MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21c/2019/11.12.10.23-PUD

SPECIFICATION OF A LAYER-2 PROTOCOL AND SERVICES FOR EMBEDDED NETWORKS ONBOARD AEROSPACE VEHICLES

Sérgio Duarte Penna

Exame de Qualificação para Doutorado do Curso de Pós-Graduação em Engenharia e Tecnologias Espaciais, orientada pelo Dr. Marcelo Lopes de Oliveira e Souza.

URL do documento original:
<http://urlib.net/8JMKD3MGP3W34R/3UCU32E>

INPE
São José dos Campos
2019

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE
Gabinete do Diretor (GBDIR)
Serviço de Informação e Documentação (SESID)
CEP 12.227-010
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/7348
E-mail: pubtc@inpe.br

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21c/2019/11.12.10.23-PUD

SPECIFICATION OF A LAYER-2 PROTOCOL AND SERVICES FOR EMBEDDED NETWORKS ONBOARD AEROSPACE VEHICLES

Sérgio Duarte Penna

Exame de Qualificação para Doutorado do Curso de Pós-Graduação em Engenharia e Tecnologias Espaciais, orientada pelo Dr. Marcelo Lopes de Oliveira e Souza.

URL do documento original:
<http://urlib.net/8JMKD3MGP3W34R/3UCU32E>

INPE
São José dos Campos
2019

“Quando estás certo, ninguém se lembra; quando estás errado, ninguém esquece”.

Provérbio irlandês

Aos colegas da Memória Técnico-Científica e aos alunos da Pós-Graduação do INPE.

ACKNOWLEDGEMENTS

Agradeço ao meu orientador Professor Doutor Marcelo Lopes de Oliveira e Souza pela sua valiosa contribuição na elaboração deste texto.

RESUMO

Protocolos de comunicação são componentes essenciais em sistemas complexos e altamente integrados embarcados em veículos aeroespaciais. A implementação deste tipo de componente de software pode demandar um alto custo de processamento, caso ele próprio seja de alta complexidade, portanto buscar protocolos mais simples que executam a tarefa com a eficiência desejada deve fazer parte de um bom de projeto de desenvolvimento de sistemas. Este trabalho apresenta um novo protocolo de comunicação para a Camada de Enlace e todos os serviços associados e necessários à sua implementação em meio-físico Ethernet.

Palavras-chave: Protocolo de comunicação, Camada de Enlace, Ethernet.

ABSTRACT

Communication protocols are essential components in complex and highly integrated systems onboard aerospace vehicles. The implementation of such type of software component may demand a high processing cost, should itself being of high complexity, therefore choosing simpler protocols that perform the task with the desired effiency must be part of a ggor system development process. This work presents a new Data Link layer communication protocol e all its associated and require services to its implementation over Ethernet physical medium.

Keywords: Communication protocol, Data Link Layer, Ethernet.

LIST OF FIGURES

Page

SUMMARY

Page

1 INTRODUCTION

Connecting people seems to be the most important consequence of a technology asset which began its development back in the 19th century with the telegraph. The ability of communicating facts over a physical medium beyond line-of-sight changed the face of the world.

“Communication”, according to a web dictionary (MERRIAN-WEBSTER, 2019), is “a process by which information is exchanged between individuals through a common system of symbols, signs, or behavior”. However, to what purpose one or more individuals would use “communication”? According to other web dictionary (LEXICO, 2019), here is a very good reason: “the successful conveying or sharing of ideas and feelings”.

Therefore, “communication” needs not only to be effective allowing two parties to connect, but it needs also to convey the correct fact or data.

Surprisingly at first, for electronic control systems embedded in modern vehicles, be it a car, bus, train, aircraft or spacecraft, “communication” is not only essential, but vital to their safe operation. “Communication” is what binds devices together forming a complex network of specialized functions.

