# AEROSPACE INFORMATION REPORT

| AIR5645 | REV. A |
|---|---|

Issued 2007-12
Stabilized 2014-09

Superseding AIR5645

## JAUS Transport Considerations

### RATIONALE

AIR5645 provides technical background information of potential relevance to the development of additional SDPs. As such, it is not dynamic in nature, but retains relevance. Updates are not likely to be required in the near term. Stabilization is indicated.

### STABILIZED NOTICE

This document has been declared "Stabilized" by the SAE AS-4 JAUS Joint Architecture for Unmanned Systems Committee and will no longer be subjected to periodic reviews for currency. Users are responsible for verifying references and continued suitability of technical requirements. Newer technology may exist.

**TO PLACE A DOCUMENT ORDER:** **Tel: 877-606-7323 (inside USA and Canada)**
**Tel: +1 724-776-4970 (outside USA)**
**Fax: 724-776-0790**
**Email: CustomerService@sae.org**
**SAE WEB ADDRESS: http://www.sae.org**

**SAE values your input. To provide feedback on this Technical Report, please visit http://www.sae.org/technical/standards/AIR5645A**

TABLE OF CONTENTS

# 1. SCOPE

This SAE Aerospace Information Report (AIR) discusses characteristics of data communications for the Joint Architecture for Unmanned Systems (JAUS). This document provides guidance on the aspects of transport media, unmanned systems and the characteristics of JAUS itself that are relevant to the definition of a JAUS transport specification.

## 1.1 Purpose

The purpose of this document is to facilitate interoperation of unmanned vehicle systems, subsystems, and payloads by indicating relevant characteristics of transport media, unmanned systems and of JAUS itself, so that new transport standards may conform to expected performance.

## 1.2 JAUS Document Organization

The Joint Architecture for Unmanned Systems (JAUS) defines a suite of standards for use in the research, development, design, acquisition and deployment of Unmanned Systems. The JAUS Standards are divided into four essential scopes: the Domain Model, the Reference Architecture, the Transport Specification, and the Compliance Specification. Each scope is individually documented.

### 1.2.1 Domain Model

The Domain Model [JAUS-DM] is an abstract model of operational requirements that must be supported by a given unmanned system. This model is a tool to be used by a developer of an unmanned system to better understand the requirements of the acquirer of that unmanned system. The Domain Model is written in language understandable by both the developer and the acquirer. This line of documents is also known as the Architecture Framework for Unmanned Systems, or AFUS.

### 1.2.2 Reference Architecture

The Reference Architecture [JAUS-RA] is the technical specification used to implement unmanned systems in compliance with the JAUS standards and to assess technical compliance with the Standard. The most recent version of the Reference Architecture [JAUS-RA] is itself divided into three major sections: the Architecture Framework, the Message Definition, and the Message Set.

#### 1.2.2.1 Architecture Framework

The Architecture Framework volume of the Reference Architecture provides a description of the structure of systems based on the Reference Architecture. This document serves as the primary mapping of Domain Model requirements to the JAUS message set.

#### 1.2.2.2 Message Definition

The Message Definition volume of the Reference Architecture specifies the current JAUS message header and the types of messages employed. This document focuses on the rules of messaging as opposed to the domain-specific semantics contained in the message set.

The JAUS message header defined by this document is an application-layer header, and is specified without reference to any particular transport medium. JAUS Transport Standards note how implementations may make use of certain fields within this header to provide more efficient communications. However, transport implementations are not required to do so in order to maintain correctness or interoperability.

#### 1.2.2.3 Message Set

The Message Set volume of the Reference Architecture specifies the domain-specific messages and their exact content.

### 1.2.3 Transport

JAUS Transport defines communication services for JAUS messages. These services are the means by which messages are conveyed from one JAUS entity to another.

There are currently two documents relevant to the discussion of JAUS message transport: the JAUS Transport Considerations Report (this document) and the JAUS Transport Specification [JAUS-TS].

#### 1.2.3.1 JAUS Transport Considerations Report

The JAUS Transport Considerations report (this document) discusses aspects of transport media, unmanned systems and widely used transport protocols, as well as characteristics of JAUS itself. This document then identifies characteristics of concern in the implementation of a compliant and efficient JAUS Transport standard. In this respect, this document serves as a meta-specification – in effect specifying the standards for a JAUS Transport standard.

This document may also be viewed as a summary of “lessons learned” by individuals involved in communications in unmanned systems. As the use of unmanned systems increases, this report will be updated and revised, as the experience of deploying more numerous, more autonomous unmanned systems increases.

#### 1.2.3.2 JAUS Transport Specification

The JAUS Transport Specification [JAUS-TS] defines a family of transports for conveying JAUS messages between different JAUS entities via different link-layer implementation technologies. These transports are designed to incorporate the concerns outlined in the JAUS Transport Considerations Report.

### 1.2.4 Compliance Specification

The Compliance Specification [JAUS-CMP] defines compliance with JAUS. Since JAUS is a message-based architecture, compliance with JAUS is assessed by verifying that the message traffic between JAUS entities satisfies the definitions and requirements of the JAUS standards.

## 1.3 Field of Application

The Field of Application for the Transport Considerations report is data communications between nodes on a computer network.

## 2. REFERENCES

### 2.1 Applicable Documents

[JAUS-RA] JAUS Working Group, "Reference Architecture Specification," Version 3.3, Huntsville, AL, June 27, 2007.

[JAUS-TS] SAE AS-4B & JAUS Working Group, “JAUS Transport Specification,” Version 1.0, SAE AS5669.

[JAUS-DM] JAUS Working Group and SAE Aerospace Committee AS-4 Unmanned Systems, “JAUS History and Domain Model,” SAE Aerospace Document AIR5664, Revision 1.0, Society of Automotive Engineers, Warrendale, PA, January 2006.

[JAUS-CMP] JAUS Working Group, "Compliance Specification (CS)," Version 1.2 , Huntsville, AL, 25 October 2006.

[X3.28] American National Standards Institute, "Procedures for the Use of the Communication Control Characters of American National Standard Code for Information Interchange in Specified Data Communication Links," ANSI X3.28-1976, December 1975.

[RFC1661] Simpson, W., "The Point-to-Point Protocol (PPP)," RFC-1661, Daydreamer, July 1994.

[RFC1662] Simpson, W., "PPP in HDLC-like Framing," RFC-1662, Daydreamer, July 1994.

