# E-MAIL STANDARDS FOR HF RADIO

Eric E. Johnson
Science Applications International Corp.

14 July 2000
(revised July 2001*)

## 1. INTRODUCTION

Electronic mail is particularly tolerant of the sometimes challenging HF radio communications medium. Its latency tolerance and store-and-forward delivery mechanism cope naturally with slow links and the occasional outage. This report documents the results of a study of the interoperability and performance of leading standards applicable to e-mail service over HF networks, and recommends specific protocol suites for next-generation networks as well as for backwards interoperability.

### 1.1 Background

The key challenges of the high-frequency (HF) radio medium that must be addressed in electronic mail (e-mail) and other data applications are listed below, along with closely corresponding mitigation approaches.

| Challenge | Response |
|---|---|
| ▪ Low signal-to-noise ratio (SNR) multipath fading channels | Robust modem waveforms, with forward error correction (FEC) and interleaving |
| ▪ Fades that overwhelm FEC | Automatic repeat request (ARQ) data link protocols |
| ▪ Propagation variation with hour, season, and sunspot cycle | Automatic link establishment (ALE) and related adaptive technologies |
| ▪ Limited channel capacity | Prioritization, flow control |

These techniques employed to overcome the challenges of the HF medium pose problems in turn for the transport and application layer protocols that convey e-mail through the network. In general, the HF subnetwork provides low bandwidth channels (due to the usual restriction to 3 kHz or narrower RF channels). Error-free data service comes only at the cost of delays that greatly exceed those experienced in the wired Internet. Beyond the delays imposed by low data bandwidths, the interleavers used to cope with burst errors result in link turnaround delays on the order of seconds to tens of seconds. This makes protocols that require frequent link turnarounds unattractive for HF applications.

### 1.2 Layers that must interoperate

Interoperability of HF e-mail systems requires interoperability of all of the communications layers and sublayers needed to convey messages. The relevant layers and sublayers include the following:

- Radios and frequency assignments
- Automatic link establishment
- Cryptographic algorithms and keys
- Data modem waveforms
- Data link protocol
- Transport protocol
- E-mail protocol

* Drafts of this report in 2000 overstated the performance of the STANAG 5066 ARQ protocol at +5 dB SNR.

The first three items are beyond the scope of this report. For the remainder of this report, it is assumed that a usable HF link is available, including communications security (COMSEC) when required. (Some timing aspects of COMSEC are addressed later.) The remaining four items are the subject of this investigation. The specific standards considered are enumerated in the next section.

### 1.3 Overview of the study

The purpose of this investigation was to evaluate the interoperability and performance of the leading open standards for HF e-mail. The standards considered are listed below by layer.

Data Modems

**MIL-STD-188-110B** contains the same waveforms as MIL-STD-188-110A and FED-STD-1052, the U.S. military and federal standards for fielded HF data modems. Draft NATO STANAG 4539 either refers to or incorporates the serial tone waveforms from MIL-STD-188-110B, including the new higher data rate waveforms (3200 – 9600 bps).

MIL-STD-188-141B Appendix C and NATO STANAG 4538 specify **burst waveforms** used for the third-generation ARQ protocols.

Data Link Protocols

**STANAG 5066** includes a selective repeat ARQ protocol that is planned for use in several near-term HF e-mail systems. It achieves its best ARQ performance with modem waveforms that "self identify" their data rate and interleaver. The MIL-STD-188-110B waveforms have this property.

MIL-STD-188-141B Appendix C and STANAG 4538 contain a third-generation ARQ (**3G ARQ**) protocol suite, including a Traffic Manager (TM) protocol, a Low-latency Data Link (LDL) ARQ protocol, and a High-throughput Data Link (HDL) ARQ protocol. These protocols are tightly coupled with the burst waveforms in the same documents.

Operation with **no data link protocol** is also considered.

Transport Protocols

The Transmission Control Protocol **TCP** is the default transport protocol for e-mail throughout the Internet. TCP provides end-to-end message integrity, and executes on the end-user computers. E-mail gateways may be placed at the boundaries of HF networks to intercept the end-to-end TCP handshakes, thereby eliminating some overhead traffic from the HF links.

The alternative considered is **no transport protocol** within the HF network. This is feasible when a data link protocol assumes the burden of message integrity within the HF network, and gateways intercept TCP at the HF network boundaries.

E-Mail Protocols

Only the protocols used to "push" e-mail messages through the network to the server nearest the recipient are considered here. (In many cases, the recipient then uses a distinct protocol such as POP3 or IMAP to "pull" messages from that server.)

The Simple Mail Transfer Protocol **SMTP** is the standard within the Internet. Its frequent link turnarounds are unattractive for networks with extended turnaround times, however.

Two HF Mail Transfer Protocols have been proposed, both called HMTP. Both are variants of SMTP that reduce the number of link turnarounds. STANAG 5066 Annex F describes a protocol here denoted **HMTP-66**. The protocol specified in MIL-STD-188-141B Appendix E is here denoted **HMTP-141.**

The scope of this investigation was limited to a desktop study, including simulation of many of the protocols and some probing of existing protocol implementations available via the Internet.

- Section 2 describes the implementations of the modems and data link protocols in a simulator.
- Section 3 discusses the interactions of TCP with HF data link protocols, and summarizes previous simulation studies of techniques to mitigate the resulting problems.
- Section 4 compares the three e-mail protocols, including analysis of their interoperability.
- Section 5 presents performance results from simulations of the various protocol combinations.
- Section 6 summarizes the results and recommends a specific approach to standardizing HF e-mail systems.