In the early stages of the development of electronic control systems, processing was done by a single complex device, such as the trajectory control system of the V2 rocket developed by the Germans during the World War II. Every step of processing and resulting action on V2’s rocket engine and tail fins was performed by a unit called LEV-3 (WIKIPEDIA, 2019) and an analog computer designed by Helmut Hoelzer, an electrical engineer (EDISON TECH CENTER, 2019). Very little “communication” was necessary, all of it in the analog world..

“Communication” using analog signals, current or voltage, prevailed until the advent of the microprocessor. An early evidence of data being communicated using digital signals was in the AGC - Apollo Guidance Computer (WIKIPEDIA, 2019).

The so-called “Digital Data Buses” (DDB) started being standardized in the beginning of the 70’s and firstly used in military aircraft, namely the F-16 Falcon, then lately in spacecraft as well.

DDBs started being used in commercial aircraft which flew for the first time in the beginning of the 80's, Boeing 767 and Airbus A320 to name two pioneer users.

DDBs proved very important as electronic control systems became more and more complex, as more complex functions could be accommodated because of more and more powerful microprocessors. They evolved, as the topology of these systems changed for every new vehicle depending on how control functions were physically allocated in electronic units.

More recently, changing from more concentrated to more distributed allocation of functions in electronic control systems has driven very important changes in DDBs technologies.

There was also an interesting migration of DDBs from one industry field to another: from aircraft to automotive, from automotive to aircraft, from aircraft to space, from Information Technology to Manufacturing.

DDB evolution does not show signs of interruption, as new data processing and data communication scenarios are created for new vehicles and new industry fields.

2 COMPUTER NETWORKING FUNDAMENTALS

The next two sections introduce two fundamental concepts which are relevant to this work.

2.1 Computer Network Architectures

The correct understanding of this work requires some knowledge of architectures used in building computers networks. In this section, the term “node” will be used freely to represent a single computer in a computer network.

Connecting computers became a necessity in the mid 70’s for a few reasons, but one very important: computers were very expensive, therefore sharing resources became strategic. If you needed to expand, it made more sense to acquire another computer tailored to your needs than to replace the one you already had by a bigger model. Luckily, computers those days enjoyed a quite long operational life: they remained operating for several years (quite commonly for 5 to 10 years).

Large computer networks appeared in the mid 80’s, when computers became smaller in size and less expensive.

In the early 70’s, it was already possible to connect two geographically separated computers using a private channel. The most common realization of this means of communication was over a telephone line. Binary digits were transformed into electric signals by a device called “modem”, which basically modulated an electric signal on transmission and demodulated it on reception (hence the name “modem” – agglutination of “modulator/demodulator”).

This “point-to-point” communication was enough for connecting two computers. If a third or fourth computer were involved, data had to be received and retransmitted to the next network node.

Even today, “point-to-point” communication is still used, in particular in the aerospace domain. In the late 70’s, a large computer network using “point-to-point” communication was created by the Advanced Research Projects Agency (ARPA) of the American Department of Defense.

In 1973, XEROX Corporation came up with a technical solution for connecting multiple computers influenced by a computer networking experiment conducted

at the University of Hawaii: the “ALOHAnet”. Using a coaxial cable, the “Ethernet” cable, multiple computers at XEROX PARC laboratory could share a common physical medium, allowing any node to directly communicate with any other node in the network, a significant improvement over “point-to-point” communication.

This “shared medium” arrangement had already posed a challenge to the creators of ALOHAnet: the recovery from the event of two computers “colliding” as they tried to transmit data at the same time. The researchers at the University of Hawaii came up with a simple protocol which inspired XEROX in the implementation of similar technique for properly handling these events: after failing to transmit by detecting a “collision”, the computers would have to wait a random chosen time interval before retrying.

In the mid 80’s, IBM adopted a design developed by researchers at the Cambridge University which arranged computers in a “ring”. This “ring” was formed by connecting computers “point-to-point” – the first to the second, the second to the third and so on – and closing the “ring” by connecting the last node to the first node.