[RFC1994] Simpson, W., “PPP Challenge Handshake Authentication Protocol (CHAP),” RFC-1994, DayDreamer, August 1996.

[RFC2284] Blunk, L. and J. Vollbrecht, “PPP Extensible Authentication Protocol (EAP),” RFC-2284, Merit Network, Inc, March 1998.

[RFC1055] Romkey, J., "A Nonstandard for Transmission of IP Datagrams over Serial Lines: SLIP," RFC-1055, June 1988.

[RFC914] Farber, David et al, "A Thinwire Protocol for connecting personal computers to the INTERNET," RFC-914, September 1984.

[RFC768] Postel, J., "User Datagram Protocol," RFC-768, 28 August 1980.

[RFC791] Information Sciences Institute, “Internet Protocol,” RFC-791, September 1981

[RFC793] Information Sciences Institute, "Transmission Control Protocol," RFC-793, September 1981.

[RFC794] Cerf, V., "Pre-emption," RFC-794, September 1981.

[RFC813] Clark, David D., "Window and acknowledgment strategy in TCP," RFC-813, July 1982.

[X.25] International Telecommunication Union, "Interface between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE)," ITU-T Recommendation X.25, October 1996.

[TECHDEV] National Academy of Sciences, "Technology Development for Army Unmanned Ground Vehicles," National Academies Press, Washington, 2003.

[FLET82] Fletcher, John G., "An Arithmetic Checksum for Serial Transmissions," IEEE Transactions on Communications, January, 1982, IEEE, New York.

[KODIS92] Kodis, John, "Fletcher's Checksum," Dr. Dobb's Journal, May 1992.

[KOOP04] Koopman, Philip and Tridib Chakravarty, "Cyclic Redundancy Code (CRC) Polynomial Selection for Embedded Networks," International Conference on Dependable Systems and Networks (DSN-2004).

[SAXENA90] Saxena, Nirmal R. and Edward J. McCluskey, "Analysis of Checksums, Extended-Precision Checksums and Cyclic Redundancy Checks," IEEE Transactions on Computers, vol. 39 no.7, July 1990, IEEE, New York.

[PEREZ83] Perez, Aram, "Byte-Wise CRC Calculations," IEEE Micro, June 1983, IEEE, New York.

[WILLIA93] Williams, Ross N., "A Painless Guide to CRC Error Detection Algorithms," Rocksoft, Adelaide, Australia, August 1993 [available via ftp: ftp.adelaide.edu.au/pub/rocksoft/crc_v3.txt]

[RAMABA88] Ramabadran, Tenkasi V. and Sunil S. Gaitonde, "A Tutorial on CRC Computations," IEEE Micro, August 1988, IEEE, New York.

[RFC1950] RFC-1950 ZLIB 3.3 Specification.
Defines a lossless compressed data format. The format currently uses the DEFLATE compression method.

[RFC1951] RFC-1951 DEFLATE 1.3 Specification
Defines a lossless compressed data format that compresses data using a combination of the LZ77 algorithm and Huffman coding, with efficiency comparable to the best currently available general-purpose compression methods.

[RFC1952] RFC-1952 GZIP 4.3 Specification
Defines a lossless compressed data format that is compatible with the widely used GZIP utility. The format currently uses the DEFLATE compression method.

[RFC3095] Bormann, C. (ed.), "RFC-3095 - Robust Header Compression (RoHC): Framework and Four Profiles," 2001.

[RFC1144] Jacobson, V., "RFC-1144 - Compressing TCP/IP Headers for Low-Speed Serial Links," February 1990.

[SUTT91] Sutterfield, Robert A., "Low-Cost IP Connectivity," Sun User Group, San Jose, CA, 9 December 1991.

2.2 Other Applicable References

[COMER96] Comer, Douglas, "Internetworking with TCP/IP," Volume 1, Prentice-Hall, New York, 1996

[TANEN96] Tanenbaum, Andrew S., "Computer Networks," Third Edition, Prentice-Hall, New York, 1996

[STEVENS93] Stevens, W. Richard, "TCP/IP Illustrated," Volume 1, Addison Wesley, New York, 1993.

[DOUGLA99] Douglass, Bruce Powel, "Custom Embedded Communications Protocols," (Whitepaper), I-Logix, Inc., Andover, MA, 1999.

[HOLZ91] Holzmann, Gerard J., “Design and Validation of Computer Protocols”, Prentice Hall, New Jersey, 1991.

[ZCONF05] Cheshire, Stuart & Steinberg, Daniel H., “Zero Configuration Networking: The Definitive Guide”, O’Reilly, Sebastopol, CA, 2005.

## 2.3 Acronyms

ACK
Acknowledgment (positive)
Generally, a byte or message sent to acknowledge successful receipt of a packet or message. Specifically, the ASCII control character octet of value 0x06.

ADCCP
Advanced Data Communication Control Procedures
ADCCP is an ANSI standard derived from IBM’s SDLC (Synchronous Data Link Control). As such, ADCCP is a bit-oriented synchronous data-link layer protocol, and is very similar to the ISO HDLC (High-level data link control) standard, which also derives from SDLC.

ANSI
American National Standards Institute
ANSI is a non-governmental, non-profit standards organization, overseeing the development and administration of technical standards in the United States. ANSI is the official U.S. representative to the ISO.

ARQ
Automatic Repeat reQuest
A transport protocol is said to be an ARQ protocol if it uses positive and negative acknowledgements and sender timeouts, and further employs packet replacement on request to overcome packet loss or corruption over noisy or unreliable channels.

ASCII
American Standard Code for Information Interchange
ASCII is a character encoding for digital representation of the English alphabet, for representation of text in computers and digital communications.

CRC
Cyclic Redundancy Check
An error check algorithm based on polynomial division over a binary field.

CRC-CCITT
A specific CRC, defined by its polynomial, initial conditions for accumulation, and terminal conditions, as defined by ITU Recommendation X.25 (which was once CCITT recommendation X.25).

CPPP
Compressed Point-to-Point Protocol
A variant of PPP (point-to-point protocol) incorporating a form of TCP header compression (Van Jacobsen header compression).

CSLIP
Compressed Serial Line Internet Protocol
A variant of SLIP (serial line internet protocol) incorporating a form of TCP header compression (Van Jacobsen header compression).

DHCP
Dynamic Host Configuration Protocol
DHCP is a protocol (set of rules) used by network clients to obtain an IP address and other networking parameters (such as the subnet mask and the IP addresses of DNS servers and default gateway) from a server providing network management services.