## 2. DATA LINK PROTOCOLS

The time available for this study allowed implementation of only two data link protocols:

- The STANAG 5066 Subnetwork Interface/Channel Access/Data Transfer protocol, operating in ARQ mode after data link setup. This suite is collectively denoted **5066-ARQ**
- A 3G ARQ subset including the Traffic Manager and the High-throughput and Low-latency Data Link protocols, **HDLV** and **LDLV**. The V suffix indicates that the TM protocol is used to vary the frame size

This section provides some details of the implementations of these protocols and their associated modems in the simulator used for this investigation.

### 2.1 Channel and modem models

A time-tested approach for successful performance evaluation projects is to determine at the outset what sort of graphs will be needed in the final report, and orient the effort specifically to producing the necessary data. In this case, the data required are the time to deliver e-mail messages as a function of SNR, using various combinations of protocols. The time to deliver a message depends on the channel principally in how many times each PDU must be sent to be correctly received.

The level of detail required in the HF channel model for this investigation is therefore satisfied if we know the probability of correct PDU reception as a function of waveform, PDU size, and SNR. Finer-grained fidelity in the specific order of good and bad frames is likely to disappear in the resulting total latency, although interactions with independent time-based processes such as TCP retransmissions require evaluation using simulation rather than a simple analytical model.

For the simulations in this investigation, measured performance of the MIL-STD-188-110B (75 through 9600 bps) and burst waveforms in Watterson-model 2-path fading channels was coded as tables of error rates versus SNR. Interpolation between measured SNR values was used as required. The MIL-STD-188-110B HDR waveforms used the 2.16s interleaver.

The SNR for each simulation was constant throughout the simulation. The Rayleigh fading effects on the modems were included in the modem measurements, but longer-term SNR variations were not simulated (as they would be in a NetSim simulation, for example).

### 2.2 5066-ARQ implementation

*Implementation Details*. Relevant portions of the 5066-ARQ protocol specified in STANAG 5066 Annexes A-C v1.1, dated 19 April 1999 were implemented in the simulated environment.

- At the start of each simulation, a C_PDU exchange using non-ARQ D_PDUs sets up the data link.
- For each simulation, the modem started at 300 bps, short interleaver. Dynamic data rate adaptation was implemented, using the self-identifying feature of the 110B waveforms. Timeouts during data link setup caused automatic reduction in the data rate through 150 and 75 bps, short interleaver, and finally to 75 bps, long interleaver if needed. After data link setup, data rate was adapted dynamically: when all D_PDUs in a multi-PDU transmission were acknowledged, the data rate for the subsequent transmission was doubled. When fewer than half of such D_PDUs were received correctly, the subsequent data rate was halved, and capped there for the rest of the simulation. (As SNR was constant, this avoided a detrimental “hunting” cycle.)
- Client datagrams (U_PDUs) were encapsulated in S_PDUs within C_PDUs within D_PDUs for transmission over the simulated modem and channel.
- The D_PDU address field was fixed at 3 octets (12 bit addresses for source and destination, comparable to the ARCS addresses used in LDLV).
- C_PDUs were segmented at 200-octet boundaries (100 octets for validation using reported 1999 prototype data)
- Selective acknowledgements and flow control were implemented as specified and/or recommended.
- Duplex operation was implemented (i.e., acks were carried with data). Acks are carried in every D_PDU after the data link is established.
- Ack-only PDUs are sent redundantly to fill the interleaver.

- 5066 ARQ features not required for this study (e.g., expedited or management PDUs) were not implemented.
- 110B modem processing delays were drawn from measurements reported in “STANAG 5066 Prototype Development and Testing : Results,” DERA CIS1.3 Portsdown West, UK. December 1998.
- Crypto/data link controller link turnaround time was modeled as a constant 1 second. (This is the lower bound on turnaround times specified in FED-STD-1052.)
- Timeout settings required some experimentation. The best approach seems to be tight timeouts for nodes sending data, loose timeouts for nodes sending only acknowledgements.
- The end of transmission mechanism was implemented, and was used to adjust timeouts at receiving nodes.
- Modem preamble detection was used to override timeouts likely to expire before the end of a PDU.

*Validation*. File transfer measurements from “STANAG 5066 Prototype Development and Testing : Results (Cont.),” DERA CIS1.3 Portsdown West, UK. December 1998 were used to check the simulator. The prototypes measured in the DERA report did not adapt data rate, used ~100-octet C_PDU segments, and used a simple file transfer protocol rather than one of the client protocols implemented in this study.

The first two differences from the 5066-ARQ simulator were accommodated directly by modifying the simulator used for validation.

The third was approximated by observing the simulator performance with an unacknowledged file transfer client, with and without TCP in the transport layer. Duplicate datagram filtering (see section 2.4) was disabled. The resulting simulator without TCP performed somewhat better than the measurements; with TCP it performed somewhat worse than the measurements. As the burden imposed by TCP is expected to be more onerous than that of the file transfer protocol used in the DERA measurements, this result lends plausibility (though not validation) to the 5066-ARQ simulator results.

## 2.3 3G ARQ implementation

*Implementation Details.* The TM, HDL, and LDL protocols specified in STANAG 4538 and MIL-STD-188-141B Appendix C were implemented as described below.

- Timings and timeouts are as specified in STANAG 4538.
- In LDLV, only four BW3 frame sizes were implemented: 64, 128, 256, and 512 octets.
- Each change of direction of client data flow on the link is followed by a TM handshake that synchronizes the nodes and announces the size of the data payload in the BW3 bursts to follow.
- Multiple datagrams can be sent in the same direction without additional TM handshakes if they use the same frame size. The node sending data can change frame sizes between datagrams without relinquishing the link by initiating a new TM handshake. It releases the link by sending redundant EOMs.

