allows some 268 million different group addresses. This large address space would be reduced by adoption of hierarchical addressing schemes for the subnetwork, be even so remains quite large. There remains the possibility to have a destination node belong to a number of broadcast groups depending on its operational role. The large address space, combined with the ability to belong to more than one broadcast group, offers the potential for significant efficiencies in broadcasts, because a close match between a group and the intended audience of a message is more likely. When a node can belong to only one group, one is forced to trade off between very large groups, with members receiving large amounts of unwanted messages; or smaller groups, with messages being retransmitted to a number of different groups. This addressing scheme also allows a broadcast compiler to send a message to a single destination node, by simply using that node’s individual address rather than a group address.

This approach is adaptive. If one of the broadcast sources is idle, or disconnects from the subnet, the capacity becomes available for other users (whereas in a TDMA or FDMA scheme it is more difficult or impossible to reallocate capacity). Because the allocation of capacity is done in software, it is flexible - it can take account of client rank, data priority, offered load, etc – and thereby use the HF channel more efficiently.