The transmission of data in this “ring” required the ownership of a special piece of data called “token”: only the computer in the possession of the “token” was allowed to transmit. After transmitting, the computer would then pass the “token” to its neighboring computer in the “ring”.

“Ethernet” evolved and managed to move away from the coaxial cable because it became a technical issue as computers in a network grew in number: cable lengths were limited and one would need eventually to replace the cable when the number of computers connected exceeded the limit allowed by a particular cable length.

Devices called “Ethernet hubs” were developed, partly introducing a return to the “point-to-point” scheme of earlier computer networking days. Instead of connecting to a coaxial cable, computers using “Ethernet” would connect “point-to-point” to this “Ethernet hub”. This “hub” worked as a collapsed form of the traditional “Ethernet cable”. Installation and reinstallation of computers in an

“Ethernet” network was facilitated and “Ethernet hubs” became commercially available with 8, 16 or more connection ports.

“Ethernet hubs” eventually turned into the “Ethernet switches” we recognize today in almost every household as part of the “access point” hardware delivered by Internet service providers.

Occasionally, “hubs” were also called “star couplers” to denote the physical resemblance of computers connected to a “hub” and a planetary system, where planets are tied to a star by gravitational forces.

Today, the term “topology” (https://en.oxforddictionaries.com/definition/topology) is commonly used to describe how computers are arranged in networks. Computer network topologies were a consequence of technical decisions made by academic researchers and engineers trying to find solutions for real problems. If one can summarize elementary computer network topologies, they can be either: a) “Point-to-point”; b) “Shared bus” or simply “Bus”; c) “Ring”; d) “Star”. Naturally, more complex topologies can be obtained by combining one or more of these four.

Another important component of computer network architectures is the communication medium access control. It was mentioned before that “Ethernet”, a “bus” topology in its origin, allowed for any two computers to start transmitting at any time and that a “token” was used to grant the privilege of transmitting in a “ring” topology to the computer which owns it. In short, access control to the physical communication medium in computer networks can be done either: a) by using an arbitration protocol, which allows for one and only one transmission at any given point in time; b) by using no arbitration protocol, but providing a recovery mechanism in the event of a failed transmission.

Most computer networks operate over metallic of fiber-optic cables, but electromagnetic waves are also another viable medium. While the first can be constructed tolerant to electronic noise and harsh environments, open-air – or “wireless” – transmissions suffer greatly in the presence of natural phenomena, such as atmospheric discharges and heavy rain, and other radio transmissions from nearby sources.

The physical nature of the communication medium drives variations on the way access to the medium is controlled. For instance, wireless transmissions tend to avoid “collisions” instead of reacting to them.

More recently, data security has become a great concern in computer networks operating in commercial aircraft because of the fear of “hacking”, that is, a malicious attack which may result in loss of property and/or human lives. Networks which operate over cables, metallic or non-metallic, are more immune to attacks because it is necessary to have physical access to the network hardware. Networks which operate wireless can be protected against “hacking” by using encryption of data, but may still suffer in the presence of high-power electromagnetic transmissions causing what it is commonly called “denial-of-service”.

One could argue that the term ”architecture” should be used exclusively when speaking about “form”, that is, what can be observed by the naked human eye. The term “architecture” in computer networks could limit the discussion around “topologies” only, which indeed define the “form” of a computer network. The careful reader will note however that this section, besides enumerating topologies, addresses also the physical medium and the type of control used in accessing it. The reason is simple: computer networks were conceived, implemented and perfected by combining these three elements: a) topology; b) physical medium; c) access control to physical medium. While the first one provides a “high-level” perspective of the network, as it dictates its “form”, the latter two elements are its “lowest-levels”.

Researchers in Academia and Industry wrote the history of computer networks in the 70’s and in the 80’s. Today, we benefit from their hard pioneer intellectual work.