*Validation*. Measurements of a similarly capable LDL implementation were provided by Harris RF Communications. The measured performance of e-mail delivery and pings was well within the 90% confidence interval of the simulated performance of similar operations.

## 2.4 Optimizations

A key optimization implemented in both of these data link protocols was a simple filtering of duplicate client datagrams. Each time a client (e.g., TCP) submitted a datagram for transmission, the data link protocol compared it to queued datagrams awaiting transmission, and silently discarded it if it was identical. When the ARQ protocol was operating in “greedy” mode, i.e., sending all waiting datagrams at one station before reversing the direction of client data flow, client datagrams were also compared to all datagrams sent since the other station had a chance to send client acknowledgements.

Although a few duplicate datagrams were accepted that an algorithm with a complete history of traffic would have rejected, this simple technique was quite effective in eliminating many of the unnecessary retransmissions sent by TCP as it “learned” the network latency.

With this optimization, the LDLV protocol implementation was just a bit slower than the well-seasoned Harris implementation, while the 5066 ARQ implementation far outperformed the prototypes tested by DERA. It seems safe to assume that more recent implementations of the 5066 ARQ probably perform similarly to the simulator.

## 3. TRANSPORT LAYER INTERACTIONS

TCP provides end-to-end reliable delivery of application data using a sliding window ARQ protocol with adaptive timeouts and window size. It is well known that the adaptive algorithms used in the most widely implemented versions of TCP are oriented to congestion control in a reliable network. As a result, they “back off” very quickly after datagram timeouts expire, and are widely believed to be unsuitable for use over wireless links.

The earliest version of the simulator used in the present study was developed to identify approaches to successfully employ TCP on lossy links such as the HF medium. Concisely, the results are as follows:

- If TCP is used in HF applications without an error-correcting data link protocol, TCP will be responsible for error correction, so its maximum timeout should be set to a value of 30 to 60 seconds.
- If an appropriate data link protocol is used to correct errors, the TCP maximum timeout should be set to a large value, effectively eliminating TCP retransmissions after TCP learns the round trip time statistics of the network.

The simulation results in section 5 suggest that TCP can indeed be used successfully in wireless applications, although the overall speed of an optimized protocol suite without TCP will generally be better than with TCP.

## 4. E-MAIL PROTOCOLS

The three e-mail protocols, SMTP, HMTP-66, and HMTP-141, employ nearly identical messages and sequences to convey a message from one machine (the client) to another (the server). Note that the server in one session may become the client in a subsequent session as it forwards a message to a machine closer to the ultimate recipient.

A typical exchange of SMTP messages is depicted below. SMTP (Internet Standard 10, RFC 821) requires that this lockstep exchange of messages from the client and acknowledgements from the server proceed as shown, resulting in 12 link turnarounds. The HMTP variants allow varying degrees of grouping of the commands and responses to reduce the number of link turnarounds.

| **Server** | **Client** |
|---|---|
| `220 server.goodguys.gov ESMTP Sendmail ready` | |
| | `HELO myhost.mycompany.com` |
| `250 server.goodguys.gov` | |
| | `MAIL From:<myname@mycompany.com>` |
| `250 <myname@mycompany.com>... Sender ok` | |
| | `RCPT To:< recipient@somewhere.mil >` |
| `250 <recipient@somewhere.mil>... Recipient ok` | |
| | `DATA` |
| `354 Enter mail, end with "." on a line by itself` | |
| | `Message, message, message`<br>`.` |
| `250 Message accepted for delivery` | |
| | `QUIT` |
| `221 server.goodguys.gov closing connection` | |

- The 220 response from the server responds to session establishment (e.g., opening a TCP socket).
- The HELO command from the client identifies the client machine.
- 250 responses from the server are positive acknowledgements.
- The MAIL command from the client identifies the sender of the message.
- RCPT commands from the client indicate intended message recipients. At least one RCPT command must elicit a 250 (positive) response if the protocol is to proceed to delivering the message to the server.
- The DATA command from the client indicates its intent to begin sending the e-mail message to the server.
- The QUIT command from the client requests session termination.
- The 221 response from the server confirms session termination.

This sample session, with minor variations, will be used to illustrate the differences among SMTP, HMTP-66 and HMTP-141 in section 4.3.

## 4.1 SMTP service extensions

Many of the capabilities of the current Internet e-mail service were not included in RFC 821, but have been added through a Service Extensions mechanism, specified in RFC 1869. One of the most used of these allows the attachment of binary documents to e-mail messages using the Multipurpose Internet Mail Extensions (MIME).

A less-known extension allows the command and response grouping that we desire for improved HF e-mail speed. This extension is officially known as Command Pipelining, and is specified in RFC 2197 (originally specified in RFC 1854).

SMTP with service extensions is known as ESMTP. Note that the 220 response in the example SMTP session above indicated support for ESMTP. A client that wishes to use particular service extensions opens the session with an EHLO command (rather than HELO). This prompts the server to respond with a series of 250 responses, each of which identifies a supported extension using an assigned keyword (such as 8BITMIME or PIPELINING).

