Project Report
NASA/G-1

Issues Involved in the Development of an Open Standard for Data Link of Aviation Weather Information

R.D. Grappel

14 July 2000

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

Prepared for the National Aeronautics and Space Administration
John H. Glenn Research Center, Lewis Field, Cleveland, OH 44135

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.
NASA/G-l

2. Government Accession No.

3. Recipients Catalog No.

4. Title and Subtitle
Issues Involved in the Development of an Open Standard for Data Link of Aviation Weather Information

5. Report Date
14 July 2000

6. Performing Organization Code

7. Author(s)
R.D. Grappel

8. Performing Organization Report No.
NASA/G-l

9. Performing Organization Name and Address
MIT Lincoln Laboratory
244 Wood Street
Lexington, MA 02420-9108

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
NASA Glenn

2. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, OH 44135

13. Type of Report and Period Covered
Project Report

14. Sponsoring Agency Code

5. Supplementary Notes

This report is based on studies performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, under Air Force Contract F19628-95-C-0002.

6. Abstract

This paper describes how an effective and efficient data link system for the dissemination of aviation weather information could be constructed. The system is built upon existing "open standard" foundations drawn from current aviation and computer technologies. Issues of communications protocols and application data formats are discussed. The proposed aviation weather data link system is independent of the actual link mechanism selected.

17. Key Words

Open System	TCP/IP
Open Standard	BUFR
Data Link	GRIB
Communications Protocol	AWIN
ATN	

18. Distribution Statement

This document is available to the public through the National Technical Information Service, Springfield, VA 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	49	

FORM DOT F 1700.7 (8-72) Reproduction of completed page authorized

ABSTRACT

This paper describes how an effective and efficient data link system for the dissemination of aviation weather information could be constructed. The system is built upon existing "open standard" foundations drawn from current aviation and computer technologies. Issues of communications protocols and application data formats are discussed. The proposed aviation weather data link system is independent of the actual link mechanism selected.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

Maintaining an accurate and up-to-date weather database is vital for any pilot's situational awareness. The requirements of weather knowledge include long-range strategic decision-making (for example, should the planned flight route be altered, or even attempted), as well as tactical avoidance of nearby hazardous weather conditions. A balance must be struck between providing too little information (leaving the pilot to blunder into dangerous situations) and too much information (confusing the pilot and obscuring the important data amidst extraneous detail). Often there is insufficient weather-gathering equipment onboard to provide the pilot with independent weather information -- typical general aviation systems may have only a voice radio and, possibly a lightning detector. Larger aircraft with a weather radar can get more weather information, but still lack the "big picture" -- and the interpretation of an onboard weather radar image can be difficult. The presence and location of hazardous aviation weather conditions (turbulence, hail, icing, etc.) often must be inferred by the pilot from multiple sources.

Meanwhile, networks of ground and satellite weather sensors are being built and deployed. An abundance of weather information is available -- but there is little means to transfer the information from the sensors to the aircraft where it is actually needed. Current aviation operations mainly depend on voice radio communications between a controller (who has access to the weather inforrnation) and the pilot. It is easy to imagine how difficult it might be (and how laborious and error-prone) for the controller to attempt to communicate complex weather information over this low-bandwidth "link". To quote the common wisdom: "A picture is worth a thousand words!"

Also, nationwide (and even worldwide) systems of digital communications are going into place. Systems such as the Internet and cellular phones routinely provide large-scale digital data links. The cost and availability of reasonable-speed digital data links have reached quite reasonable levels for incorporation into aviation systems, even at the low end of the general aviation spectrum. Communications standards incorporating the features necessary for aviation data links have been developed and approved.

This paper will indicate how an effective and efficient data link system for aviation weather dissemination might be constructed. The system is built upon existing foundations, drawn both from the current aviation and computer-data research communities. Issues of communications protocols and application data formats will be discussed. The proposed system will be independent of the actual link mechanism chosen. "Open Standards" will be used as much as is possible.

2. Data Link Protocols

One major area of required standardization for a weather-information data link is the set of communications protocols to be employed. This section will discuss a set of requirements and desired capabilities for the weather data link. The protocols used by several existing or experimental datalink systems will be surveyed. It should be noted that none of these meets the entire desired set of capabilities. Finally, a communications protocol that does meet nearly all the requirements/capabilities will be discussed, along with ways to fully realize the desired functionality.

It should be noted here that this section deals primarily with what is termed the "data link layer" and the "network layer" communications protocols. Communications protocols are defined in terms of multiple layers, each layer encapsulating a certain set of additional functions that depend upon the lower layers for their functions. The lowest layer in the "stack" is typically termed the "physical layer". The physical layer deals with the details of moving a bit or byte across the specific physical interface between systems. Above the physical layer is the "data link layer". The data link layer deals with communicating a given sequence of bits (or bytes) across the physical interface. Both the physical and data link layers are usually specific to a given communications interface. Above the data link layer is the "network layer". The network layer deals with communications of messages (data packets) over the data link layer. The network layer is usually independent of the communications medium in use (the lower layers isolate these dependencies). Issues such as error handling, message segmentation, routing, etc. are typical functions of the network layer. Higher layers in the communications protocol "stack" deal with more application-oriented issues such as data representation/compression, network security, etc.

The architecture of the weather datalink system starts with the definition of various data "subnetworks". These subnetworks combine the physical and data link layer functions and provide the means to transfer bits across the air-ground interface. There are a number of existing aviation subnetworks already defined and in service – including Mode S, VHF, HF, and Satcomm. These subnetworks each have their particular set of special interfaces and operations. In order to build a uniform, "open system" architecture for weather datalink, special interface functions termed the "Airborne Data Link Processor" (ADLP) in the avionics, and the "Ground Data Link Processor" (GDLP) in the ground system, are added to each subnetwork as illustrated in Figure 1 below. The ADLP/GDLPs transform the subnetwork's particular operations to a standardized, "open system" interface. By isolating the specific details of each subnetwork at this point, the higher layers of the weather datalink system may treat each subnetwork as an identical set of defined functions. (Note: a new datalink subnetwork might be built to utilize the standardized "open system" interfaces from the outset – hence, it would not need the ADLP/GDLP functionality.)

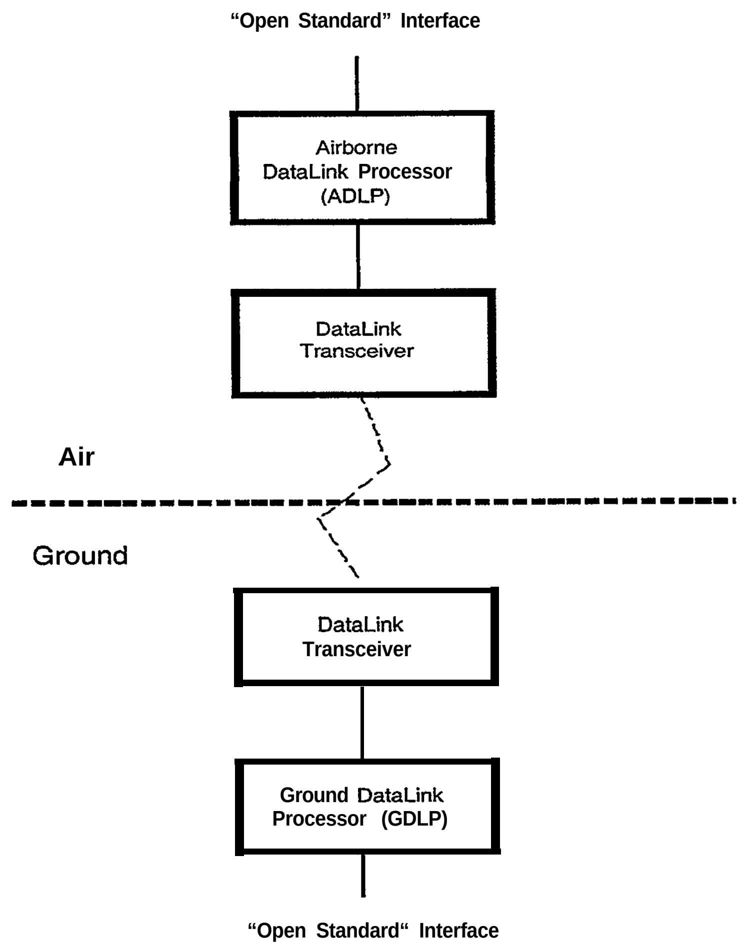

Figure 1 – The Subnetwork Architecture of an "Open System" Weather Data Link.

Figure 2 below illustrates the architecture of the higher levels of the "open system" weather datalink system. The ovals at the center of the figure symbolize the subnetworks with their "open system" interfaces. (Note: Figure 2 only shows three existing aviation datalink subnetworks for simplicity, although any number of datalink subnetworks may exist in the "open system" architecture.) Weather datalink applications on the ground connect to one or more routers that direct application data messages to the chosen subnetwork. Similarly, routers in the avionics connect to one or more subnetworks and direct appropriate application messages to their proper airborne application processing functions.

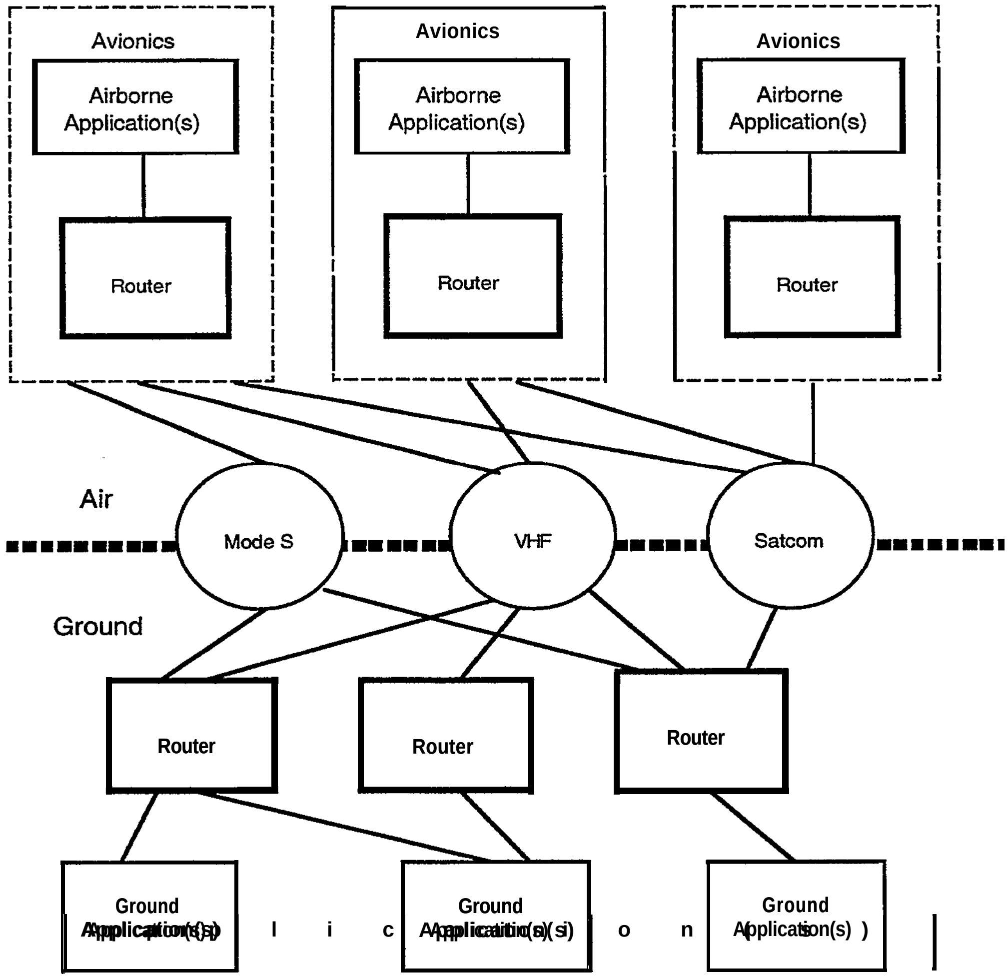

Figure 2 – The Higher-Level Architecture of an "Open System" Weather Data Link.