DLE
Data Link Escape
An ASCII control character, an octet of value 0x10, this character is used in the implementation of data transparent protocols such as those consistent with [X3.28].

DNS
Domain Name System
A fundamental element of internet infrastructure, the Domain Name System translates human-readable hostnames to IP addresses.

EMI
Electromagnetic Interference
Interference by electromagnetic signals that can cause reduced data integrity and increased error rates on communication channels.

FIFO First-In, First-Out
This term describes a simple queuing technique in which requests, data, or processes are handled in order of arrival; the term also is used as a noun, describing an instance or implementation of a queuing mechanism with that handling behavior.

HMI Human-Machine Interface
See “OCU – Operator Control Unit”

IANA Internet Assigned Numbers Authority
The IANA is responsible for assignment of IP addresses, top level domains and Internet protocol code point allocations (such as port assignments). The IANA assignment data is maintained on the IANA website, http://www.iana.org.

IETF Internet Engineering Task Force
The engineering body responsible for technical specifications, definitions and direction for the continued development of internet technologies.

IP Internet Protocol
The internet layer of the TCP/IP protocol suite stack defining the packet format for message packets to be sent across an internet, and the protocol for delivering those packets to their intended destination. The protocol is Internet Protocol (IP); the packet defined is the IP datagram. See [RFC791].

IPS Internet Protocol Suite
The protocol suite defines network communications using the TCP/IP family of protocols.

ISO International Organization for Standardization
A non-governmental organization that leverages the activities of the national standards organizations of 146 countries, ISO is the largest developer of technical standards in the world.

JAUS Joint Architecture for Unmanned Systems
A suite of standards used in the research, development, design, acquisition and deployment of Unmanned Systems.

JPEG Joint Photographic Experts Group
This committee created a standard of the same name, defining means of encoding and compressing image data into a stream of bytes, and of decompressing that stream into an image; both lossy and lossless compression techniques are defined. The standard also defines the file formats for storing the stream of bytes.

MSN Message Sequence Number
A unique identifying “serial number” assigned to packets as transmitted; the MSN is typically used in the detection of missed messages, in providing assurance of correct sequence of delivery, and in the requesting of retries when multiple outstanding messages are supported.

MTU Maximum Transfer Unit
The Maximum Transfer Unit (MTU) is a term for the size of the largest datagram that can be passed by a layer of a communications protocol.

NAK Acknowledgement (negative)
Generally, a byte or message sent to indicate unsuccessful receipt or non-receipt of a packet or message. Specifically, the ASCII control character octet of value 0x15.

OCU Operator Control Unit
A device by means of which a human operator may control an Unmanned System.

OSI

Open Systems Interconnect
A data communications model developed by ISO to assure communications interoperability across disparate systems.

PDU

Protocol Data Unit
An application message (for the purposes of this specification, this will be assumed to be a JAUS message) being propagated down the protocol stack on the sender side, or up the protocol stack on the receiver side.

PPP

Point-to-Point Protocol
An encapsulation protocol for sending IP datagrams across serial communications links, PPP also incorporates strong blockchecks, control protocols and a high degree of configurability. [RFC1661]

RF

Radio Frequency
Electromagnetic energy or signaling based thereon whose frequency is normally associated with radio wave propagation.

SLIP

Serial Line Internet Protocol
An encapsulation protocol for sending IP datagrams across serial communications links. [RFC1055]

TCP

Transmission Control Protocol
A reliable, connection-oriented message delivery protocol defined by [RFC793] and related IETF documents.

UAV

An unmanned aerial vehicle; may be teleoperated or autonomous.

UGV

An unmanned ground vehicle; may be teleoperated or autonomous.

USV

An unmanned surface-of-water vehicle; may be teleoperated or autonomous.

UUV

An unmanned undersea vehicle; may be teleoperated or autonomous.

UDP

User Datagram Protocol
An unreliable best-effort connectionless message delivery protocol defined by [RFC768] and related IETF documents.

XML

Extensible Markup Language
A general purpose markup language providing users the ability to define their own tags. XML is a simplified subset of an earlier markup language, SGML (Standard Generalized Markup Language).

## 3. CONTEXT

The context for JAUS Transport is network data communications. The discussion of any transport standard in the context of network data communications should include both:

- The Protocol Stack for the standard
- The Compliance Regime against which the standard may be evaluated.

Section 3.1 provides a discussion of network communications protocol stacks and their associated models. Following discussion of the OSI and Internet protocol models, the JAUS protocol model is introduced, and the roles and responsibilities of the JAUS Transport Layer described. Concluding 3.1, several key structures of the JAUS Transport Layer are identified and discussed.

Section 3.2 provides information regarding how JAUS Transport fits the compliance models developed for JAUS.

### 3.1 Protocol Suite Models

The Field of Application for JAUS transport is data communications between nodes on a computer network. Within this field of application, it is typical to describe a standard in terms of a suite of protocols which form a protocol stack for the standard.

Several models of the suites of protocols involved in such data communications already exist. The following subsections briefly describe two of the most widely used models: the seven-layer protocol stack of the OSI (Open Systems Interconnect) model and the four-layer protocol stack of the IPS (Internet Protocol Suite) model (commonly referred to as the TCP/IP protocol stack model). Section 3.1.1 presents the seven-layer OSI Model. In 3.1.2, the IPS Model is presented, and the mapping of OSI Model layers to IPS Model layers is discussed. The final subsection (3.1.3) describes the protocol stack assumed by the JAUS Standards, and describes its layers in terms of the preceding two models.

### 3.1.1 The OSI Model

The Open Systems Interconnect (OSI) model, developed by the International Standards Organization (ISO), is a seven-layer protocol model. Each layer represents a different level of abstraction, from bits on a medium (such as a wire) to application-level messages. The seven layers of the OSI model are shown below:

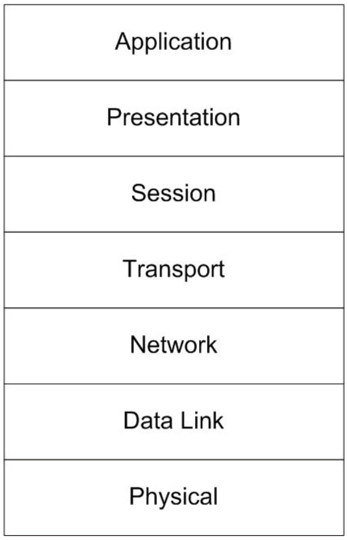

FIGURE 1 - OSI MODEL

The Physical Layer defines the transmission of raw data over the defined communication channel. The Physical Layer defines data representation, including signaling levels and other link properties (e.g., whether single-ended or differential signal drive is used, voltage levels and assigned data sense, presentation choices such as whether data is presented serially or in some parallel arrangement, signal timing, whether transmission is unidirectional, half-duplex or full duplex, and other related link properties). The Physical Layer defines the channel-level protocol for establishing and tearing down physical connection, and the physical interface (such as related connector pinouts) for the channel.