The example shown below illustrates the EHLO command and response, including the technique used to indicate whether each 250 response is the last in the list: if the 250 is followed by a – character, this is not the final response; in the final response, the 250 is followed by a space character.

| **Server** | **Client** |
|---|---|
| 220 server.goodguys.gov ESMTP Sendmail ready | |
| | EHLO myhost.mycompany.com |
| 250- server.goodguys.gov<br>250-8BITMIME<br>250-SIZE<br>250 PIPELINING | |
| | MAIL From:<myname@mycompany.com> |

## 4.2 Command pipelining

Clients employ the command pipelining mechanism as follows (quoting from RFC 2197):

> "When a client SMTP wishes to employ command pipelining, it first issues the EHLO command to the server SMTP. If the server SMTP responds with code 250 to the EHLO command, and the response includes the EHLO keyword value PIPELINING, then the server SMTP has indicated that it can accommodate SMTP command pipelining."
>
> "Once the client SMTP has confirmed that support exists for the pipelining extension, the client SMTP may then elect to transmit groups of SMTP commands in batches without waiting for a response to each individual command. In particular, the commands RSET, MAIL FROM, SEND FROM, SOML FROM, SAML FROM, and RCPT TO can all appear anywhere in a pipelined command group. The EHLO, DATA, VRFY, EXPN, TURN, QUIT, and NOOP commands can only appear as the last command in a group since their success or failure produces a change of state which the client SMTP must accommodate. (NOOP is included in this group so it can be used as a synchronization point.)"

**HMTP-141** specifies use of command pipelining just as stated above.

**HMTP-66** goes a step further (quoting from STANAG 5066 Annex F):

> "HMTP combines the MAIL command, RCPT command, and the mail data for multiple mail messages into a single transmission to reduce the delays associated with link turnaround associated with the dialog between HMTP-client and HMTP server. SMTP service extensions for Command-Pipelining perform a similar grouping of SMTP commands, but to remain backwards-compatible with SMTP servers that do not implement Command Pipelining, the service extensions do not start grouping commands until an initial handshake is completed between SMTP client and server. The HMTP protocol groups all commands to provide the greatest efficiency, but consequently is not interoperable with SMTP servers."

Both of the HMTP variants require fall-back to basic lockstep SMTP operation if their flavor of pipelining is not supported.

### 4.3 Comparison of e-mail protocols

The table below illustrates the key differences among our three e-mail protocols in messages and their sequences *as seen on a link*. As explained later, the order of events *within the computers* is affected by buffering.

| SMTP | | HMTP-141 | | HMTP-66 | |
|---|---|---|---|---|---|
| Server | Client | Server | Client | Server | Client |
| 220 | | 220 | | 220 | |
| | HELO | | EHLO | | EHLO |
| 250 | | 250* | | | MAIL From |
| | MAIL From | | MAIL From | | RCPT To |
| 250 | | | RCPT To | | DATA |
| | RCPT To | | DATA | | Message |
| 250 | | 250 | | | . |
| | DATA | 250 | | | QUIT |
| 354 | | 354 | | 250* | |
| | Message | | Message | 250 | |
| | . | | . | 250 | |
| 250 | | | QUIT | 354 | |
| | QUIT | 250 | | 250 | |
| 221 | | 221 | | 221 | |
| **Link turnarounds** | **12** | | **6** | | **2** |

Notes: 1. Only the first few characters of the commands and responses are shown.
2. The notation 250* indicates a variable-length response containing a list of EHLO keywords.

At the server, there is no difference in the processing of SMTP commands among the three protocols (except that SMTP rejects EHLO commands). Any size batch of commands from the client is held in a transport layer buffer at the server, and any SMTP or HMTP server application reads and responds to one command at a time. Thus, a server that supports any of the three protocols *should* be able to process batches of commands.

Each command generates an immediate response, but these responses may not be sent immediately to the transport layer. The differences among the three protocols at the server lie in the points in the handshake at which the server delivers buffered responses to the transport layer.

- SMTP sends every response as it is generated.
- HMTP-141 buffers responses until it responds to a EHLO, DATA, VRFY, EXPN, TURN, QUIT, or NOOP command, at which point all buffered responses are sent.
- HMTP-66 buffers all responses for all mail messages, sending them in a batch in response to a QUIT command just before ending the connection.

### 4.4 Interoperability

The three e-mail protocols are intended to be interoperable. However, implementation flaws in the various e-mail servers in the world have resulted in subtle incompatibilities. The following description is from RFC 2197.

> "In the best of all worlds it would be possible to simply deploy SMTP client software that makes use of command pipelining: batching up multiple commands into single TCP send operations. Unfortunately, the original SMTP specification [RFC 821] did not explicitly state that SMTP servers must support this. As a result a non-trivial number of Internet SMTP servers cannot adequately handle command pipelining. Flaws known to exist in deployed servers include:
>
> (1) Connection handoff and buffer flushes in the middle of the SMTP dialogue. Creation of server processes for incoming SMTP connections is a useful, obvious, and harmless implementation technique. However, some SMTP servers defer process forking and connection handoff until some intermediate point in the SMTP dialogue. When this is done material read from the TCP connection and kept in process buffers can be lost.
>
> (2) Flushing the TCP input buffer when an SMTP command fails. SMTP commands often fail but there is no reason to flush the TCP input buffer when this happens. Nevertheless, some SMTP servers do this.
>
> (3) Improper processing and promulgation of SMTP command failures. For example, some SMTP servers will refuse to accept a DATA command if the last RCPT TO command fails, paying no attention to the success or failure of prior RCPT TO command results. Other servers will accept a DATA command even when all previous RCPT TO commands have failed. Although it is possible to accommodate this sort of behavior in a client that employs command pipelining, it does complicate the construction of the client unnecessarily."

A server that includes the keyword PIPELINING in its response to the EHLO command is merely stating that it abides by a list of nine rules that ensure a reliable implementation of the basic SMTP protocol. Of particular interest to us are two of these rules: it must not lose buffered incoming commands in its transport layer queue, and it must send all buffered responses whenever its transport layer queue of incoming commands is emptied. The former rule ensures that such servers will correctly process arbitrarily long batches of commands. The latter rule ensures that responses to a batch of commands will always be sent after the end of the batch, no matter how short the batch; this ensures backward interoperability with SMTP clients.