2.1 Requirements and Desired Features of a Weather Datalink Protocol

The following section lists seven requirements or highly desirable features in a data link protocol suitable for aviation weather applications. A short rationale for the inclusion of each of these requirements and features will be given here.

2.1.1 Open Standard

Clearly, utilizing a communications protocol that is an industry "open standard' makes the process of weather data link application development and certification easier. Implementations of the protocols will be available commercially -- for use as test systems if not also used as part of the application coding itself. There will be a track record of the benefits and drawbacks of the protocol in the public domain. The existing standards for the protocol need only be referenced, rather than requiring a set of new protocol standards to be written and agreed upon.

An impact of the incorporation of "open standards" into the weather data link system is a potential reduction in the cost of producing avionics. COTS implementations, test equipment, documentation, etc. can shorten the process of developing weather data link applications and systems. The path to certification can be shortened through the use of existing, well-known datalink techniques and protocols.

2.1.2 Efficient for Both Short and Long Messages

The range of aviation weather applications covers messages from only a few bytes in length to very long messages requiring many kilobytes or more. Typical textual messages such as single "Meteorological Aviation Reports" (METARs) and "Terminal Aerodrome Forecasts" (TAFs) typically run to about a hundred characters (assuming no compression is applied). However, combining all the METARs for a significant geographic region might run into several kilobytes. Graphical messages can be quite large in size. An example of what is probably a worst-case weather datalink message is the NEXRAD national mosaic. This database requires about 7 megabytes to cover the continental U.S. with a resolution of 2 kilometers (a bit more than 1 nautical mile) per 8-bit pixel. Each pixel can store information on several weather phenomena simultaneously, since the NWS standard for precipitation data utilizes only 3 bits. A straightforward, lossless runlength compression algorithm reduces a typical national mosaic (7 megabytes) down to an average length of 300 kilobytes. Hence, the national NEXRAD mosaic (about 300 kilobytes updated every 5 minutes) is probably as much weather data as any application will require on the datalink. This suggests that a reasonable weather data link system supporting several weather products need provide a worst-case throughput of a few kilobytes/second. (It should be noted that a practical implementation would not actually transmit the entire national NEXRAD database as a single message. Especially in the case of a broadcast datalink, the database would likely be sent as a sequence of independent sections, so that a receiver would not have to wait for error-free receipt of the entire database in order to display the desired part of it.)

Protocol overhead is also a significant concern, especially for shorter messages. It is very inefficient to have the message overhead be a significant fraction of the total datalink bandwidth usage. Minimizing the protocol overhead is an important consideration in the practical implementation of a weather data link. Sending excess bits over the data link costs money -- to the data link provider and the receiver.

2.1.3 Supports Broadcast and Multicast

A number of weather data link message applications will need to send the same (or similar) information to multiple aircraft in a geographic region. For example, all the aircraft seeking to land at a particular airport will want to access the weather data for the airport. The operational concepts for many weather data link applications involve sending information about a large-scale region of airspace to all the aircraft within that airspace. Hence, a broadcast or multicast weather data link provides the most efficient means to transfer the common data to each aircraft. The broadcast (or multicast) will use less bandwidth and require less link overhead (no aircraft request or end-to-end acknowledgements, etc.). (In the context of this paper, "broadcast" refers to the transmission of a given message to all connected end-systems, while "multicast" refers to the transmission of a given message to each of a predefined set of recipients.)

2.1.4 Supports Reliable Addressed Messaging

There are a number of weather data link applications that would benefit from (or might even require) reliable end-to-end messaging (in contrast to the broadcast-multicast applications discussed in 2.1.3 above). One example is the desire to have a positive indication that a message was received without error at an aircraft. This might be required for operational purposes (for example, the current requirement that an aircraft have accessed the most-recent ATIS message for a given terminal airspace). In cases where an aircraft accident potentially due to weather is being investigated, there would be a strong desire to have a log noting which weather data link messages were received (and at what time) by the aircraft. A second example involves application operational concepts that tailor individual messages to the requirements of the requesting aircraft. It may be more efficient in the use of data link bandwidth to provide only the information an aircraft needs -- even if multiple aircraft are in the airspace. There may not be significant overlap between the needs of each aircraft, so sending a common (i.e., broadcast) message would be less efficient than sending multiple tailored messages.

Provision of an end-to-end protocol provides for more reliable communications. The protocol can request retries in case of missing or erroneous data (but only in those cases). By contrast, a broadcast link would have to perform multiple broadcasts to attempt to ensure that all receivers successfully obtained the message. Management of the data link can be done more effectively when the link congestion and message traffic is well known. There is always a trade-off between the simplicity of broadcast and the reliability of end-to-end addressed dataliriks.

2.1.5 Support for Priority-Based Messaging

Weather data link messages may be divided into two general categories: “strategic” and “tactical” information. Strategic information is typically used for long-term flight planning -- decisions about avoidance of hazardous weather conditions along the proposed route of flight. Tactical information, on the other hand, is typically used for short-term (immediate) avoidance of local flight hazards. Clearly, the provision of good strategic weather information can often minimize the need for tactical weather decisions, but this may not always be possible. Weather is a highly-dynamic process that can change dramatically over relatively short time periods. Storm severity can change in a matter of minutes -- microburst downdrafts have lifetimes measured in minutes -- the decay of dangerous wake vortices is also a short-term phenomenon. Hence, while the weather data link system is routinely dealing with the flow of strategic information, there needs to be a means to interrupt this flow when a tactical message needs to take precedence. The weather data link system needs to have a means of attaching a priority to certain messages in order to insure their timely transmission.

2.1.6 Provisions for Mobile Routing

Clearly, any data link system trying to provide connection-oriented messaging to aircraft (see 2.1.4 above) must deal with mobile routing -- aircraft data link connectivity, like the weather, will be highly dynamic. The weather data link system will need to deal with the intermittent loss of links and dramatic shifts in message volume over time. Hence, an efficient mobile routing protocol must be part of the weather data link system. Management of data link congestion needs to be done transparently with respect to the weather applications.

2.1.7 Provisions for Link-Independent Routing

It is likely that weather information will be made available to aircraft via multiple data link media. Some data may use satellite links, while other data may come over VHF Data Link (VDL) or Mode S. The availability and suitability of various data links will change as aircraft travel. There will be a desire to minimize the costs associated with datalink -- when multiple links are available, the routing will want to use the cheapest suitable link. There will be administrative decisions made about which links may carry what information, or by what means a given piece of weather data is to be provided. Hence, it will be highly desirable for the weather data link protocol chosen to have the facility for "policy-based" routing. There will be a need for the protocol chosen to support multiple simultaneous links -- for example, if an aircraft is flying out of coverage of one link and needs to have a second link ready to take over ("make before break").

2.2 Existing and Developmental Systems

A number of weather data link systems have been developed and demonstrated by various organizations over the years. These systems have used a variety of data link methodologies, communications protocols, and physical data links. This section will give a brief description of some of the data link protocols and techniques used in these systems. The intent in this section is to survey the current field, highlighting where these system data link protocols do and do not provide for the desired data link features as listed in section 2.1 above.

2.2.1 Airline Communications and Reporting System

The Airline Communications and Reporting System (ACARS) is the primary data link in use on commercial aircraft today. It is a 2,400 bits-per-second, addressed, textual-format link supported over a 25 kHz bandwidth VHF radio link (see reference 1). ACARS is used today to carry a number of textual weather messages, along with airline operations (AOC) and other types of messages. A number of experimental and developmental weather data applications (along with ATC) have been hosted on the ACARS air-ground link. Digital "Automatic Terminal Information Service" (ATIS) is a weather data link that is currently supported over ACARS.

In order to extend ACARS beyond simple textual messaging, the ARINC 622 protocol was devised (see reference 2). The ARINC 622 protocol converts a binary message into a text string suitable for ACARS by transforming each 4-bit "nibble" of the message into a hexadecimal character. Control information, including data length and an overall message cyclic redundancy check (CRC), is added in the ARINC 622 protocol. It should be noted that ARINC 622 acts as a data expander where four bits of data becomes an 8-bit ACARS character. The idea is that the overall compression achieved by binary coding of the data is sufficient to outweigh the 2: 1 expansion involved with the ARINC 622 protocol itself.

A replacement system for ACARS is under development, called VDL-2. The VDL-2 system provides for a total data rate of 3 1.5 kilobits-per-second on a 25 kHz bandwidth VHF radio link. VDL-2 will support fully-binary data, and it will be one of the supported subnetworks of the ATN (see 2.2.2 below).

2.2.2 Aeronautical Telecommunications Network (ATN)

The ATN is a data link system being developed by the International Civil Aviation Organization (ICAO) to provide a worldwide standard for aeronautical data link communications. The ATN will be a general communications system for aviation to support all sorts of AAC, APC, ATC, and AOC applications. The ATN will be used for ground-ground communication as well as ground-air messages (possibly even air-air). The applications currently defined for the ATN include ADS, ATIS, and CPDLC.

The goals of the air-ground ATN project parallel most of those described in section 2.1 above. In particular, the ATN is designed to provide an "open standard', subnetwork-independent, reliable communications channel for data link applications in aviation. The ATN system is defined in the ICAO SARPS document (reference 3) by a highly-tailored and somewhat modified subset of the OSI protocols. The ATN defines the functions for each of the seven layers in the OSI model. Each of the supported ATN datalink subnetworks (Mode S, VDL, Satcom, HF) interfaces to the higher layers using the "open standard" IS0 8208 protocol. Specifications of the subnetwork ADLP and GDLP functions for each of these subnetworks have been written as part of the ATN design. Mobile routing functions are provided utilizing the IS0 "open standard" Interdomain Routing Protocol (IDRP). Although the ATN OSI protocols are complex and involve significant overhead both in data bandwidth and processing, they are largely based on open standards. The tailoring and modification of the OSI protocols chosen by the ATN standards help to reduce the system overhead. Additional features of the OSI model (including data security and presentation control) can be incorporated into the ATN design through existing provisions of the OSI protocols.