The Data Link Layer relies on the Physical Layer for access to the defined communication channel. The Data Link Layer provides a logical communications channel that appears free of undetected transmission errors. For reliable communications channels, the Data Link Layer is typically responsible for detection and correction (by retransmission request) of corrupted, missing, or duplicated message packets. For unreliable best-effort data channels, the responsibility of the Data Link Layer is typically limited to the detection and correction of corrupted message packets.

The Network Layer relies on the Data Link Layer for logical transmission of non-corrupted messages. The Network Layer is responsible for the routing of messages from source to destination; the Network Layer is typically responsible for network bandwidth management (detection of congestion and adjustment of various protocol timing parameters) as well.

The Transport Layer uses the services of the Network Layer to provide end-to-end delivery of messages. For reliable protocols, end-to-end delivery requires assurance of not only error-free delivery, but of full delivery (no dropped messages or packets) and in-order delivery. Unreliable transports (such as the User Datagram Protocol (UDP) of the Internet Protocol Suite) assure delivery of error-free packets without guarantees of full or in-order delivery.

The Session Layer of the OSI model provides higher-level logical data communication services, such as synchronization for recovery of aborted transfers of large streams of data.

The Presentation Layer of the OSI model builds on the Session Layer to provide additional high-level data communication services, such as accepting and presenting structured data rather than unstructured sequences of bytes. Conversion of data representations (such as the conversion of character encodings) is also in the domain of the Presentation Layer. Data encoding issues such as byte ordering (endianism) are dealt with by conversion services of the Presentation Layer. Data compression and encryption are also typically regarded as Presentation Layer services.

Application software using the services of the preceding six layers is modeled within the OSI model's Application Layer.

### 3.1.2 The Internet Protocol Suite Model

The Internet Protocol Suite (also known as the TCP/IP Protocol Suite, often referred to as the TCP/IP suite) is a family of protocols predicated on a four-layer protocol model, shown below:

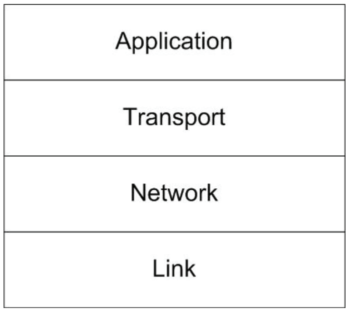

FIGURE 2 - INTERNET PROTOCOL SUITE MODEL

The Internet Protocol Suite model (or "TCP/IP Model") lacks Presentation and Session Layers, and collapses the Physical Layer and Data Link Layer of the OSI model into a single Link Layer.

The Link Layer of the TCP/IP Model absorbs the scope of the Data Link Layer of the OSI model, and the bulk of the OSI model's Physical Layer. The Link Layer of the TCP/IP Model defines a best-effort, unreliable, error-free data communications channel. Thus, while the Link Layer provides detection (and generally recovery via retry) of corrupted packets and guarantees that it will not pass corrupted packets up the protocol stack, it does not provide assurance of delivery nor does it provide assurance of in-order delivery. These assurances are left to layers above the link layer.

The Network Layer of the TCP/IP Model corresponds to the Network Layer of the OSI Model. Implementations of the TCP/IP Network Layer provide end-to-end routing of messages, but do not provide assurance of in-order delivery or guarantees of delivery.

The TCP/IP Model's Transport Layer corresponds closely to the Transport Layer of the OSI Model. Two end-to-end protocols are generally implemented in the TCP/IP Transport Layer. The first protocol, the User Datagram Protocol (UDP) provides connectionless, unreliable communications that allows applications to provide their own implementations of message-sequence assurance and communications bandwidth management. The second protocol, the Transmission Control Protocol (TCP), provides connection-oriented, reliable communications between applications. TCP provides message-sequence assurance and bandwidth management (albeit with unknowable additional delivery latencies).

In the TCP/IP Model, the Application Layer generally encompasses the higher-level communications services modeled in the Session and Presentation Layers of the OSI model.

### 3.1.3 The JAUS Model

The communications protocol stack model assumed by the JAUS specifications is a two-layer model. The two layers and their correspondence to the four layer TCP/IP model and the seven layer OSI model are shown below:

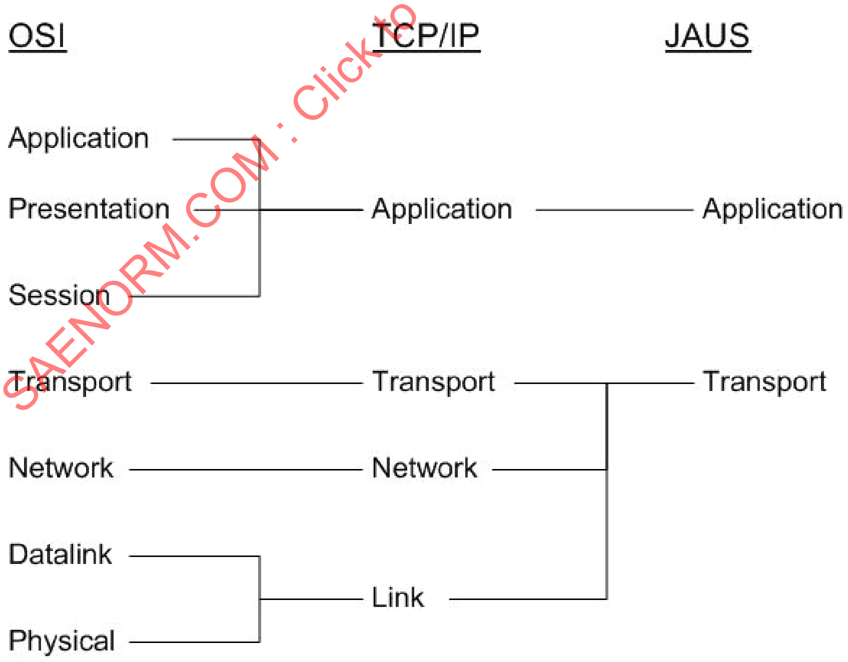

FIGURE 3 - MAPPING NETWORK LAYER MODELS TO JAUS

In this model, the Transport Layer subsumes the functions of the Link, Network and Transport Layers of the TCP/IP model. The Application Layer of the JAUS model maps directly to the Application Layer of the TCP/IP Model.