Because MIL-STD-188-141B Appendix E mandates compliance with the PIPELINING service extension, HMTP-141 servers will be interoperable with any client. If we assume that HMTP-66 servers comply with the rules in RFC 2197 (implied but not mandated in STANAG 5066 Annex F), we can summarize the interoperability of the three protocols as shown below. All of the protocol combinations eventually transfer the message.

<table>
<tr><th rowspan="2">CLIENT</th><th colspan="3">SERVER</th></tr>
<tr><th>SMTP</th><th>HMTP-141</th><th>HMTP-66</th></tr>
<tr><td>SMTP</td><td colspan="3">HELO followed by normal SMTP transfer: 12 link turnarounds.<br>Message sent once.</td></tr>
<tr><td>HMTP-141</td><td>EHLO rejected. Fall back to normal SMTP transfer after EHLO command. 14 link turnarounds.<br>Message sent once.</td><td colspan="2">EHLO accepted. Pipelined transfer with 6 link turnarounds.<br>Message sent once.</td></tr>
<tr><td>HMTP-66</td><td>EHLO rejected. Fall back to normal SMTP transfer after complete HMTP-66 transmission. 14 link turnarounds.<br>Message sent twice.</td><td>EHLO accepted. On-air appears to be HMTP-66 transfer with 2 link turnarounds: the intermediate response batches are queued in the server transport layer.<br>Message sent once.</td><td>Normal HMTP-66 transfer with 2 link turnarounds.<br>Message sent once.</td></tr>
</table>

### 4.5 Support for SMTP pipelining

HMTP-141 trades away the performance benefit of reducing link turnarounds to the minimum in return for full compliance with Internet standards. Is this a worthwhile exchange?

One benefit of compliance with Internet standards is that direct interoperation of HF clients with the existing Internet infrastructure is possible. The previous sections have shown that (yet to be quantified) performance gains are possible when sending e-mail over HF links to a server that supports pipelining. To the extent that Internet servers support pipelining, HF clients could achieve this performance gain without requiring dedicated HF e-mail gateways. Thus, we need to know the degree of support for pipelining in the existing e-mail infrastructure.

The results of an informal sampling of e-mail servers are shown below. Several additional servers were investigated but they lacked any ESMTP support.

| Server | PIPELINING | EXPN | VERB | 8BITMIME | SIZE | DSN | ONEX | ETRN | XUSR | HELP | AUTH |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ***.NMSU.Edu | | • | • | • | 5M | • | • | • | • | • | |
| ***.NMSU.Edu | | | | • | • | • | • | | • | • | |
| ***.ieee.org | | | | • | • | • | • | | • | • | |
| ***.scott.af.mil | | • | • | • | • | • | • | • | • | • | |
| ***.dera.gov.uk | • | | | • | | | | | | | |
| ***.defence.gov.au | | | | • | • | | • | • | • | • | |
| ***.harris.com | | • | • | • | • | • | • | • | • | • | |
| ***.thomson-csf.com | • | • | | • | 5M | • | | • | | • | • |

The results of this survey suggest that pipelining is not widely supported: only 25% of the servers supporting ESMTP responded with the EHLO keyword PIPELINING. Although many e-mail servers may nevertheless correctly handle command batches, it appears that dedicated HF e-mail gateways will be needed in most cases.

Another informal survey was undertaken to determine typical path lengths (i.e., number of relay servers) for e-mail in the existing Internet. Universal practice (at least among e-mail received from participants in HF standards development) is that each organization maintains an e-mail gateway/firewall between individual workstations within the organization and the external Internet. Active probing determined that, in many cases, separate gateways are maintained for outgoing and incoming messages. Thus, the need to maintain a gateway at the boundary of an HF radio subnetwork is not at all a departure from current practice.

## 5. PERFORMANCE

At this point, we have determined that the various e-mail protocols are, in fact, interoperable, though with perhaps wide variations in the time required to deliver a message through an HF radio network. We now explore the performance of the selected combinations of e-mail, transport, and data link protocols. As noted before, the complex interactions among the protocols preclude use of simple analytical models. Resource constraints made it infeasible to pursue a program of measured performance, leaving a simulation study as the only viable option.

Implementation of the e-mail protocols in the simulator was straightforward. In each case, scripts for the client and server were prepared, and the e-mail protocols simply stepped through them as the appropriate messages were received. Each simulation began with a call by the client to open a connection to the server. The server responded with a 220 message (see section 4), and a dialog similar to the example in section 4 was conducted over the transport connection to transfer a single 5000 octet e-mail message. (Analysis of 311 e-mail messages recently received via the Internet showed a median size of 3618 octets. This was close to the 5000 octet size used in the testing of prototype 5066 ARQ systems, so the latter value was used here as well.)

When TCP provided the transport service, it opened the connection using the usual SYN and SYN/ACK handshake. The default TCP timeout of 500 ms expired repeatedly during this initial handshake; each timeout resulted in a doubling of the next timeout duration. (This, and other details of the adaptive algorithms in TCP are nicely described in *Internetworking with TCP/IP* volume I, by D.E. Comer, starting in section 13.15.) The TCP simulator fully implements sliding window flow control (including avoidance of the fabled Silly Window Syndrome), ARQ, segmentation and reassembly, and selectable maximum segment size and timeout upper bound. The simulations in this section all used a maximum TCP segment size of 1460 octets, corresponding to the usual Ethernet maximum transmission unit of 1500 octets. Unless otherwise noted, the maximum TCP timeout was set to one hour.

The simulations cover an SNR range from –10 to +30 dB. The lower limit is set by the most robust waveform evaluated (the 3G LDL burst waveform). The upper limit is set by radio receiver performance (e.g., the harmonic distortion specification in MIL-STD-188-141B).