It should be noted that the OSI protocols were developed to provide a reliable, end-to-end addressed datalink. There is no support in the OSI protocols (and, hence, the ATN) for broadcast or multi-cast applications. All ATN applications must set up a unique point-to-point link before data can be transferred. It was seen that this can be a significant drawback for some aeronautical applications, so a number of the ATN subnetworks developed "subnetwork-specific services". These provide a means for ATN users to access particular functions (such as broadcast) of a given subnetwork. The drawback to subnetwork-specific protocols is that the application using them must be tailored to the specific subnetwork, and the goal of link transparency is lost.

As was mentioned above, the "general-purpose" nature of the ATN entails considerable system overhead. For example, the ATN design requires 22 bytes to specify an address (as compared to the Internet 4-byte address, and the Mode S 3-byte aircraft address). The ATN address structure is highly hierarchical, allowing for flexible prefix-addressing of address domains. Still, the roughly 10-to-the-53rd-power addressing range requires a lot of bandwidth. Some simulation studies of the ATN design using the full OSI upper-layer stack and assuming SO-byte messages transmitted every 5 minutes or so (typical of short weather data link applications) showed that the ATN link efficiency (ratio of user-data to total link traffic) would be less than 15 percent. Using a compressed, connectionless transport layer with no upper-layer services upped the ATN link efficiency for this data loading to nearly 65 percent. Longer messages would provide for more efficiency. These inefficiencies are not major concerns for ground-ground applications over high-speed links, but they are very expensive to maintain across bandwidth-limited air-ground links.

2.2.3 Graphical Weather Service (GWS) Textual Weather Service (TWS)

The GWS/TWS system was developed by M.I.T. Lincoln Laboratory to operate over the Mode S subnetwork of the ATN [the Mode S-Specific Protocols (MSP) are defined in reference 4]. Both are based on a request-reply paradigm. GWS provides access to selected portions of the national NEXRAD database precipitation mosaic -- the requestor specifies the center point latitude/longitude, radius (25-200 nautical miles), and time of the image desired. TWS provides access to selected components of the national METAR and TAP textual database -- the requestor specifies the desired reference identifier for the reporting site.

The Mode S subnetwork provides addressed, reliable end-to-end transmission of short data packets. Mode S data link is interspersed with the surveillance function of the Mode S sensor, so the data "bursts" must be short to deal with the short time that a given aircraft is within the antenna beam. (The Mode S terminal sensor antenna rotates in an approximately 5-second period termed a "scan" – enroute sensors have a 12-second scan). The Mode S data link protocols provide for two types of messages. Standard Length Messages (SLM) contain 7 data bytes each, and up to 4 SLMs may be combined in a single message unit. GWS/TWS uses a single downlink SLM for the service request. Extended Length Messages (ELM) contain 10 data byes each, and up to 16 ELMs may be combined in a single message unit. Hence, the maximum length of a Mode S data packet is 160 bytes, but ground-to-air uplink overhead dealing with multiple sensors communicating with a given aircraft reduces this to an effective total of 149 bytes. The Mode S subnetwork provides error correction and retry support for each SLM or ELM.

A one-byte MSP header provides for up to 63 "channels" over the MSP data link. In order to provide for uplink GWS/TWS response messages longer than 149 bytes, a special GWS/TWS application message segmentation protocol was added on top of MSP that allows up to 16 Mode S ELMs to be combined into a single GWS/TWS message (taking up to 16 antenna scans) of up to 2,384 total bytes. (Since transfer of each Mode S packet requires a rotation of the sensor antenna (-5 seconds), longer application messages could not be used with an adequate response time. A practical upper bound is around 3 to 4 ELMs.) The first ELM in the message starts with a 3-byte GWS/TWS header incorporating an overall 16-bit message cyclic redundancy check (CRC). Subsequent ELMs get a single byte GWS/TWS header.

The application protocol includes a 6-bit application identifier (to select various formats of GWS and TWS) and a message number to provide the ability to have up to 31 outstanding requests in the system at a time. The MSP protocols assume that only one ground sensor is in communication with a given aircraft application at a time. End-to-end addressing consists of the aircraft's Mode S address (part of the underlying Mode S protocols) and the MSP channel number.

Note that the maximum message length limitation of 2,384 bytes for the MSP applications requires extensive data compression for graphical images. GWS images are typically 256-by-256 square pixel arrays with each pixel representing a 2 kilometer square surface region (the map is about 135 nautical miles in radius). Each 2-bit pixel value represents a reduced set of the NWS precipitation weather levels. Hence, a typical raw GWS image occupies 16 kilobytes. A special-purpose compression scheme (the Weather Huffman algorithm, see reference 5) was developed that would compress GWS images down to less than 4 kilobytes with no distortion and less than 500 bytes with minor distortion. Compression was not required for TWS text (other than using 6 bits per character for the upper-case only alphanumeric data).

2.2.4 RTCA Flight Information Service-Broadcast (FIS-B) MASPS

Work is ongoing within the RTCA Special Committee SC-195 to develop a Minimum Aviation System Performance Standard (MASPS) document for a system to provide Flight Information Services (primarily textual and graphical weather data applications) via a broadcast datalink (see reference 6). While the original use for this document is to support the FAA's procurement of a national FIS-B system utilizing VHF datalink (VDL), the work is intended to produce protocol standards that are as link-independent as possible.

The MASPS assumes a unidirectional broadcast uplink with a repeat cycle of 5 minutes or less. Weather products to be supported include textual applications (ex. METARs, TAFs, SIGMETS, AIRMETS, etc.), and graphical applications (NEXRAD mosaics, etc.). A variety of data formats, representation schemes, and compression methods are included in the MASPS document. The MASPS requires that all protocols and algorithms be "open systems" and that all data compression methods (if any are used) be lossless (to simplify certification testing).

It should be noted that the curvature of the Earth limits the effective "line of sight" range of ground-based links such as VDL or Mode S. Assuming that the broadcast antenna is sited on a modest tower (less than 500 feet high) and no obstructions lie on the horizon, the Earth's curvature imposes a minimum altitude floor for reception that is a function of range from the broadcast antenna as shown in the following table:

TABLE 1 – "Line of Sight" Range as a Function of Receiver Altitude

Range (nautical miles)	Altitude Floor (feet)
40	1,000
55	2,000
70	3,000
80	4,000
90	5,000
95	6,000
125	10,000
155	15,000
175	20,000

Hence, the FIS-B system will limit its "local" broadcast weather products (primarily intended for general aviation (GA) aircraft flying below 10,000 feet) to a region of about 100 nautical miles in radius from the broadcast site.

The underlying link layer defined in the MASPS for FIS-B applications is the IS0 3309 "Unnumbered Information (UI) Frames" protocol. This is a purely broadcast subset of the protocol that is used in industry-standard HDLC systems. The IS0 3309 protocol entails a 13-byte overhead for each data packet, providing primarily an error detection function via a cyclic redundancy check (CRC). The IS0 3309 protocol incorporates two 32-bit addresses -- source and destination. The destination address is defaulted in the broadcast mode. Two special "flag" bytes (7E hexadecimal) frame the data. The protocol does not incorporate any data length field -- the ending flag byte marks the end of the frame. This has the advantage of placing no upper bound on the amount of data that can be carried in one frame. It has the disadvantage that "zero-bit stuffing" must be incorporated into the protocol -- i.e., no data can have 6 one-bits in a row or it might be construed as a flag byte.

Beyond IS0 3309, the MASPS defines a special protocol [termed the "Application Protocol" (AP)] providing for message segmentation and reassembly, application function descriptor, geographic reference, compression method selector, and message time stamp. The typical AP overhead is about 11 bytes. (The geographic reference in the AP header is used to allow applications receiving the PIS-B uplinks to quickly discard messages that are not of local interest based on their geographic area of coverage).

2.2.5 Weather Downlink

Each of the preceding weather data link applications in this section has uplinked weather information available on the ground to airborne recipients. There are also applications that downlink weather information gathered onboard aircraft to ground recipients. This weather downlink can be used to gain a fuller picture of current weather conditions and potential flight hazards that can provide an aid to ATC operations. (Short-lived weather phenomena such as wake vortices, wind shear, and microbursts may be detected when an aircraft flies through one in a terminal area.) The downlinked weather can be added to the database that will be available for uplink to aircraft now coming into the impacted airspace. Among the local data items available in the avionics of many commercial aircraft (and high-end GA aircraft) are:

(1) air temperature
(2) air pressure
(3) dew point
(4) humidity
(5) wind speed and direction
(6) turbulence measure
(7) icing

Taken together with the current aircraft position report, the weather downlink applications add airborne "snapshots" to the available weather database.

2.2.5.1 Mode S

Each Mode S transponder contains an array of 255 56-bit data buffers, termed "Ground Initiated Comm-B" (GICB) registers. As defined in reference 7, these registers contain information obtained from many areas of the avionics. A given register may be read out from the ground by interrogating the transponder using the GICB protocol -the Mode S transponder response will contain the specified 56-bit register contents. (Note that this GICB read-out can be performed air-air as well as ground-air.) Registers 44 and 45 (hexadecimal) are defined to hold current weather data as follows:

Register 44 (hex)	Wind Speed (knots) Wind Direction (degrees true) Air Temperature (quarter degrees C) Air Pressure (hPa) Turbulence (none, light, moderate, severe) Humidity (percent)
Register 45 (hex)	Turbulence (none, light, moderate, severe) Wind Shear (none, light, moderate, severe) Microburst (none, light, moderate, severe) Icing (none, light, moderate, severe) Wake Vortex (none, light, moderate, severe) Air Temperature (quarter degrees C) Air Pressure (hPa) Radio Altitude (feet)

Note that register 44 (hex) contains the "routine" weather report, while register 45 (hex) contains the "hazard" report. (Each data item in these registers has an associated status bit to indicate whether the data is currently available.) It is assumed that the aircraft location at the time of the report will be obtained through some other means (Mode S tracking, readout of the position GICB registers, etc.)

2.2.5.2 AUTOMET

The RTCA Special Committee 195 has drafted a Minimum Operational Performance Standard (MOPS) for automated downlink reporting of air-derived weather data (reference 8). The AUTOMET document provides format specifications for weather downlinks and control uplinks. AUTOMET does not define the physical link to be used for data transfer – although initial implementations will probably use ACARS (see 2.2.1 above) and later VDL. AUTOMET uplink commands can specify the AUTOMET downlink report generation to occur at specific intervals in time, distance, or altitude. The AUTOMET downlink reports contain the standard weather data as described above; temperature, pressure, wind speed/direction, humidity, etc.

2.3 Proposed Weather Data Link Protocol Selection - TCP/IP

After consideration of several different protocols in use or proposed for aviation data link, it seems that there is little reason for the weather data link system not to use the most-common set of communications standard protocols available today -- TCP/IP. These are the protocols underlying the Internet, as well as nearly all-commercial computer-computer links. As will be discussed further in this section, TCP/IP is a match for nearly all of the seven "requirements/desired features" that were enumerated in section 2.1 above. Where TCP/IP does not support the desired features directly (or efficiently), there are available techniques to work around the problems or, at least, to minimize them.