The Transport Layer is responsible for interfacing with operating system drivers and thereby with the communication media. The Transport Layer provides an interface to the Application Layer above it. From the perspective of the JAUS Application Layer, the specific responsibilities of the JAUS Transport Layer include:

- Sending and receiving actual messages via the underlying transport medium.
- Prioritizing packet transmission by assigned message priority. (The role of message priority is discussed in 4.2.1.1.)
- Packet reception and received packet handling by message priority. (The role of message priority is discussed below, in 4.2.1.1)
- Corruption checks on packet data, with transport-specific mechanisms for NAK and retry.
- Data reduction and data compression for improving transport performance by reducing required bandwidth in reduced-bandwidth environments[1]
- Data encryption and decryption for applications requiring secure transmission of information.

Some of these responsibilities will be covered by the JAUS Transport Standards themselves, while others will be covered by the native transport on which JAUS Transport rides. In each case, the characteristics of the individual native transport must be examined to determine if such support is available.

From the perspective of JAUS Transport Layer, the primary communications responsibilities of the JAUS Application Layer include:

- Packing of application headers and application data into individual JAUS application-layer messages for transport.
- Version information for application data, which is provided to allow unambiguous parsing of application message content.
- Assignment of message priority.
- Division of large, application-layer data sets into individual messages. (The transport of large messages is discussed in 4.3.2.)

These application-layer considerations are discussed individually and in greater depth in Section 4.

[1] Please refer to 5.4.

## 3.2 Compliance Levels

JAUS Compliance is defined by the JAUS Compliance Specification [JAUS-CMP]. Compliance is defined at three architectural levels, and is assessed in accordance with five rules of compliance. The five general compliance rules are satisfied by the first of the two Transport rules, as shown in the figure below:

TABLE 1 - JAUS COMPLIANCE RULES

| JAUS Compliance Rules | Transport Rules |
|---|---|
| All supported messages shall be explicitly listed for JAUS compliance of the subject JAUS element | (1) Transport shall not modify the content or format of any delivered transport payload. |
| All supported messages shall explicitly state the version of the [JAUS-RA]. | (1) Transport shall not modify the content or format of any delivered transport payload. |
| All supported messages shall follow the JAUS message conventions, definitions, formats and messaging rules as defined by the [JAUS-RA]. | (1) Transport shall not modify the content or format of any delivered transport payload. |
| All supported Inform messages shall respond to the corresponding Query messages. | (1) Transport shall not modify the content or format of any delivered transport payload. |
| All supported Event Notification messages shall respond to the corresponding Event Setup messages. | (1) Transport shall not modify the content or format of any delivered transport payload. |

A second Transport rule is required to accommodate the possibility that subsequent revisions of a specific Transport may change the layout of the Transport packet. Thus the two Transport rules are:

1. Transport shall not modify the content or format of any delivered transport payload.

2. All Transport packets shall explicitly state the version of the Transport.

The three levels of compliance, and their applicability to Transport, are discussed in the three following subparagraphs[2].

### 3.2.1 Inter-Subsystem (Level I)

Level I compliance addresses the compliance requirements for interoperability between subsystems. The purpose of Level I compliance is to support the interoperation of subsystems. Any defined Transport must support Level I compliance as it applies to communications between entities residing on distinct subsystems requiring the services of that Transport for communication.

### 3.2.2 Inter-Nodal (Level II)

Level II compliance addresses the compliance requirements for interoperability between nodes. The purpose of Level II compliance is to support the interoperation of nodes. Any defined Transport must support Level II compliance as it applies to communications between entities residing on distinct nodes requiring the services of that Transport for communication.

[2] For definitions of the terms subsystem, node, and component, please refer to [JAUS-RA] Part I, Section 3 System Topology.

### 3.2.3 Inter-Component (Level III)

Level III compliance addresses the compliance requirements for interoperability between components. The purpose of Level III compliance is to support software reuse. The technical mechanism for Level III compliance has not been defined within the JAUS standards at this time; therefore, Level III compliance is not testable. As a result, Transports are not required to support Level III compliance at this time.

## 4. JAUS APPLICATION FEATURES OF CONCERN TO JAUS TRANSPORT

This section examines features of the JAUS Application Layer Header that may be of interest or concern to JAUS Transport. Currently, there are two general categories of features of the JAUS Application Layer Header that are of such concern:

- The possible use of fields contained in the JAUS Application Layer Header by JAUS Transport.

- The requirements imposed on Transport by the semantics of the JAUS Application Layer Header.

The JAUS Application Layer Header fields of use to Transport, and their usage, are discussed in the subsections that follow.

Section 4.1 describes the different types of usage that may be made of Application Layer Headers by the Transport Layer.

Section 4.2 examines the fields of the JAUS Application Layer Header, demonstrates the characteristics of some of these fields that make them relevant to JAUS Transport, and assigns each such usage into one of the categories described in 4.1.

Section 4.3 builds upon these previous discussions to determine any transport functionality or characteristics that are implied by Application Layer header semantics.

Section 4.4 summarizes the results of these discussions.

### 4.1 Categories of Cross-Layer Header Usage

When examining how Application Layer header fields may be used by the Transport Layer, it is important to note that there are actually two different types of potential usage here:

- Use of a field from the JAUS Application Layer Header within the Transport Layer itself, in order to affect how the message is transported.

- Use of a field from the JAUS Application Layer Header as the individual JAUS message is moving across the boundary between the Application and Transport Layers of the protocol stack.

The first bullet point, above, is termed Header Transgression. The second bullet point is actually the means by which the Application Layer specifies the desired transport functionality to the Transport Layer: This will be referred to as Desired Transport Functionality.. These terms will be used in the following sections to describe the type of Application Layer Header field usage.

For example, consider a JAUS message that is being transferred from an IP-based network to a serial one. The message, once received via the IP-based network, would have to be passed to the serial transport for transmission. The semantic content of the message's application data would not be of concern to the transport link – the transformation is purely of the transport-to-transport variety. Any Application Layer fields that must be looked at in such a context in order to accomplish this kind of transformation would qualify as being of the Header Transgression variety.