The result of each simulation is the time required to deliver a single message using the selected protocol suite. The average latency for several independent simulations is converted to "messages per hour," effectively assuming that a new message transfer begins immediately after the previous message is delivered. Note that the message throughput calculated in this way is extremely sensitive to message size.

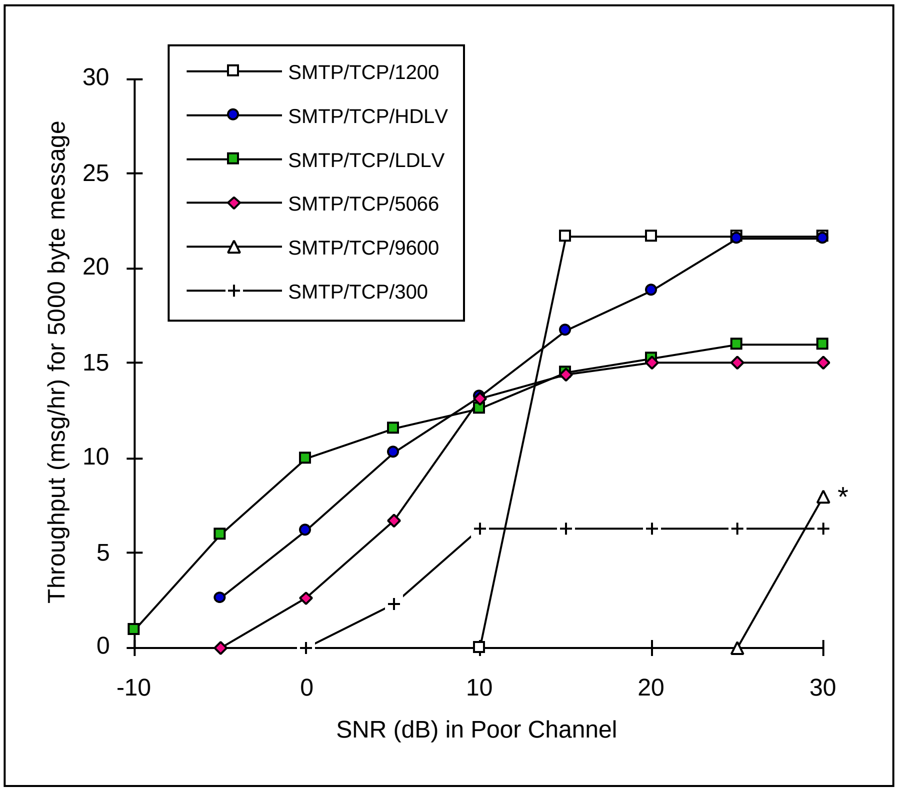
The charts in the following sections show performance including all protocol overheads (except for ALE), and cannot be compared to charts that do not include all of the protocols necessary to transfer e-mail over the HF channel, or to results for different message sizes.

### 5.1 SMTP with TCP: HF as a transparent subnetwork

As a starting point for our investigation of e-mail performance in HF networks, let us consider the e-mail protocol structure least attuned to the HF medium: end-to-end SMTP and TCP, using an HF subnetwork as just another subnet in the Internet. We examine three cases:

- SMTP and TCP using only a 110B modem with no data link ARQ protocol
- SMTP and TCP using 5066 ARQ with a 110B modem (75 – 9600 bps)
- SMTP and TCP using 3G ARQ

The system using a 110B modem with no data link protocol is presumed to have no adaptive data rate mechanism, and simply runs the modems at a fixed data rate, presuming no knowledge of channel conditions. Three rates were simulated: 9600 bps, 1200 bps, and 300 bps. The ARQ protocols also have no initial knowledge of the channel, but adapt their data rates as described in the respective standards. As discussed earlier, the 5066 ARQ protocol starts at 300 bps and adapts its rate up or down based on experience with the channel. The 3G ARQ protocols compute frame size considering only the message size; data rate adaptation occurs as a result of code combining.




* The throughput for 9600 bps at 30 dB reflects an artificial reduction of the TCP maximum timeout to 30 seconds. Without this modification, throughput was reduced to less than 1 msg/hr by the slow retransmission rate.

Several aspects of the SMTP/TCP results are of interest.

- The unaided modems performed quite well when given a channel that resulted in very low error rates. When the BER exceeded $10^{-4}$, however, it was necessary to modify the TCP timeout limit so that TCP would act as an adequate ARQ mechanism. As the BER approached $10^{-3}$, normal-size TCP segments experienced nearly 100% frame error rates, and no throughput was possible.