2.3.1 TCP/IP Meets Almost All the Requirements

There can be little doubt that TCP/IP is the most "open" of all communications protocol standards. Since TCP/IP is the chosen protocol of the Internet, software implementations are widely available. The TCP/IP protocol stack is a standard part of all UNIX systems. TCP/IP support is provided under Microsoft "Windows" for PCs and on Apple Macintosh computers. Numerous stand-alone TCP/IP implementations are commercially available for embedded processor systems of many types. A single IC-chip hardware-only implementation of TCP/IP is commercially available at a cost of less than $10 each (see 9). The TCP/IP protocol documentation is freely available from the Internet Engineering Task Force (IETF), and dozens of books (including reference 10) describe the operation and parameters of TCP/IP. TCP/IP satisfies requirement 2.1.1 above with ease.

The basis of the TCP/IP protocols is the network-layer "Internet Protocol" (IP). IP provides an "unreliable, connectionless datagram" service. (Note: TCP/IP protocols make no assumptions about the underlying datalink layer in the system. This could be a serial line, Ethernet connection, VDL, Mode S, Satcom, or other datalink.) There is no network-layer mechanism in IP for error handling or maintaining packet ordering. The 20-byte IP header contains the following fields:

Version Number:	4 bits	-- current IP standard is version 4
Header Length:	4 bits	-- measured in 32-bit increments
Type of Service (TOS):	8 bits	-- optimize delay, cost, reliability, or throughput
Packet Length:	16 bits	-- measured in bytes (max. is 65,535)
Identifier Code:	16 bits	-- identifies the data packet
Fragmentation Flags:	3 bits	-- controls for fragmenting IP packets of a message
Fragment Offset:	13 bits	-- offset of fragment in message (in 8-byte units)
Time To Live (TTL):	8 bits	-- max. number of router hops allowed
Protocol Type:	8 bits	-- selects UDP, TCP, etc.
Header Checksum:	16 bits	
Source Address:	32 bits	
Destination Address:	32 bits	

Built upon IP are two standard transport-level protocols: User Datagram Protocol (UDP) and Transmission Control Protocol (TCP). UDP is a very simple datagram-oriented protocol -- each output operation produces exactly one UDP datagram, which then results in one IP datagram transmitted. UDP provides for a simple, minimum-overhead, broadcast/multicast paradigm (such as described in requirement 2.1.3 above). Messages that are lost or corrupted are simply ignored. The S-byte UDP header contains the following fields:

Source Port Number:	16 bits	-- identifies the sender
Destination Port Number:	16 bits	-- identifies the receiver
UDP Length:	16 bits	-- measured in bytes (max. is 65,535)
UDP Checksum:	16 bits	

The UDP header enables a receiver to determine if a particular packet was intended for it, and to do an overall error check. Hence, for a simple broadcast link using UDP, the total overhead is 28 bytes per packet. The maximum data payload available in a UDP message is 65,535-28 or 65,507 bytes.

Note that a header compression scheme (see 11) has been defined over UDP/IP for use in broadcast operation. The compression works by recognizing the common UDP/IP header elements in the continual stream of binary data. The common elements are sent periodically -- at all other times, a 16-bit identifier code is sent in their place. A receiver must wait to receive the periodic full UDP/IP header, record the common elements from the header, and can then apply these common elements to subsequent compressed packets. For a non-fragmented data stream, the 28 bytes of UDP/IP header are compressed to 4 bytes -- the 16-bit UDP checksum and the 16-bit Identifier Code. Multiple streams, each with differing common elements, may be maintained through the same broadcast device. Thus, if there are two applications, each with their own UDP port number, two streams would be created.

IP supports three kinds of addressing: unicast, broadcast, and multicast. Broadcast and multicast apply only to UDP, where it makes sense for an application to send a single message to multiple recipients. Broadcast sends the same message to all recipients, while multicast sends the same message to a selected subset of the recipients. Special "well-known" addresses are used in UDP/IP to specify broadcast-multicast. Setting the high-order 3 bits of the address to ones is the indicator for multicast -- the remaining 28 bits specify a multicast group identifier. Setting all the address bits to ones indicates a broadcast. (There are several other addressing conventions available in UDP/IP.) It can be seen that UDP/IP provides for the broadcast/multicast requirements of 2.1.3 above. Broadcast or multicast messages of up to 65,508 data bytes are supported by UDP/IP. This supports the short and long (medium?) message length requirements of 2.1.2 above.

TCP provides a service to applications totally distinct from UDP, even though they both rest on top of IP. TCP provides a connection-oriented, reliable byte-stream service. The term "connection-oriented" means that the sender and receiver applications must establish a TCP connection with each other prior to any exchange of data. (This is also termed a "virtual circuit".) There are exactly two end points communicating with each other over a TCP connection. The TCP connection is full-duplex -- there are independent data flows in both directions. TCP does not support broadcast or multicast (UDP provides for these). TCP provides "reliability" by doing each of the following functions:

(a) Segmenting the data into variable-sized packets that are reassembled upon reception.
(b) Providing a data time-out to force retransmission of lost data
(c) Acknowledging receipt of each data packet
(d) Maintaining an end-to-end checksum on the header and data
(e) Resequencing data packets that are received out of order
(f) Removing duplicated packets
(g) Providing flow-control to manage buffers in the receiver

Hence, TCP provides for the reliable addressed messaging requirement of 2.1.4 above.

The TCP header requires 20 bytes (much larger than the S-byte UDP header). The fields in the TCP header are summarized below:

Source Port Number:	16 bits -- identifies the sender
Destination Port Number:	16 bits -- identifies the recipient
Sequence Number:	32 bits -- indicates which packet in overall message
Acknowledgment Number:	32 bits -- indicates last packet successfully received

Header Length:	4 bits -- measured in 32-bit words
Reserved:	6 bits
Flag Bits:	6 bits -- URG ==> urgent pointer valid
	ACK ==> acknowledgement number valid
	PSH ==> pass data as soon as possible
	RST ==> reset the connection
	SYN ==> synchronize sequence numbers
	FIN ==> the sender is finished sending data
Window Size:	16 bits -- in bytes, up to 65,535
TCP Checksum:	16 bits
Urgent Pointer:	16 bits

The Urgent Pointer mechanism (valid when the URG bit is set) provides a means for TCP to deal with high-priority data where the normal flow of communications is to be interrupted for some special information. This feature of TCP provides for the priority-based messaging requirement of 2.1.5 above.

In addition to the TCP header fields, an extra 32-bit "Options" field may be added. One possible option called the "Window Scale", allows extension of the window size specification from 16 to 32-bits. This provides for very large data packets (gigabytes). This option can only appear in a SYN packet. (SYN packets are used to establish the parameters for a given data exchange between end-systems.) A second possible TCP header option is the "Timestamp". This option places a 32-bit timestamp (typically in 500 millisecond units) value in every data segment. The acknowledgement also holds a timestamp. This TCP option allows the TCP connection to better measure and manage the delays over the link. A third possible TCP header option, called the "Maximum Segment Size" (MSS), allows the recipient to determine how much buffering will be needed. The MSS option can only appear in a SYN packet. This MSS option may be used for routing decisions -- choosing a path that will not require segmentation and reassembly of the message packets.

Mobile Routing support for a single subnetwork link has recently been made a feature of TCP/IP (see reference 12). Hence, TCP/IP routing supports the mobile routing requirement of 2.1.6 above. The provisions for link-independent routing as defined in 2.1.7 are not currently addressed by the TCP/IP standards -- a work-around procedure called "policy routing" is described in 2.3.2.1 below.

The following table will seek to summarize how the TCP/UDP/IP protocols provide for the features/requirements of section 2.1 above. An additional column shows how the ATN (based on OSI protocols) fits with these features/requirements. The notation "x" indicates that the protocol provides this function, while the notation "-" indicates that the protocol does not provide this function. Lower-case letters in the table refer to footnotes.

TABLE 2 – Protocol Provisions for Requirements and Desired Features

Requirement/Feature	IP	UDP	TCP	ATN
Open Standard	x	x	x	a
Long / Short messages	b	b	x	x
Broadcast / Multicast	x	x	-	c
Reliable-Addressed		-	x	x
Priority Messaging		-	x	x
Mobile routing	x	x	x	x
Link-independent routing	-		d	x

(a) ATN utilizes tailored and extended IS0 protocol specifications.

(b) IP and UDP have an upper bound of 65,508 bytes of data per message. TCP can handle messages with gigabytes of data.

(c) ATN provides broadcast support only through subnetwork-specific protocols.

(d) TCP can be extended to provide link-independent routing, see 2.3.2.1 below.

In conclusion, it appears that the TCP/UDP/IP protocols provide a good match to the desirable features/requirements list defined for weather data link systems. There is a choice of protocols and options available to fulfill the requirements, with the exceptions to be described in the next section.

2.3.2 Disadvantages of TCP/IP

While the TCP/IP protocols amply support most of the desired features for weather data link, there are two areas where they fall short (as indicated in 2.3.1 above). This section will discuss these two shortcomings and suggest how they might be overcome.

2.3.2.1 Support of Multiple Subnetworks

The Mobile IP standard (reference 10) makes the assumption that there is only one subnetwork link being used for mobility to a given end-system at one given time. The standard states that if two paths are found to be available from the message sender to the addressed receiver, then the message packets are to be duplicated and then routed over both paths. It is assumed that this situation of multiple routing paths only arises when the receiver is in transition from connection by one subnetwork to the other. The assumption makes sense in the standard model -- there is a good chance that the subnetwork in the previous domain may fail to transfer the message packet before the transition occurs -- this "make before break" design helps to ensure that the message will get through by one subnetwork or the other.

Extending the Mobile IP standard to deal with multiple active subnetworks can be done by instituting a form of "policy routing" (reference 13) beyond the usual TCP/IP routing algorithms. Mobile IP specifies that each time a mobile node (aircraft) enters the coverage of a given subnetwork (and periodically thereafter), a registration packet is sent to the node's "home agent". This registration packet allows a sender to attach some user-specific information. Multiple subnetworks may be distinguished from each other by the user-specific information (subnetwork identity, quality-of-service available, etc.). When the registration packet arrives at the "home agent", the user-specific information is used to indicate the multiple routes (one for each subnetwork supported by the aircraft) in the forwarding database of the router. When a data link message packet arrives at a router that has multiple routing entries in its database, the router applies a pre-existing policy to determine which route to use. The policy could use the source IP address, any static information (i.e., domain 'x' always prefers subnetwork 'A' over subnetwork 'B'), or the type of service (4-bit TOS) field in the IP packet. The policy could be administrative (i.e., weather application 'x' always uses subnetwork 'Q') or based on some subnetwork metric (i.e., use the "cheapest" available subnetwork). The port number in the TCP/UDP header could also be used to send certain applications over certain subnetworks. (Note that the aircraft would have to employ an equivalent policy-routing function to manage its multiple routing entries).

The "policy module" has been implemented in a straightforward manner as part of a standard TCP/IP system (such as Berkley UNIX) as a pseudo-driver. The use of a "pseudo-driver" allows the standard routing logic to remain unmodified and highly efficient for normal tasks. Only those aircraft supporting multiple subnetworks would require the application of the additional policy routing code.