On the other hand, consider the example of a robot whose JAUS Application Layer is preparing to send a status report to its controller. The message would most likely be passed to a process or function whose job it is to actually send the message – by determining where and how to send it, based on the fields from the application header (mapping JAUS Identifiers to Transport Addresses, determining when to send the message based on its Priority, etc.). This is an example of the usage of Application Layer fields that would qualify as being merely of the "Desired Transport Functionality" variety.

Where appropriate, the JAUS Application Layer Header fields described in the sections that follow will specify the type of Application Layer header field usage employed: None, Header Transgression, Desired Transport Functionality.

## 4.2 JAUS Application Layer Header Usage by Transport

The specific fields found within the JAUS Application Layer Header may (and probably will) change over time, with new fields added and/or old fields removed, as the JAUS standards progress. Maintaining a loose coupling between the layers of the protocol stack is therefore desirable, as the loose coupling allows both Transport and Application standards to develop and move forward without the retarding influence that increased interdependence between these layers would bring. This is a fundamental principle behind the design of protocol stacks for standards suites.

However, when a developer implements a JAUS Transport Standard for a particular communications medium, some Application Layer header fields may be examined in the course of conveying a JAUS message from one JAUS entity to another, but only in order to improve transport efficiency. Since the boundary transgression this requires between the application and transport layers is in general undesirable, these cases should be restricted to those instances where significant gains in performance may be had. And in no case shall this boundary transgression be required – it is used to achieve improved performance only, not correctness. This restriction maintains the loose coupling of standards on the protocol stack.

The current JAUS Application Layer Header[3] defines an abstract message header without reference to any particular transport medium. (Indeed, one could consider this header as specifying the message format for working with JAUS messages within a single processor or computer node – say, for communication between processes via FIFOs or shared memory.) The subsections that follow summarize the possible usage (if any) of each of the current JAUS Application Layer Header fields by Transport. In future versions of JAUS, new Application Layer fields should likewise be examined in future versions of this document for their relevance (or lack thereof) to JAUS Transport.

### 4.2.1 Message Properties

The Message Properties field of the JAUS Header provides information important to the efficient handling and delivery of the JAUS message. Information provided in the Message Properties fields includes the priority of the message, the version of JAUS application data associated with the message header and message, and other information related to specific JAUS capabilities.

### 4.2.1.1 Priority

The Priority field specifies the priority for processing a message. Messages of higher priority should be processed before those of lower priority.

Particularly in reduced-bandwidth environments, the message priority is key information to the Transport. Given a reduced-bandwidth environment, it is likely that multiple messages are queued for transmission at any given time. It is not acceptable for a Safety-Critical message to wait for all previously queued messages to be transmitted before it is transmitted in turn: In such a scenario, operational safety could be compromised by the resultant delay in propagating a critical message.

[3] The JAUS message header is defined in [JAUS-RA] Part 2, Section 3.3.1.

As a result, implementations of transport functionality that are capable of encountering this kind of queued message dispatch behavior are strongly advised to transgress the layers of the protocol stack to access the Priority field from the Application Layer header, in order to enable transmission-by-priority in reduced-bandwidth Transport implementations. It is strongly suggested that implementations of transport functionality capable of encountering this kind of queued message dispatch behavior should handle queued messages of like priority in a first-in, first-out (FIFO) manner.

Additionally, if messages will be used whose transmission time would unacceptably delay the transmission of a higher priority message, it may be necessary for the Transport implementation to define mechanisms to abort transmission of such long messages in order to enable timely transmission of a subsequently queued, higher priority message.

Refer to 5.3 for further insight into the use and support of the JAUS priority field for operation in limited-bandwidth environments.

The type of Application Layer header field usage for the Priority field is Desired Transport Functionality bordering on Header Transgression, in cases such as the example described in 4.1.

#### 4.2.1.2 ACKNAK

The ACKNAK field provides an application-level end-to-end acknowledgment mechanism. Transports require an ACKNAK mechanism for implementation of automatic recovery by retry request (ARQ), used to recover from partial corruption of a received packet. Since the basic requirements and operation of these two acknowledgment mechanisms are quite different, Transports implementing an ARQ mechanism must implement a Transport level ACKNAK separate and distinct from the JAUS header ACKNAK.

Any Transport implementing ACKNAK and ARQ mechanisms must document those mechanisms in the Transport Specification [JAUS-TS].

The type of Application Layer header field usage for the ACKNAK field is None.

#### 4.2.1.3 Service Connection

The Service Connection field is strictly related to an application-level protocol.

The type of Application Layer header field usage for the Service Connection field is None.

#### 4.2.1.4 User-Defined

The User-Defined field of the JAUS Application Layer header (also known as the “Experimental” field) is strictly related to application-level messaging.

The type of Application Layer header field usage for the User-Defined field is None.

#### 4.2.1.5 Version

If JAUS Transport uses any of the other application-level header fields, it must also examine the Version field of the JAUS Application Layer Header. This is required in order to verify the locations and/or existence of the other header fields, as different header versions will have different header fields.

The type of Application Layer header field usage for the Version field is therefore derived from the combination of the usages of all other application-level header fields.

#### 4.2.2 Message Code

The Message Code field (also known as the "Message Command" field) of the JAUS Application Layer header is strictly related to application-level messaging.

The type of Application Layer header field usage for the Message Code field is None.

#### 4.2.3 Destination Identifier

The JAUS Destination Identifier field is a composite identifier described in [JAUS-RA][4]. This identifier is used as a transport-independent designator at the application layer (see 4.3.3). It is used at the point where the Application Layer meets the Transport Layer, in order to map that identifier into a transport address to which the message should be sent.

The type of Application Layer header field usage for the Destination Identifier field is Desired Transport Functionality.

#### 4.2.4 Source Identifier

The JAUS Source identifier field is a composite identifier as described in [JAUS-RA][5]. This identifier is used as a transport-independent designator at the application layer (see 4.3.3). It is used at the point where the Application Layer meets the Transport Layer, in order to map that identifier into a transport address from which the message should be sent.

The type of Application Layer header field usage for the Source Identifier field is Desired Transport Functionality.

#### 4.2.5 Data Control

The Data Control field specifies both the size of the JAUS message body, and the handling of fragmentation and reassembly for large data set messages within the application layer.

##### 4.2.5.1 Data Size

The Data Size subfield is used by Transport to determine the size of the data payload to be transported.

The type of Application Layer header field usage for the Data Size subfield is Desired Transport Functionality.