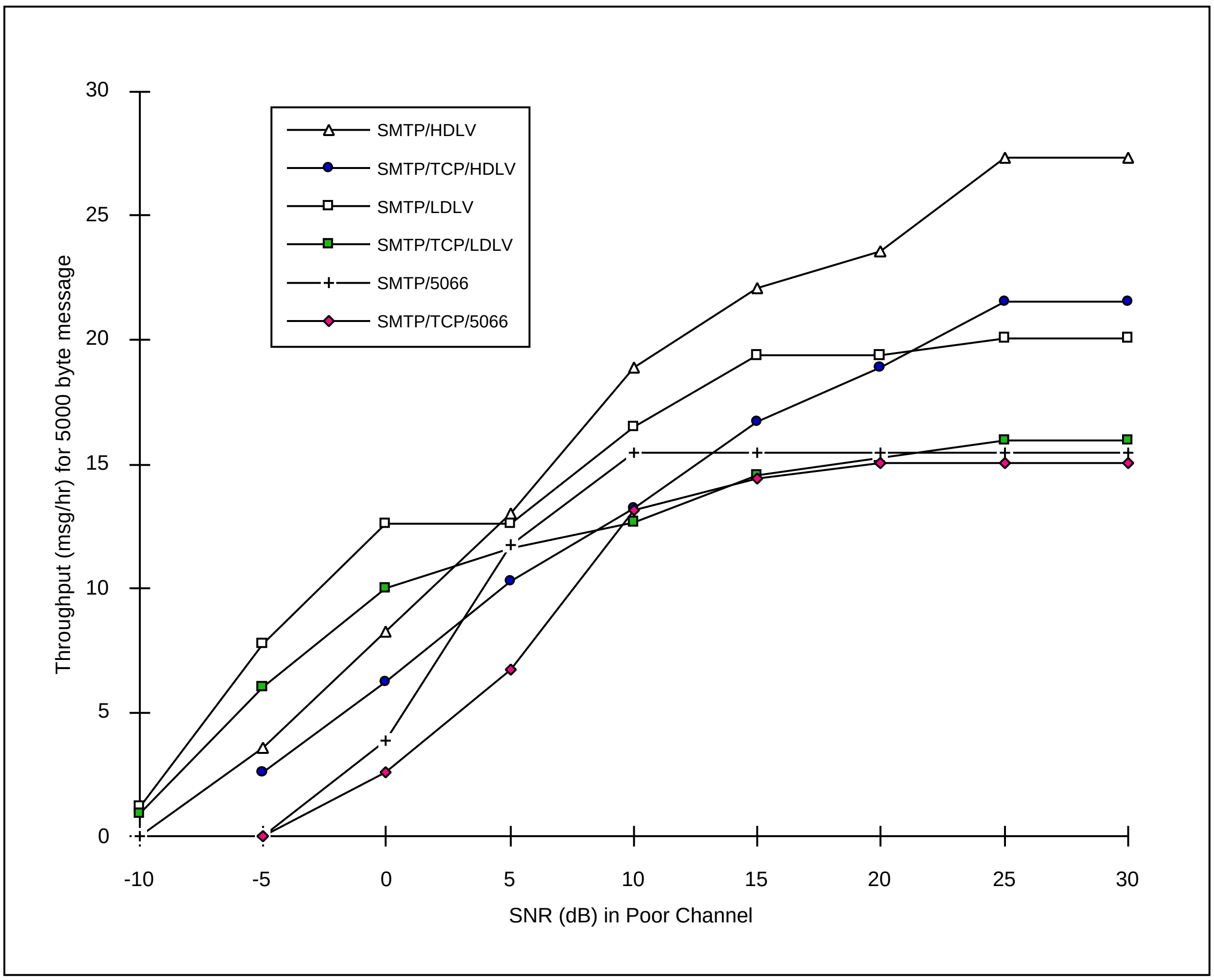
- The ARQ protocols coped with the entire range of SNR values by adapting their modem data rates, and by using smaller frame sizes than the TCP segment size.
- 5066 ARQ was unable to make effective use of its 9600 bps modem at high SNR because of the number of link turnarounds required to boost the modem data rate from its default initial value to the best rate the channel could support. It also suffers from sequence number starvation at the higher data rates due to the fixed frame size.
- The curious dip in the LDLV curve at +5 dB SNR seems to result from a threshold in its code combining ARQ. This anomaly is even more pronounced in later results.

## 5.2 SMTP without TCP

In order to isolate the effects of the various protocols on email system performance, we next eliminate TCP and run SMTP as a direct client of the data link ARQ protocols. (Use of SMTP with only a 110B modem is not feasible over the SNR range considered, due to the lack of error correction in the resulting system.)

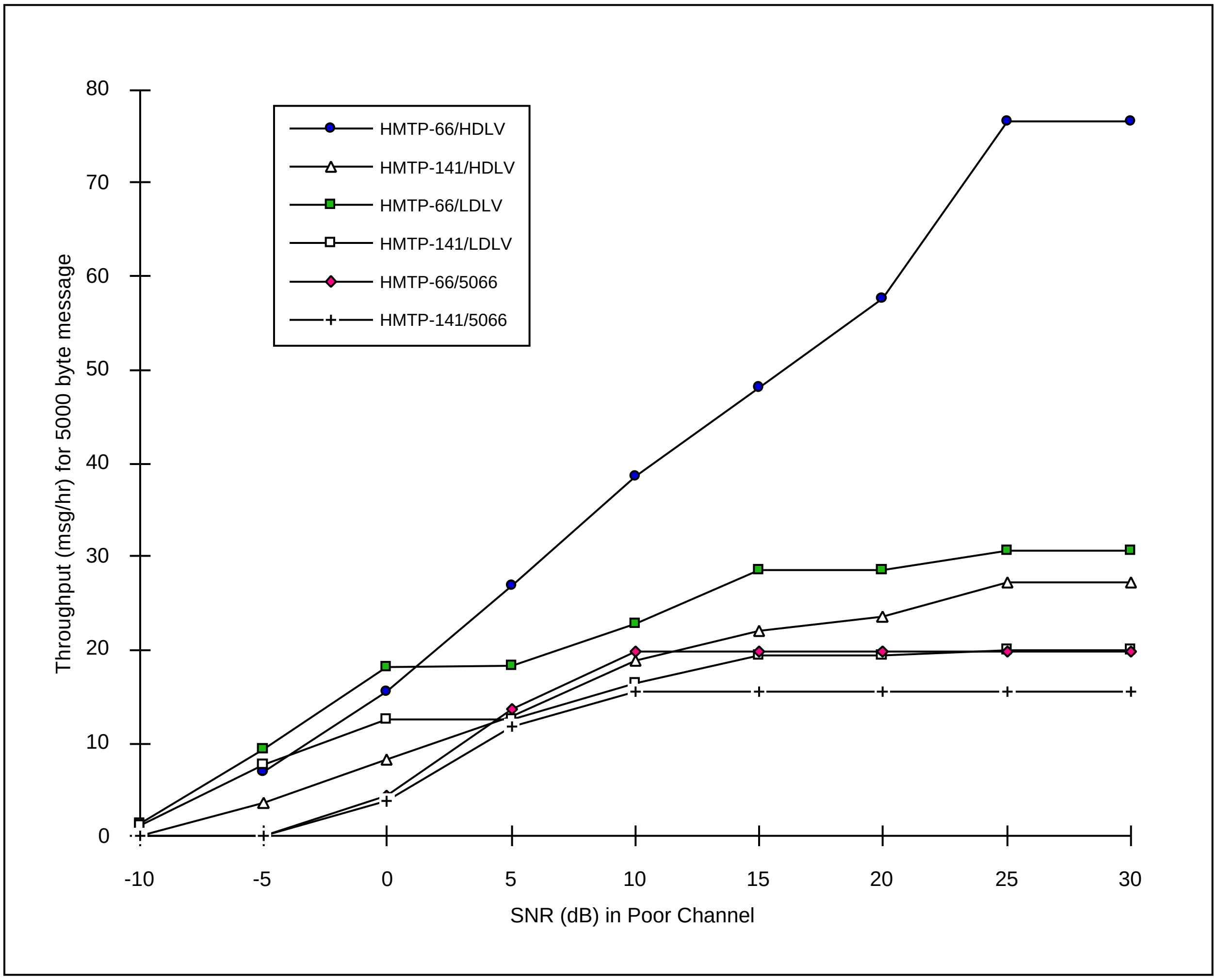
The throughput results for SMTP with and without TCP are plotted below. TCP clearly reduced message throughput somewhat, but the performance gained by employing HF gateways to eliminate TCP was typically only 25-30%. The extra message latency that TCP introduces may be acceptable in some applications which do not otherwise require a gateway at the interface with the Internet.