Note that the "policy module" described here is only required for those weather datalink system implementations that must support multiple datalinks. If an aircraft only employs a single datalink at a time, the "standard" mobile IP routing protocol may be used in its avionics and the mobile IP protocol would be totally compatible with the proposed weather data link system.

2.3.2.2 Provisions for Long-latency Subnetworks

The TCP protocols concerning message flow and congestion control can have problems with the interaction between high-speed subnetworks and low-speed subnetworks. Since the initial design of TCP assumed nodes interconnected by more-or-less equivalent high-speed landlines, this was not considered a significant problem at first. However, with slower links (i.e., satellite, Mode S, etc.) typical of the aviation weather application, care must be taken to deal with TCP's inherent problems in dealing with long-latency subnetworks.

Consider a TCP sender with a large message to send. It may send out a large number of data packets as fast as it can, assuming that almost all will get through to the recipient. Some time later, the sender gets an acknowledgement back, and may need to retransmit a few packets. However, it is possible that the link (or intermediate routers) could not handle the flood of data packets and many of them did not get through. Now, the sender must wait for the acknowledgement, retry again, wait again, etc. for many cycles of the link latency. If the link takes a relatively long time to respond, then the effective throughput can be greatly diminished. On the other hand, the sender could send out just a single data packet and wait for reception. Upon acknowledgement, the sender gets a bit "bolder" and sends two packets. Next time, it can send four packets, and so on. This avoids saturating the link and spreads the utilization over the entire transfer period. After the exponential growth to a good packet rate, the TCP system goes into a congestion-avoidance mode maintaining a linear growth rate that slows down the system to avoid overload.

Clearly, there are a lot of tradeoffs possible to gain efficiency here. First of all, the standard TCP packet data size can be no larger than 64 kilobytes -- the default for most implementations is 4-8 kilobytes. The larger the data size, the fewer packets need be sent and the smaller the impact of congestion control. Second, several extensions to TCP have been defined by the IETF to assist in the performance of TCP implementations:

RFC 1323: TCP Extensions for High Performance
RFC 2018: TCP Selective Acknowledgement Options
RFC 2001: TCP Fast Retransmit/Fast Recovery

Third, alternative non-TCP special subnetwork protocols may be defined to work over a specific subnetwork to optimize its behavior. (This is similar to the "subnetwork-specific protocols" defined in the ATN system described in section 2.2.2 above.) One approach to this is to have a "subnetwork" that is actually several TCP links working in parallel.

Fortunately for the weather data link applications, there is seldom so much data that the TCP/IP limitations are of great impact. The TCP extensions described above deal with the problem successfully up to message sizes in many 10's up to 100's of kilobytes. Using large data sizes also helps to make TCP more efficient.

3. Data Link Application Formats

Beyond the selection and tailoring of the communications protocols, the task of specifying weather data link applications requires a review of the types and formats of the data being communicated. Weather data applications may be broken down into three main categories: textual, gridded graphical, and iconic graphic. The following sections will discuss each of these methods for weather data expression via a datalink. Display and compression/encoding algorithms for each category will be covered here. Textual techniques are described in section 3.1 below. Special-purpose gridded graphical compression algorithms are described in section 3.2 below. The Gridded Binary (GRIB) and Binary Universal Form for the Representation of meteorological data (BUFR) formats developed by the World Meteorological Organization (WMO) especially for the transmission and storage of weather data will be outlined in sections 3.4 and 3.5 below. GRIB and BUFR incorporate elements of each of the weather data display types, and also incorporates many other features to control the display generation of meteorological information. The NCAR-proposed gridded-graphical AWIN format will also be discussed in section 3.6 below. A summary of the tradeoffs involved with each format standard is given in section 3.7 below.

3.1 Textual Data

Text is the most-basic format for weather information. Almost all current weather information datalinked to aircraft is in the form of text. METARs, TAFs, SIGMETs, etc., are all currently generated and transmitted in textual form. Text is the simplest and cheapest weather data link format to decode and display. Textual cockpit displays are relatively inexpensive -- and they can be made to take up a minimum of scarce instrument panel space. While future weather applications may choose to switch over to graphical representations, basic textual weather information will remain a significant part of the weather data link system

In its simplest form, textual data is simply a stream of fixed-length characters chosen from a defined alphabet. For example, shown below is a sample METAR for Nashville, TN taken from reference 14. This highly-abbreviated text message for 12:50 UTC indicates current winds from 330 degrees at 18 knots, wind direction variable between 290 and 360 degrees, visibility 0.5 statute miles, visibility for runway 31 is 2700 feet, in heavy snow with blowing snow and fog, obscured sky at 800 feet, temperature 0 degrees C, dewpoint -3 degrees C, altimeter setting 29.91 inches, rain ended 42 minutes past the hour and snow began 42 minutes after the hour.

METAR KBNA 12502 33018KT 290V360 1/2SM R31/2700FT +SN
BLSNFG WOO8 00/M03 A2991 RMK RERAE42SNB42

Since aviation weather and ATC data is standardized in the English language, the character set chosen is ASCII (also known as IS0 alphabet #5). ASCII characters occupy a byte (8 bits). Well-known textual stream compression algorithms are available that can reduce the required bandwidth for textual formatted data. Such stream compression algorithms have the property that the more repetition occurs in the input data, the better they compress. (Aviation weather messages tend to have significant repetition.) Hence, longer text strings tend to compress better than shorter strings. In fact, a short string with little redundancy might even be somewhat expanded by the algorithm. The RTCA SC-195 has done experiments with the DEFLATE compression algorithm (reference 15) that is incorporated into its Flight Information Service standards (see 2.2.4 and reference 6). DEFLATE is a highly-optimized text stream compression algorithm available in the public domain. A set of 44 Terminal Aerodrome Forecasts (TAFs) was randomly selected and compressed with DEFLATE. The compression of each TAF ranged from a best of 0.53 (354 characters uncompressed / 189 bytes compressed) to some worst-case TAFs that were actually expanded by as much as 10 percent. Averaged over the 44 individual TAFs, the DEFLATE algorithm achieved a compression of 0.73 compared to the ASCII text (7,239 total characters uncompressed / 5,296 bytes' compressed). However, if the set of 44 TAFs was bundled as a single message and compressed as a unit, DEFLATE achieved compression down to 1,732 bytes (compression 0.24).

ICAO SICASP has developed an algorithm for the efficient expression of ATC and weather data link messages called "Data Link Applications Coding" (DLAC) (see reference 16). DLAC achieves compression of textual-format data in several ways. First, since current textual ATC and weather data link messages use only upper-case alphanumeric characters, DLAC defines a 6-bit reduced character set. This immediately achieves a 25% compression over ASCII. Second, DLAC recognizes that aviation traffic control and weather messages are highly constrained and tend to have a rather limited vocabulary and phraseology. A 1000-word "standard dictionary" is defined in DLAC containing the most-common words/phrases used in ATC messages. Whenever a dictionary entry is found in the input text, its lo-bit dictionary index is encoded instead of the character string. Control bits are incorporated into the message encoding to switch between DLAC dictionary coding and DLAC character coding. For example, the text string "SEVERE THUNDERSTORM EXPECTED 50 NAUTICAL MILES SOUTHEAST OF AIRPORT" is compressed by DLAC from 68 text characters to 20 encoded bytes (compression 0.29 compared with ASCII text). The example text string has 6 DLAC dictionary entries. An average DLAC compression is more likely to be about 50 percent. Unlike stream compression algorithms, DLAC does not require long messages to achieve its best compression. Also, DLAC has provisions for adding new dictionaries beyond the standard one -- so additional weather-related terms could be readily incorporated. (Note that DLAC compression actually does better when words are fully expressed rather than using the terse abbreviations commonly used in today's weather data link messages.)

A subsequent step undertaken in ICAO is to reduce standard ATC and weather textual messages into a highly encoded form that recognizes the fixed format of the messages and suppresses unnecessary bits. The format is described using "Abstract Syntax Notation version One" (ASN. 1). ASN. 1 (see reference 17) was developed to represent message syntax at the Presentation Layer of the OSI model. ASN. 1 message encoding is not strictly-speaking a compression algorithm -- it is rather a very efficient form for representing data that is in a well-known and structured format. ASN. 1 encoding may be converted to/from a bit-stream via two sets of "encoding rules". The Basic Encoding Rules (BER) provide for unlimited flexibility and extensibility, but they are highly inefficient in terms of bit-usage overhead. A second set of encoding rules, Packed Encoding Rules (PER), has been defined to allow highly efficient bit-wise expression of ASN.l syntax when the sender and receiver both know the syntax specification ***a priori***, and where the syntax definition is highly constrained (see reference 18).

A simple example of the application of ASN. 1 and PER is the expression of the time of day. In text, this might be written as "10302" (10:30 Zulu), requiring 5 characters (40 bits in ASCII, 30 bits in DLAC 6-bit coding). However, the actual data here requires only 11 bits -- 5 bits for the hour (O-24) and 6 bits for the minute (O-59). An ASN.l representation of the time in this way (encoded using PER) would achieve a high degree of compression over the equivalent text. ICAO is currently in the process of defining METARs, TAFs, etc., in ASN.l formats.

3.2 Gridded Graphical Data

The most-basic form of graphical weather data is the gridded image. This format consists of a rectangular array of pixels, and each pixel holds a value indicating the strength of a weather feature of interest (i.e., precipitation intensity, level of turbulence, etc.). A common example of this sort of weather data is the NEXRAD precipitation map, where the content of each pixel indicates the NWS 7-level radar precipitation intensity for that area. The pixels are typically 2 kilometers (about 1 nautical mile) in extent. Such gridded graphical images can require a lot of data -- a 256-by-256 pixel image (-275 nautical miles on a side) with 3-bit pixels (7 NWS weather levels) would require 24,576 bytes. Clearly, some extensive data compression would be useful to handle these large maps over a bandwidth-limited (or expensive) weather data link.

Reference 19 describes some research conducted into the issue of gridded graphical data compression. A number of standard, lossless techniques, including runlength encoding, Huffman coding, and Lempel-Ziv algorithms, were examined. The best of these algorithms achieved 6-to-1 lossless compressions on a set of typical NEXRAD 256-by-256 maps. If isolated pixels were filtered from the input map (allowing a small loss in the compression), the standard algorithms achieved as much as 10.5-to-1 compression. Two new gridded graphical compression algorithms were developed at Lincoln Laboratory: the Weather-Huffman (WH) (reference 5) and the Polygon-Ellipse (PE) (reference 20). These algorithms can achieve lossless compressions of up to 12-to-l. In addition, they are capable of controlled-loss compression beyond 60-to-l. WH and PE each take a maximum bit count as an input parameter -- they will encode using the minimum distortion required to get under the bit limit. (Setting the bit limit to "infinity" causes lossless compression.) WH and PE both prioritize their distortion by weather level -- the higher the level, the higher the fidelity. Both algorithms distort by enlarging weather regions rather than shrinking them. No region of high-level weather can be totally lost. As the name implies, PE works by fitting the weather with a combination of polygons and ellipses. WH is based on Huffman encoding, but employs Hilbert scanning and a unique combination of other techniques to maintain best fidelity for highly-compressed maps. (Highly-compressed PE maps with their inherent distortion tend to look too "cartoonish" when evaluated by pilots.) WH was used in the GWS system (section 2.2.3 above). WH maps proved to be quite acceptable to pilots even at high levels of compression. (Figure 3 below shows a sample 256-by-256 pixel 3-level input NEXRAD precipitation image of a storm front on the left side of the figure and a WH lossy-compression version of this data on the right side of the figure. WH has compressed the NEXRAD image more than 54-to-1 with quite reasonable fidelity to the original input.)