##### 4.2.5.2 Data Flags

The Data Flags subfield supports operations involving large data sets – their fragmentation and reassembly at the Application Layer.

The type of Application Layer header field usage for the Data Flags subfield is None.

#### 4.2.6 Message Sequence Number

The Message Sequence Number field supports application-level semantics.

The type of Application Layer header field usage for the Message Sequence Number field is None.

[4] [JAUS-RA] Part 2, Section 3.3.1.3.
[5] [JAUS-RA] Part 2, Section 3.3.1.3.

### 4.2.7 Header Usage Summary

Consider again the example given in 4.1 of a transport gateway between IP and serial networks. The foregoing discussion of the usage of Application Layer header fields in the Transport Layer demonstrates that in such a situation the only relevant question is whether or not messages may end up being queued at that location in the network. If messages may be queued, then the Priority field (and consequently the Version field) will be used via Header Transgression.

This analysis intuitively makes sense, as the Transport Layer has no insight into which Application Layer messages should be sent ahead of others, and therefore must examine the JAUS Application Layer Header data in order to determine that.

## 4.3 Semantics/Functionality Required or Implied by the JAUS Header

The JAUS Application Layer Header fields describe both the content of the message and the characteristics that are relevant to where and how the message should be conveyed. Some of these characteristics require the support of transport features.

### 4.3.1 Broadcast Semantics

The “broadcast” values described in [JAUS-RA] as part of the composite JAUS IDs require transport support for broadcast semantics. A transport standard for a particular communications medium must be able to implement the conveyance of a message “to all subsystems within a system”, or “to all nodes within a subsystem”, or “to all nodes within the whole system”. (Intranode broadcasts are not subject to internode transport, and are therefore outside the scope of Transport Standards)

The implementation of broadcast semantics does not require that broadcast be done via a single message only. For instance, if a medium supports several channels, individual messages may be required to be sent across these individual channels in order to reach all the desired entities.

The crucial aspect of the implementation of broadcast semantics in JAUS is that a message reaches as many of the desired recipients as the current network connectivity allows with as few messages as possible.

### 4.3.2 Transporting Large Messages

When transporting large messages across a communications network, consideration must be given to those messages which exceed the Maximum Transfer Unit (MTU) of a layer of the protocol stack. When this happens, measures must be taken to break up the larger message into chunks that are MTU-sized or smaller, convey these smaller pieces across the network, and then reassemble the message on the other side.

In such situations, there are two, primary questions that should be addressed:

1. Can the message being conveyed tolerate loss of some of its pieces and still be useful?

2. What is the maximum possible message size that a receiver should prepare to receive?

Each of these questions is addressed below.

#### 4.3.2.1 Tolerating Packet Loss in Large Messages

Typically, large messages are not tolerant of packet loss. Such messages usually require reliable transmission, unreliable but complete transmission, or unreliable incomplete transmission where acceptable losses are subject to defined constraints:

- Reliable transmission, albeit with the usually unbounded latency that reliable transmission requires in typical unmanned systems network environments.
  - Example: Sending a complete map of the current environment – no matter how large it is, the receiver needs the entire map.
- Unreliable but complete transmission, in which the receiver needs a complete message, but if the current transmission is incomplete, no resend is required.
  - Example: Sending motion JPEG video stream (a stream of individual JPEG images) – if one image is incomplete, the receiver should just dump it, as the next image should typically be along shortly anyway.
- Transmission of data which can tolerate some loss, but only when that loss is on clearly delineated boundaries.
  - Example: Wavelet video schemes can tolerate the loss of some video packets, so long as the structure of the received packets conforms to the expected boundaries within the data stream.

The first two cases demonstrate the need for support in JAUS transport standards for transport of complete large JAUS messages both with and without retry semantics.

The third case suggests support for Application Layer message division and reassembly in the JAUS Application-Layer standards, as in these situations, it is only the Application Layer which has the information necessary to divide the larger message on the appropriate boundaries.

#### 4.3.2.2 Maximum Packet Size

When implementing a protocol stack, the message receiver needs to know the maximum size of a single packet. This allows the static (non-runtime) allocation of buffers, which is highly desirable in terms of performance, clarity and reduced complexity.

For historical reasons, JAUS has typically defined a maximum overall message size (at the Application Layer) of 4096 bytes. This is, conveniently, the maximum size for an atomic write to a FIFO in typical, current Unix/Linux implementations.

In order to accommodate this backwards compatibility easily, each JAUS Transport Specification should define its maximum packet size to conform to an Application Layer message of no larger than 4096 bytes, as well as define a maximum packet size for that transport medium.[6]

[6] A transport for a given communications medium may also define its maximum packet size in terms of characteristics of that medium. For example, a serial transport may restrict its maximum packet size to a smaller value for lower speed links, in order to accommodate various timing parameters of JAUS protocols.

### 4.3.3 Using JAUS Identifiers as Addresses

The use of unique identifiers (the “JAUS IDs”) for each JAUS entity has long been a topic of discussion within the JAUS standards body, particularly with regard to whether or not they are necessary when there is almost always a unique one-to-one mapping between subsystem/node numbers and transport layer addressing. Indeed, the term “JAUS Address” has been used to refer (informally) to the JAUS ID fields in the JAUS Application Layer header.[7]

There are advantages to the JAUS identifier scheme:

- A good principle for the design of messaging standards is to keep addressing and identity as two separate concepts that should not be confounded. Preserving this distinction keeps the semantics of the standard/protocols clean.

- The need for two parties in a protocol to refer to some distinct third party is a common occurrence. This need results in these “references to third parties” being embedded within the body of protocol messages.

    - Example: Consider a Mission Planner that wants to plan an activity with two robots. In each robot’s plan, the Mission Planner may want to refer not only to itself (the source of the mission-planning messages) and the current robot (the destination of the mission-planning messages), but also to the other robot (the 3rd-party reference).

    - Without a unique, non-transport-dependent way to refer to a third party, transport infrastructure that includes gateways, heterogeneous media and/or network address translation (NAT) would each have to build and deploy JAUS-specific software to look inside packets and make appropriate address translations. This is very inefficient and undesirable.

There are also disadvantages to this scheme:

- Since the one-to-one mapping is by far the most common case, the use of the identifiers tends to amount to a redundant layer of addressing, increasing the size of the message header for typically little gain.

- The maintenance required by a JAUS system in order to ensure that all JAUS identifiers are unique is a non-trivial logistics problem, particularly as the size of the system scales up.