Again we see the plateau in 5066 ARQ performance. This suggests that setting the initial data rate and frame size to match the channel would be of great benefit, if the channel could be characterized reliably *a priori*.



## 5.3 HMTP without TCP

Finally, we examine the two HMTP variants. Because they will normally operate with dedicated gateways, they were simulated without TCP. As expected, the fully pipelined protocol specified in STANAG 5066 Annex F provides faster message delivery than the partially pipelined version in MIL-STD-188-141B Appendix E.



# 6. RECOMMENDATIONS

## 6.1 Summary of technical findings

This study investigated the interoperability and performance of a small number of candidate modems and protocols for use in HF e-mail systems. Interoperability was evaluated through study and analysis of the relevant standards. Performance was studied using a simulator created for this project. The simulator was checked using a few measurements of similar systems.

The findings of this study are as follows:

- Any e-mail server software that follows the rules for SMTP command pipelining (RFC 2197 or RFC 1854) will be interoperable with all of the e-mail protocols studied. HMTP as specified in MIL-STD-188-141B mandates compliance with RFC 1854. Compliance with these RFC 2197 is implied in STANAG 5066 Annex F, but does not appear to be explicitly required.
- E-mail client software may pipeline commands to essentially any extent when working with a pipelining-capable server as long as the client correctly deals with the responses.
- E-mail clients must fall back to standard SMTP operation when working with a server that cannot support pipelining so that universal interoperability is maintained. Sending a batch of SMTP commands that includes a message of significant size can result in sending that message twice if the server is not first interrogated for support of pipelining.
- Limiting pipelining to the extent allowed in the Internet standards results in reduced performance with apparently little gain in access to the Internet infrastructure. An informal survey showed that perhaps 25% of e-mail servers advertise support for Internet-standard pipelining.
- The Internet Transmission Control Protocol TCP can be used transparently over HF radio subnetworks, although its presence on HF links does reduce overall performance. The link layer ARQ protocols studied (3G ARQ and 5066 ARQ) support TCP operation over their respective ranges of channel conditions without requiring operator interaction (e.g., to specially tune TCP parameters).
- When channel conditions are not known *a priori*, the 3G ARQ protocols can provide significantly higher performance than the STANAG 5066 ARQ protocol in both throughput and SNR range.
- The 5066 ARQ suite offers a higher-speed modem than 3G ARQ, and an ability to carry client data in both directions during a session. Its high-SNR performance was limited here because it must "adapt up" to the channel from its initial settings while 3G instead "adapts down" via code combining in low SNR links.

## 6.2 Recommendations for HF e-mail standards

These results suggest the following program of evolution of the HF e-mail standards and technology:

- Little appears to be gained by maintaining two slightly different HF e-mail delivery protocols under the name HMTP. The higher performance of the HMTP defined in STANAG 5066 Annex F should prevail, and the U.S. MIL-STD-188-141B Appendix E should be revised to allow full pipelining of SMTP commands while continuing to require explicitly that servers comply with the command pipelining rules in RFC 2197. Implementers would of course remain free to perform a non-pipelined EHLO handshake when pipelining support by a distant server was uncertain.
- HF e-mail systems in challenging environments should employ the 3G ARQ suite for the higher performance and increased robustness that it provides.
- Continued development of the 3G ARQ protocol suite should address duplex flow of higher-layer data and acknowledgements, code combining using a dense constellation (e.g., the 12,800 bps waveform from MIL-STD-188-110B), and multiple channel operation.
- STANAG 5066 ARQ protocol performance could improve dramatically in high-SNR channels if its initial data rate and frame size are matched to the channel. This should be straightforward in applications with slowly-varying channels. Techniques to accurately characterize more challenging channels in advance should also be investigated.