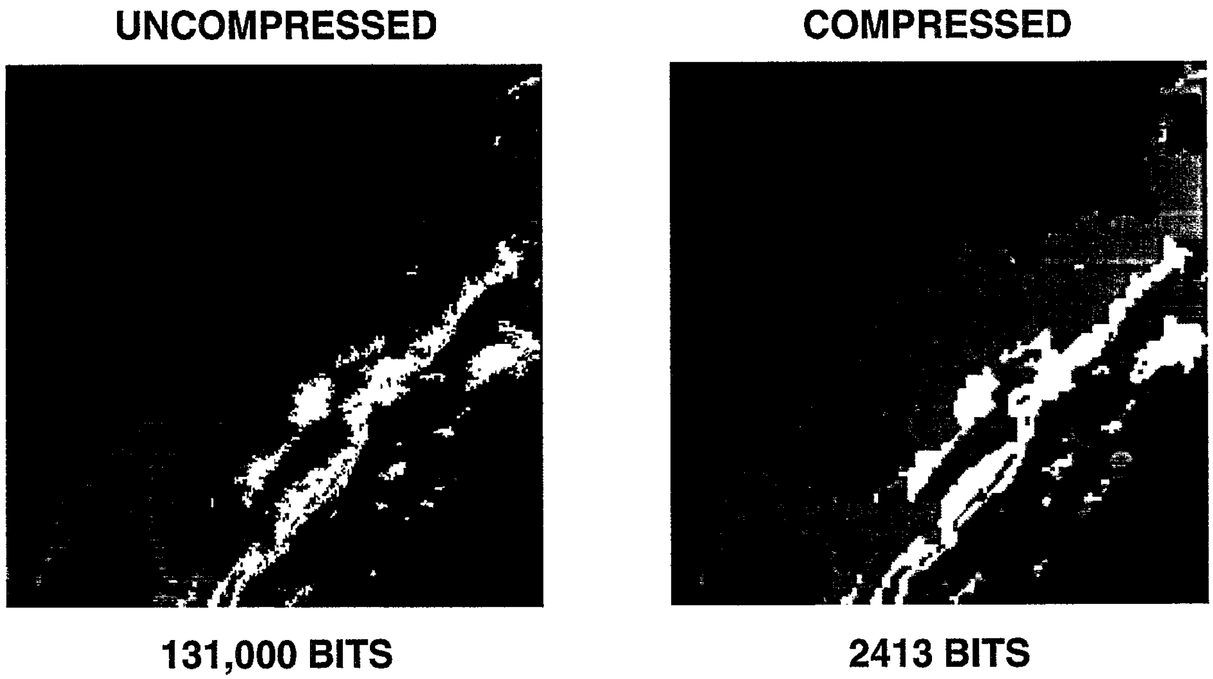

Figure 3 – Uncompressed NEXRAD and WH-compressed Precipitation Images.

A complete weather datalink product consists of more information beyond the basic graphical component. An unlabelled product graphic such as shown in Figure 3 above is not of much utility to a pilot. Among the additional necessary items of identification information included in the weather product would be:

(a) Product Identifier (ex. precipitation, icing, turbulence, etc.)
(b) Geographic reference
(c) Scale (number and size of pixels)
(d) Product Time

Depending upon the operational concept of the weather datalink product, these and similar data items may or may not need to be incorporated into the product message bit-stream. For example, the Graphical Weather Service (GWS) weather datalink described in section 2.2.3 is designed as a request-reply system. The pilot selects a particular weather product, geographic reference, scale, time, etc., in the downlink request message. The ground subsequently uplinks a response graphical image tailored to the pilot's specific request parameters, The GWS avionics match the stored pilot request parameters with the uplinked response graphic to complete the product – i.e., no additional information beyond the gridded graphical image is required in the uplink message. A second graphical weather product example is the Flight Information Service via Broadcast (FIS-B) system described in section 2.2.4 above. Since it is a broadcast-only system, FIS-B products must incorporate product identification information beyond the product graphical image in the uplinked message bits. FIS-B weather products include a time-stamp, product type specification, and geographic reference in the FIS-B product header record. Depending on the particular weather product, some additional information may be encoded or assumed by specified product convention.

3.3 Iconic Graphical Data

The term "iconic graphical" format refers here to the representation of a graphical image or other weather data using the parameters of a set of selected geometric figures (icons). The iconic graphical encoder generates and transmits the parameters of the particular icons to be represented. The iconic graphical decoder regenerates the display pixel grid or the weather information using the received iconic parameters. Iconic format is frequently the most efficient way to transmit graphical data. There are many examples of the use of the iconic graphical representation in weather data link applications.

The Polygon-Ellipse (PE) algorithm (section 3.2 above and reference 20) used to compress gridded graphical data is actually a form of iconic graphical presentation algorithm. Weather regions at a given NWS level are represented by PE either as filled polygons (transmitted as a sequence of vertex points) or filled ellipses (transmitted as two focus locations and a distance parameter). For weather images that are relatively simple (or can be reduced to a simple representation), the iconic approach of PE results in a very efficient way to represent the weather image.

The Integrated Terminal Weather System (ITWS) and Terminal Weather Information for Pilots (TWIP) applications make extensive use of iconic graphical formats. One example is the iconic representation of a microburst as a filled "lozenge" shape superimposed onto the background gridded graphical precipitation image. The microburst display icon is defined by the center points of two end circles and the circle radius. Since microbursts are typically circular (or extended circles), and since the pilot is mainly interested in the presence, location, and severity of the microburst -- using the iconic form maximizes the impact of the display without any unnecessary confusing details. A second example is the use of poly-lines (a sequence of connected line segments) to represent storm motion or a gust-front. A wind field might be displayed as an array of arrows whose direction gives the wind direction at that "pixel" and whose length indicates the wind speed. (Figure 4 below illustrates an iconic representation of microbursts and storm fronts from superimposed on a gridded graphical precipitation image background. The data was collected by the Atlanta "Terminal Doppler Weather Radar" (TDWR). The numbers in the microburst icons provide the windspeed change in knots due to the microburst. The right side of the figure illustrates how the image might be displayed in a textual format. Levels of precipitation are indicated with the symbols "-" for moderate and "+" for heavy. The letter "G" indicates gust fronts, "M" indicates microbursts, etc.)

Note that iconic weather datalink products may also require the additional identification information as described for gridded graphical weather products in section 3.2 above. As was the case for gridded graphics, an unlabelled icon is not useful information for a pilot in most cases.

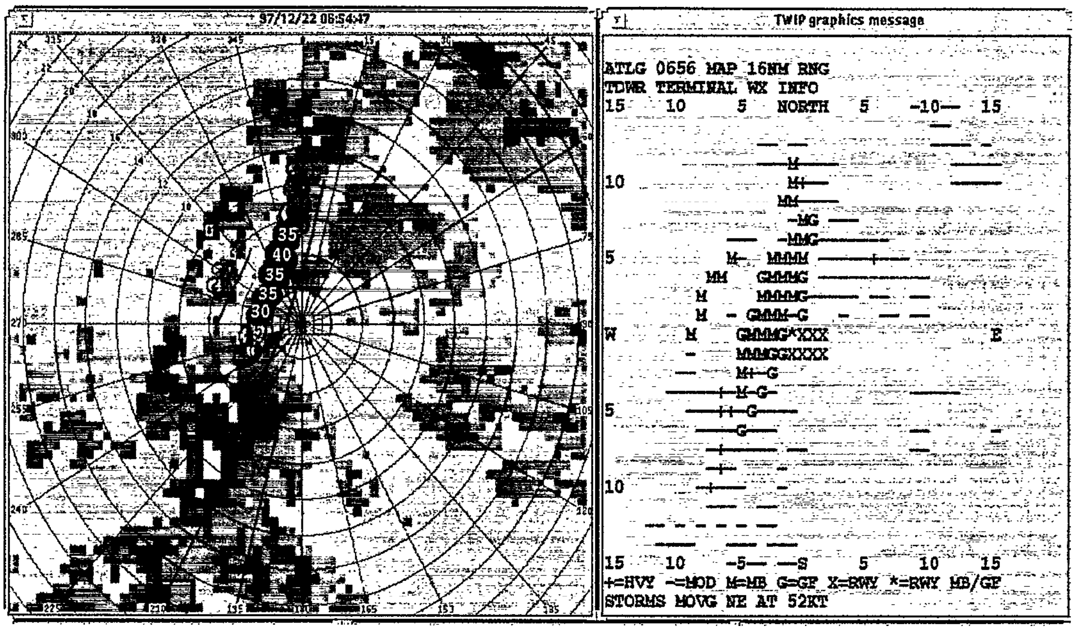

Figure 4 – Iconic Representations of Microbursts and Storm Fronts.

3.4 GRIB

The World Meteorological Organization (WMO) designed the GRIdded Binary (GRIB) general-purpose, bit-oriented data exchange format as an efficient means to transmit large volumes of gridded weather information via standard communications protocols (reference 21). By packing weather information into GRIB code, messages can be made more compact than existing textual bulletins. GRIB is also intended to be a data storage and archiving format. GRIB is extremely flexible -- it provides mechanisms for data formatting and storage that go far beyond simply display. It is quite a bit more complex than most other display formats described above. Software implementations of GRIB are available from several sources (NOAA, NCAR) over the Internet. Some weather data applications using GRIB are specified in the RTCA FIS-B documents described in section 2.2.4 above.

Each GRIB record contains a single weather parameter with values located at an array of grid points, or represented as a set of spectral coefficients, for a single layer of display, encoded in a bit stream. A GRIB record consists of six logical sections, two of which are optional. Each section contain an even number of bytes (GRIB enforces 16-bit word alignment). Zero padding bits are added where necessary to align the sections. The six sections of a GRIB record are the “Indicator Section” (IS), “Product Definition Section” (PDS), “Grid Description Section” (GDS) (optional), “Bit Map Section” (BMS) (optional), “Binary Data Section” (BDS), and the “End Section” (ES).

The 8-byte GRIB IS defines the record in a “human readable” way with the four letters “GRIB” in ASCII. The next 24 bits (3 bytes) give the total length of the GRIB message in bytes. The final byte of the GRIB IS holds the version number for the GRIB implementation (currently 1).

The GRIB PDS contains 40 or more bytes that serve to define the weather product. Among the fields in the GRIB PDS are the date-time of the observation, identification of the center where the data was gathered, time period between observations, the type of grid in use, the map projection in use (Lambert, Mercator, etc.), whether there is a GDS or BMS in this record, etc.

The GRIB GDS (when present) is used to define data geographic grids not already specified in the PDS (not the predefined GRIB defaults). The number of rows and columns are defined, as is the data point representation. GRIB allows for "quasi-regular" or "thinned" grids -- e.g., a latitude/longitude grid where the number of points in each row is reduced as one moves from the equator towards the poles.