As previously noted, the standards group has debated these issues extensively. The conclusion of the committee is that the “third-party reference problem” is decisive. This is functionality that the JAUS standard must support; hence, the JAUS Transport standards must support it efficiently.

The impact of this decision is eased by additions/modifications to the design of the JAUS Transport standards that mitigate these disadvantages. In particular:

- The addition of header compression techniques to the JAUS Transport Standards allows a JAUS system to reclaim most of the bandwidth lost to “redundant” JAUS identifiers. In practice, in systems where the JAUS identifiers are truly redundant with the transport layer addressing, these header compression techniques should allow the system to compress away the redundant portions of the JAUS messages.

- The addition of “best practices” guidelines for each individual transport standard allows for an easy mapping between transport layer addressing and JAUS identifiers. If followed consistently within a JAUS system, these best practices will typically allow the automatic derivation of a JAUS identifier from a transport layer address, simplifying the logistics problem.

[7] For a full description and discussion of JAUS identifiers, refer to [JAUS-RA], Part I, Section 3 System Topology and Part II, Section 3.4, Message Routing.

## 4.4 Summary: JAUS Application Features

The examination of features of the JAUS Application Layer accomplished in this section delineates those characteristics of JAUS which should be considered when designing a transport standard for JAUS messaging across a communications medium.

# 5. COMMUNICATIONS MEDIA FEATURES OF CONCERN TO JAUS TRANSPORT

The established standards and best practices of network communications provide important background information necessary to understand the development of JAUS Transport standards. This section provides an introductory discussion of these concerns and their applicability to the JAUS Transport.

## 5.1 IP-Based Transports

IP-based transports typically provide significant infrastructure which any JAUS transport implementation should take advantage of. Some characteristics of such infrastructure are universal; the implications of these infrastructure characteristics are discussed in this section.

### 5.1.1 IP Addressing

IP-based transports by definition support IP addressing – typically version 4 or version 6. The assignment of these addresses is outside the scope of the JAUS transport specification. Each unmanned system deployed must implement IP addressing infrastructure, accompanying routers, etc. as required by the environment in which it operates.

JAUS IP-based transports should use static IP addressing, or self-derived IP addressing (as is used in “Zero Configuration” networking [ZCONF05]). The use of dynamic address allocation of the type provided by a DHCP server is undesirable, as unmanned platforms may initialize themselves in locations that are out of communications range of such network infrastructure. Similarly, reliance on DNS name resolution and host file mapping is undesirable. However, for dedicated, in-place deployments, these issues may not be of concern.

IP addresses of JAUS entities should be accessible for modification. The addresses should be accessible for re-assignment without rebuilding, reinstalling, or otherwise modifying the software operating the JAUS host node(s) involved.

### 5.1.2 Support for Broadcast Semantics

In IP infrastructure, this is typically done via an a priori configured multicast address, allowing multicast-enabled routers to easily scope the transmission of such messages.

Multicast addressing configuration should be accessible for modification without rebuilding, reinstalling, or otherwise modifying the software operating the JAUS host node(s) involved.

## 5.2 Data-Transparent Low-Overhead Protocols

Reduced-bandwidth environments require link-level protocols that are efficient and are robust in the presence of data loss and data corruption. These protocols most often must also do their own packetization, as the overhead required for generalized packetization is often deemed unacceptable. Therefore, this section examines the characteristics of (typically serial) transport media whose across-the-wire format can be interpreted as a sequence of bits.

In general, successful link-level protocols for these environments share several characteristics. In the following subsections, several necessary characteristics for robust operation in these environments are discussed. One of the considerations in the design of link-level data link control protocols is the design of packet framing for variable length data, such as the JAUS payload. A key element in packet framing design is the concept of data transparency. These considerations and their ramifications for transport design are discussed in the subparagraphs that follow.

### 5.2.1 Data Link Control and Data Transparency

Typically, encapsulated protocols for transmission in lossy environments make use of unique characters or character digraphs (sequences of two characters) to delimit sections of the packet (to uniquely identify the transport header, the payload, and the transport trailer, for example). Thus, several such delimiting characters or character sequences must be defined.

When the data payload consists of a sequence of bytes, each of which is a member of a subset of the available values (for example, when the data payload consists exclusively of byte values that exclude the set of ASCII link control characters), single byte characters may be employed. When the data payload consists of a sequence of bytes which may assume any values (as is the case for the JAUS payload), other conventions must be sought.

Various conventions have been used, but one of the most influential has been that of ANSI standard X3.28-1976[8], which provides for unique link-control digraphs (prefixing all link-control characters with an ASCII DLE byte) and for data transparency (by requiring that any DLE-valued data in the packet be prefixed with an ASCII DLE byte; this practice is referred to as "DLE insertion").

### 5.2.2 Frame Check and Header Check

Each packet transmitted must include some form of frame check or blockcheck if packet corruption over the channel is to be detected. Packet corruption detection enables the Transport to perform packet replacement by retry to overcome the corruption associated with a noisy communication channel. But a packet blockcheck is in itself inadequate if the header data is required to make a response. (For example, if the communications channel is addressable, the sender's address must be valid in order to respond; or, if multiple messages may be pending response, the header must have sequence numbering, and the sequence number must be valid in order to respond.) Since response cannot be made to a corrupt packet unless the packet header is valid, some form of header-specific blockcheck must be present in addition to the packet blockcheck: If the packet blockcheck shows that the received packet is invalid, the receiver can request retry only if the header blockcheck indicates that the header data is not corrupted.

Thus, any link-level transport implementing ARQ capabilities must employ an independent header blockcheck in addition to the packet blockcheck.

### 5.2.3 Effectiveness of Blockchecks

A number of different blockcheck algorithms are employed in commonly used transports. Three families of blockchecks dominate: simple two's complement checksums, various one's complement checksums, and cyclic redundancy checks (CRC).

Simple two's complement checksums are easily computed. They are, however, limited in effectiveness[9]. They may be reasonably used to span small blocks of data; they have been used to provide a blockcheck of transport header data within a packet protected by a stronger blockcheck.

Some variants of one's-complement checksums offer protection against single-bit and some multibit errors nearly comparable to those of similarly sized CRCs. Examples of these blockchecks include Fletcher's checksum[10] and variants thereof. These blockchecks are much more readily computed than a CRC, and may be applied advantageously to moderately small packets.

[8] [X3.28]
[9] [FLET82], Table I.
[10] [FLET82] and [KODIS92].