The GRIB BMS (when present) is to either provide a bit map or a reference to a pre-defined bit map from the data gathering center. A 1 bit in the BMS map indicates that data for that grid point is contained in the GRIB BDS, while a 0 bit in the BMS map indicates that the grid point is missing. The bit map can be used to suppress data from irregular regions. As an example, consider data for "sea temperature". The bit map would be used to suppress data from grid points over land.

The GRIB BDS contains the packed data and scaling values needed to reconstruct the original data. GRIB allows each data point to be represented in a signed, scaled-integer format that allows a wide range of values to be efficiently represented without using floating-point. In a sense, this representation is a form of data compression. The BDS may be simply packed in a contiguous sequence, stored as spherical harmonic coefficients, or even a second-order complex packing.

The 4-byte GRIB ES defines a "human-readable" indication for the end of the GRIB record. It contains the four ASCII characters "7777".

Note that the GRIB specification incorporates the product identification information as described in section 3.2 above into the message encoding. The GRIB PDS contains this information (and more) -- at the expense of 40 or more bytes of message overhead. The GRIB optional GDS might also be required, at additional overhead. The designer of a particular weather datalink product must tradeoff the requirements for product identification information against the message overhead when selecting a format such as GRIB. There may be simpler and more-efficient ways to encode the particular product's identification information.

3.5 BUFR

The WMO designed the Binary Universal Format for the Representation of meteorological data (BUFR) to provide an efficient, general-purpose, bit-oriented data exchange format to transmit large volumes of point weather information via standard communications protocols. Like GRIB (section 3.4 above), BUFR is available, extremely flexible, and useful for both data display and archiving. BUFR is defined in reference 22.

Like GRIB, BUFR data consists of a sequence of six sections. The 8-byte BUFR "indicator section" defines the start of the record in a "human readable" way with the four letters "BUFR" in ASCII. The next 24 bits (3 bytes) give the total length of the BUFR message in bytes. The final byte holds the version number for the GRIB implementation (currently 1).

The BUFR "identification section" contains 18 or more bytes that serve to define the weather product. Among the fields in the section are the date-time of the observation and the identification of the center where the data was gathered.

The BUFR "optional section" may be used to add extra information about the originating data center.

The BUFR "data description section" provides the number of data subsets contained in the BUFR record. Flag bits indicate whether the data is actual weather observations or derived (forecast) data. Another flag bit indicates the use of data compression. Following these bits, the section contains the actual data points. Like GRIB, BUFR provides for packed and scaled values with a fixed binary format that is machine-independent. Unlike GRIB, a BUFR "point" may be a complex structure that could be an "icon" definition.

The 4-byte BUFR "end section" defines a "human-readable" indication for the end of the BUFR record. It contains the four ASCII characters "7777".

3.6 AWIN Format

The Aviation Weather Information (AWIN) format developed by NCAR has become a *de facto* industry standard for the transfer of integrated turbulence, icing, and convection weather gridded information (two or three-dimensional) to commercial weather service providers. The AWIN format has been developed to reduce datalink overhead and to permit value-added processing in avionics, such as vertical or horizontal profiling and flexible geo-referencing. The AWIN format has been proposed for incorporation into the FIS-B standard (see section 2.2.4).

The AWIN format uses a flat-Earth latitude/longitude projection. An AWIN weather file consists of an AWIN header record followed by the data for the volume of airspace. (The AWIN header fields are each 32-bits in size – either integer or IEEE-standard floating-point values. AWIN assumes the "big-endian" convention for the representation of all multi-byte values.) The AWIN header contains the following fields:

User Data Identifier (product selector)
Data Generation Time (seconds since 1/1/70)
Forecast-Valid Time (seconds since 1/1/70)
Number of Data Dimensions (2-D or 3-D)
Number of Grid Points in the Latitude Direction
Number of Grid Points in the Longitude Direction
Number of Altitude Slices (ignored for 2-D data)
Data Point Type (8/16/32-bit signed/unsigned integer, 32-bit floating-point)
Reserved 1,2, Integer
Lower-Left Comer Grid Latitude (degrees, floating-point)
Lower-Left Comer Grid Longitude (degrees, floating-point)
Delta Latitude Between Grid Points (degrees, floating-point)
Delta Longitude Between Grid Points (degrees, floating-point)
Data Scale Factor (floating-point)
Data Bias Factor (floating-point)
Missing Data Value (if data has this value, it is not valid)
Reserved 3,4 Floating-Point
Altitude Array (altitudes for 3-D slices, in feet)

The AWIN volume data immediately follows the AWIN header. The data in each altitude slice is given in the order indicated by the "Altitude Array" values in the AWIN header. Each altitude slice of data is padded as necessary to align on an S-byte boundary. The actual data values are computed using the formula:

True Data Value = (Transmitted Value * Scale) + Bias

where the scale factor and bias values come from the AWIN header.

Note that the AWIN format has no provisions for data compression or sparse data arrays. It has only one form of geographic reference and projection. Hence, it is more efficient, but less flexible than other "open standard" gridded representation techniques such as GRIB (see section 3.4 above). The AWIN format capabilities are similar to those of BUFR (see section 3.5 above).

3.7 Selecting an "Open Standard" Weather Product Format

As has been discussed in section 3 previously, there are a number of available "open standards" for various message formats that may be used to express weather datalink products. A summary of these "open standard" formats is shown in Table 3 below.

TABLE 3 – "Open Standards" for Datalink Weather Products

Acronym	Name	Source	Usage
ASN. 1	Abstract Syntax Notation version 1	ISO/IEC 882418825	Text (1)
DLAC	Data Link Applications Coding	ICAO SICASP	Text
DEFLATE	---	Internet RFC 1951	Text (2)
GRIB	Gridded Binary Format	WMO	Gridded
BUFR	Binary Universal Format for the Representation of meteorological data	WMO	Griddedkonic
AWIN	Aviation Weather Information	*de facto industry (NCAR)*	Gridded
PE	Polygon-Ellipse compression	MIT/LL Patent 5,363,107	Gridded
WH	Weather Huffmau compression algorithm	MIT/LL ATC-261	Gridded

(1) May be applied to general format specification
(2) May be used for general data compression

These formats tend to fall into three basic types: text (section 3.1), gridded graphical (sections 3.2, 3.4, 3.5 and 3.6), and iconic (section 3.3). Table 4 below indicates the format type for a set of weather products that are suitable for datalink to an aircraft. Note that a number of these products could be expressed in multiple ways, depending on the design and operational-concept of the particular application.

TABLE 4 – Formats for Datalink Weather Products

Weather Product Type	Product Format
AIRMET	Text
ATIS	Text
Cloud Cover (Satellite Imagery)	Gridded
Echo Tops	Gridded
Gust Front	Iconic
Icing Forecast	Gridded
Lightning	Gridded/Iconic
METAR	Text/Iconic
Microburst	Iconic
PIREP	Text
Radar Precipitation	Gridded
SIGMET	Text
Storm Motion	Iconic
TAF	Text
Turbulence Warning	Gridded/Iconic
TWIP	Text
Volcanic Ash	Gridded/Iconic
Wind Field	Gridded (vector)

As was described in section 3.1 above, there are several “open system” approaches to textual format expression. If the message syntax can be generally reduced to a set of “fill in the blanks” (a “phrasebook” approach) and highly constrained, then expression of the messages in ASN.l is likely optimal in terms of bandwidth efficiency. The use of ASN. 1 combined with PER encoding rules yields minimum bit usage so long as the message syntax is defined in known standards (ex. ATIS, METAR, TAF). The use of BER encoding rules provides for complete syntax flexibility, but at the expense of bit efficiency. Where the message syntax is not well-constrained (ex. PIREP, SIGMET), ASN. 1 expression defaults to ASCII text. If compression is desired for these types of textual messages, then either DEFLATE or DLAC techniques are recommended. As was stated in section 3.1 above, DEFLATE works best for large volumes of text in a given message with significant textual redundancy. It can handle highly-abbreviated aviation text. DLAC works well when the text contains fully spelled out words from its aviation dictionary. It will not compress abbreviations or messages whose contents are not drawn largely from the DLAC aviation dictionary. Decisions about which textual expression technique would be appropriate for a given weather product would need to consider the degree of syntax “constrainability”, the need for compression (or bit efficiency), and the trade-off between full-word and abbreviation usage.

There are several “open system” approaches to gridded graphical format expression. GRIB (section 3.4), BUFR (section 3.5), and AWIN (section 3.6) are alternative gridded techniques incorporating complete expression of weather product header information (time, site location, etc.) along with concise representations of gridded data sets in 2 or 3 dimensions. GRIB is the most complete and flexible of these techniques, while AWIN is the most bit-efficient (at the expense of flexibility). BUFR lies between the others on the flexibility-versus-efficiency scale. None of these techniques provides for significant data compression of the gridded data. GRIB and BUFR provide a technique called "second-order packing" which provides a degree of lossless compression by determining the range of grid point values and choosing an optimally-sized grid point encoding. GRIB does allow for spherical harmonics as a form of lossy compression. A stream-compression algorithm (such as DEFLATE, as described for textual formats above) could be applied to any of the gridded graphical techniques to achieve some degree of compression. The WH algorithm (described in section 3.2) provides for a high-degree of data compression, either lossless or lossy. However, WH deals only with the grid data points – an additional format would be required to include to incorporate the weather product header information. Alternatively, WH could be incorporated within a modified BUFR specification. Note that WH was developed with weather products similar to radar precipitation images in mind. If the weather product requires a large range of pixel values (beyond 8-16 – such as might be required for Cloud Cover images), or if the weather product image is not generally composed of lower-level "blobs" surrounding higher-level "blobs", then WH will not achieve high compression ratios. Further work will be required to develop a complete "open standard" technique to express gridded graphical data if optimal bitwise compression is to be included.

Note that a product like "wind field" actually requires a gridded vector format. Each point in the grid actually consists of two component parts – a scalar magnitude and a direction. Gridded vector products could be expressed using GRIB, BUFR, or AWIN by subdividing each point value into two subfields. Further work will be required to specify a fully "open standard" technique for the expression of gridded vector products. Table 5 below summarizes the tradeoffs involved with the various gridded graphical formats.

TABLE 5 – Gridded Graphical Format Tradeoffs

"Open Standard"	Product Header	3-d Grid	Sparse Grid	Complex Grid Point	Grid Point Scaling	Lossless Compression	High Compression Ratio (Lossy)
GRIB	Yes	Yes	Yes	No	Yes	Yes (1)	No (2)
BUFR	Yes	Yes	No	Yes	Yes	Yes (1)	No
AWIN	Yes	Yes	No	No	Limited (3)	No	No
PE	No	No	No	No	No	No	Yes (4)
WH	No	No	No	No	No	Yes (5)	Yes (4)

(1) Second-order packing technique reduces grid point range and selects "optimal" point scaling
(2) Spherical-harmonics algorithm
(3) Can select 8,16, or 32-bit integers or 32-bit floating-point grid points
(4) Compression ration greater than 50-l (can approach 100-l)
(5) Approaches 10-1 compression ratio --

There are currently no complete "open system" approaches to the expression of iconic graphics. (BUFR could be used to express iconic forms, if the icon definitions were made part of the standard.) Each form of "icon" has been developed ***ad hoc*** for its individual application. The only sort-of "open standard" technique for the expression of certain types of iconic representation (the PE algorithm in section 3.3 above) can only deal with certain types of icons (closed polygons and ellipses). PE (like WH) would need to be augmented with an additional format to incorporate the weather product header information. Further work will be required to design a more-capable "open standard" technique for the expression of iconic data. This technique will need the capability of defining a wide range of iconic forms. (For example, icons such as arrows, open polygonal lines, text labels, etc., would need to be incorporated into the iconic "toolkit".) The iconic technique will also need the general weather product header mechanisms from the gridded techniques such as GRIB or AWIN. One approach to the generation of an iconic format standard would be to extend BUFR to allow specification of an iconic "object" as a data element.

4. Data Link Security

There are issues of link security that bear on the design of any data link. None of the aviation weather data link systems currently provide for any form of link security, since they consider weather information to be "non-critical", "information-only" data. Data link protocols such as OSI used by the ATN (section 2.2.2), however, do provide mechanisms to support link security functions. These security functions have three areas of operation:

(1) Protect the data from unauthorized access
(2) Protect the data from unauthorized modification or tampering
(3) Ensure that the recipient of the data knows what entity sent it

These functions are generally provided through the use of encryption and digital signature (or message digest) algorithms.

In the OSI model used in the ATN, security functions are provided by a "System Management Entity" (SME) application operating above the protocol stack. Hooks are provided in the lower-layer OSI protocols to pass the necessary indicators and keys. Details for the SME and these protocol hooks are defined in reference 3.

The IETF has recently begun to address issues of TCP/IP link security, since the Internet now routinely deals with private transactions. A number of documents dealing with security provisions for TCP/IP have been drafted. RFC 2406 "The Encapsulating Security Payload (ESP)" protocol provides a means for two end-systems to send and receive IP packets using encryption to keep the contents secure. RFC 2402 "IP Authentication Header (AH)" protocol provides for two end-systems to use digital signatures to authenticate IP packets. RFC 2408 "Internet Security Association and Key Management Protocol (ISAKMP)" provides a framework by which end-systems can exchange security parameters over IP. Finally, RFC 2246 "The TLS Protocol Version 1.0" defines a transport-layer security mechanism for processes to operate securely above the TCP/IP protocol stack. TLS provides for efficient cryptographic security between end-systems. The TLS "Record Protocol" provides for symmetric encryption and also for message digests to ensure the integrity of data. The TLS "Handshake Protocol" operates above the TLS Record Protocol and is used to negotiate the parameters of the secure connection and to transfer keys.

5. Conclusions

This paper has sought to describe some of the major technical issues involved in the architecture and design of a weather information data link system. Section 2 of this paper deals with weather data link communications protocols. Section 2.1 sets forth a set of desired features and requirements for communications protocols to be used for weather data link applications. Section 2.2 of this paper gives some capsule summaries of existing and experimental weather data link systems. Note that none of these systems completely provides for the desired feature set. TCP/UDP/IP was selected as the closest match to the desired features and requirements of a weather data link. In particular, TCP/UDP/IP is shown in section 2.3.1 to provide the following:

(a)TCP/UDP/IP is a readily available, open standard
(b) TCP/UDP/IP efficiently handles both long and short messages
(c) UDP/IP supports broadcast and multicast messaging
(d) TCP/IP supports reliable addressed messaging
(e) TCP/IP contains a mechanism for high-priority interrupt messages
(f) TCP/UDP/IP handles mobile routing

TCP/UDP/IP is shown in section 2.3.2 to fall short of the desired weather data link feature set in the following:

(a) Provision for link-independent / policy routing
(b) Provisions for long-latency networks

Methods to work around or minimize these shortcomings are described.

Section 3 of this paper deals with weather data link applications and data formatting design issues. Textual, gridded graphic, and iconic graphic weather applications are discussed. The GRIB, BUFR, and AWIN formats developed especially for weather data are outlined. Compression and display algorithms developed for each type of weather data link application are covered. It is concluded that while there are "open standard" techniques appropriate for textual applications, the existing "open standard" techniques for gridded and iconic graphic applications are insufficient. In the gridded graphical area, there needs to be a combination of the flexibility and expressiveness of formats such as GRIB with the efficient compression performance of techniques such as WH. In the iconic graphical area, BUFR would need to be extended with a library of iconic forms appropriate for weather datalink applications.

Section 4 of this paper deals with issues of data link security. While data link security is not currently considered a problem for weather data dissemination, it could become one in the future. TCP/IP and OSI mechanisms for protection from unauthorized access, modification, and for authentication of data link messages are described.

In summary, it is argued that an efficient, effective, flexible, and reliable aviation data link system for the dissemination of weather information may be constructed from a set of existing standards and techniques. There is little need to "reinvent" -- the designs and system architectures already exist to provide pilots with useful and timely weather "situational awareness" information.

Acronyms

AAC:	Aeronautical Administrative Communications
ACARS:	Airline Communications and Reporting System
ADLP:	Airborne Data Link Processor
ADS:	Automatic Dependent Surveillance
AIRMET:	AIRman's METeorological information
AOC:	Aeronautical Operational Control
APC:	Aeronautical Passenger Communications
ARINC:	Aeronautical Radio INCorporated
ASCII:	American Standard Code for Information Interchange
ASN.1:	Abstract Syntax Notation version 1
ATC:	Air Traffic Control
ATIS:	Automatic Terminal Information Service
ATN:	Aeronautical Telecommunications Network
AUTOMET:	AUTOmated Meteorological Transmission
AWIN:	Aviation Weather INformation
BDS:	Binary Data Section (GRIB)
BER:	Basic Encoding Rules (ASN. 1)
BMS:	Bit Map Section (GRIB)
BUFR:	Binary Universal Format for the Representation of meteorological data
COTS:	Commercial Off-The-Shelf
CPDLC:	Controller-Pilot Data Link Communications
CRC:	Cyclic Redundancy Check
DLAC:	Data Link Applications Coding
ELM:	Extended Length Message (Mode S)
ES:	End Section (GRIB)
FIS:	Flight Information Service
GA:	General Aviation

GDLP:	Ground Data Link Processor
GDS:	Grid Description Section (GRIB)
GICB:	Ground-Initiated Comm-B
GRIB:	GRIdded Binary
GWS:	Graphical Weather Service
HDLC:	High-level Data Link Control
HF:	High Frequency radio
ICAO:	International Civil Aviation Organization
IDRP:	Inter-Domain Routing Protocol
IETF:	Internet Engineering Task Force
IP:	Internet Protocol
IS:	Indicator Section (GRIB)
ISO:	International Organization for Standards
ITWS:	Integrated Terminal Weather System
MASPS:	Minimum Aviation System Performance Standard (RTCA)
METAR:	METeorological Aviation Report
MOPS:	Minimum Operational Performance Standard **(RTCA)**
MSP:	Mode S-Specific Protocol
NCAR:	National Center for Atmospheric Research
NEXRAD:	Next Generation Weather Radar
NOAA:	National Oceanographic and Atmospheric Administration
NOTAM:	NOTice to AirMen
NWS:	National Weather Service
OSI:	Open Systems Interconnection
PDS:	Product Definition Section (GRIB)
PE:	Polygon-Ellipse weather compression algorithm
PER:	Packed Encoding Rules (ASN. 1)
PIREP:	PIlot REPort
RFC:	Request For Comment

RTCA:	Requirements and Technical Concepts for Aviation
SARPS:	Standards and Recommended Practices (ICAO)
SC:	Special Committee (RTCA)
SICASP:	SSR Improvements and Collision Avoidance Panel (ICAO)
SIGMET:	SIGnificant METeorological infomation
SLM:	Standard Length Message (Mode S):
SME:	System Management Entity (OSI)
SSR:	Secondary Surveillance Radar
TAF:	Terminal Aerodrome Forecast
TCP:	Transmission Control Protocol
TDWR:	Terminal Doppler Weather Radar
TLS:	Transport Layer Security
TWIP:	Terminal Weather Information for Pilots
TWS:	Textual Weather Service
UDP:	User Datagram Protocol
VDL:	VHF Data Link
WH:	Weather-Huffman compression algorithm
WMO:	World Meteorological Organization

References

(1) ARINC Document 619-1, "ACARS Protocols for Avionic End Systems", January 1997.

(2) ARINC Document 622-3, "ATS Data Link Applications Over ACARS Air-Ground Network", September 1998.

(3) ICAO "Manual of Technical Provisions for the Aeronautical Telecommunications Network (ATN)", 1st edition 1998, document 9705.

(4) ICAO "Manual of the Secondary Surveillance Radar (SSR) Systems", 1st edition 1997, document 9684-AN951.

(5) "The Weather-Huffman Method of Data Compression of Weather Images", J. L. Gertz, Lincoln Laboratory Project Report ATC-261, October 1997.

(6) "Minimum Aviation System Performance Standards (MASPS) for Flight Information Services-Broadcast (FIS-B) Data Link", RTCA Special Committee 195, Draft version 4.0, May 1999.

(7) ICAO SICASP "Manual on Mode S Specific Services", First Edition 1997.

(8) RTCA SC-195 "Automated Meteorological Transmission (AUTOMET) MOPS", DRAFT July 1999.

(9) "iChip S7600A", Seiko Instruments, Torrance, CA.

(10) "TCP/IP Illustrated, Volume 1 -- The Protocols", W. Richard Stevens, Addison-Wesley Publishing, 1994.

(11) IETF RFC 2507 "IP Header Compression".

(12) IETF RFC 2002 "Mobile-IPv4 Specification".

(13) Discussion with Ted Signore of MITRE-CAASD, June 18, 1999.

(14) "Aeronautical Information Manual (AIM)", U.S. Department of Transportation Regulations, ASA-96-FR-AM-BK, 1996

(15) IETF RFC 195 1 "DEFLATE Compressed Data Format Specification version 1.3", May 1996.

(16) "Aeronautical Data Link Design Guidelines", DOT/FAA/SA-90/1.

(17) ISO/IEC 8824-l: 1995 Information Technology – Abstract Syntax Notation One (ASN. 1): Specification of the Basic Notation.

(18) ISO/IEC 8825-2: 1996 Infornation Technology – ASN.1 Encoding Rules – Part 2: Specification of Packed Encoding Rules (PER).

(19) "Encoding Approaches for Data Link Transmission of Weather Graphics", J. L. Gertz and R. D. Grappel, Lincoln Laboratory Project Report ATC-205, December 1993.

(20) "Storage and Transmission of Compressed Weather Maps and the Like", J. L. Gertz and R. D. Grappel, U.S. Patent 5,363,107.

(21) "The WMO Format for the Storage of Weather Information and the Exchange of Weather Product Messages in Gridded Binary Form", J. D. Stackpole, NWS, NOAA, February 1994 (updated March 1997).

(22) "A Guide to the WMO Code Form FM 94 BUFR", W. Thorpe.