2.2 The ISO/OSI Layered Communication Model

The Open Systems Interconnection model is a product of a project conducted by the International Standards Organization (ISO) and was published as a standard (7498) in 1984 (ISO, 1994).

It describes a seven-layer abstraction communication model, where one layer has to be concerned only with the interface to the layer immediately above it,

which it serves, and the interface with the layer immediately below it, which it is served by.

These seven layers are:

Layer 1: Physical Layer (lowest)

The physical layer is responsible for the transmission and reception of encoded binary digits over a transmission medium. Examples of Layer 1 protocols are IEEE 802.3 and Ethernet physical layers, serial transmission protocols such as RS-232, Universal Serial Bus (USB), IBM's Bluetooth, among others.

Layer 2: Data Link Layer

The data link layer provides actual data transfer between two directly connected nodes. Examples of Layer 2 protocols are IEE 802.3 (combined with IEEE 802.2 LLC) and Ethernet data link layers, Asynchronous Transfer Mode (ATM) for audio and video streaming, among others.

Layer 3: Network Layer

The network layer provides the transferring of variable length data structures (usually called "packets") from one node to another. The most famous example of a Layer 3 protocol is the Internet Protocol (IP), but others can be accounted for, such as Apple's Appletalk, Novell's Internetwork Packet Exchange (IPX) and Digital Equipment Corporation's DECnet.

Layer 4: Transport Layer

The transport layer provides the transferring of arbitrary length data sequences adding extra services such as segmentation (dividing a sequence into smaller pieces for transmission), error detection and recovery. Examples of Layer 4 protocols are Transmission Control Protocol (TCP), User Datagram Protocol (UDP) and Novell's Sequenced Packet Exchange (SPX).

Layer 5: Session Layer

The session layer provides establishing, managing and terminating virtual "connections" between a local and a remote application. Real-time Transport

Protocol (RFC 3550) developed for audio and video streaming over IP is one of the few true Layer 5 protocols.

Layer 6: Presentation Layer

The presentation layer helps bridging different syntax and semantics between two Application Layer applications. One of the few Layer 6 protocol examples is the Multi-purpose Internet Mail Extensions (MIME), which allows for sending non-textual attachments over e-mail.

Layer 7: Application Layer (highest)

The application layer is the OSI layer closest to an end user software application. There is a multitude of Layer 7 protocols in use today: Hypertext Transfer Protocol (HTTP) used in the World-Wide-Web, File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), Simple Network Management Protocol (SNMP) and Telnet, all of them operating over IP.

The most frequent implementations of network protocols usually concentrate in the first four lower layers and the uppermost layer.

Figure 3.1 – ISO/OSI Layered Communication Model.

OSI model				
Layer			Protocol data unit (PDU)	Function[5]
Host layers	7	Application	Data	High-level APIs, including resource sharing, remote file access
	6	Presentation		Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
	5	Session		Managing communication sessions, i.e. continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4	Transport	Segment, Datagram	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing
Media layers	3	Network	Packet	Structuring and managing a multi-node network, including addressing, routing and traffic control
	2	Data link	Frame	Reliable transmission of data frames between two nodes connected by a physical layer
	1	Physical	Symbol	Transmission and reception of raw bit streams over a physical medium

Source: Adapted from Wikipedia (2020).

Physical layer protocols implementing serial point-to-point communication, such as RS-232, RS-422 and RS-485, are used for a multitude of upper layer protocols. USB is another example rich example of flexibility, used for connecting portable storage devices, microphones, loudspeakers, video cameras, keyboards, pointing devices to personal computers, cell phones and modern TV sets.

For most applications, it is sufficient connecting directly the Data Link Layer to the Application Layer, that is, once the application receiving the data is identified, it should get it with as little processing delay as possible.

2.3 IEEE 802.3 Standard for Ethernet

The Ethernet protocol for networking communication developed by XEROX in 1973 became a “de facto” standard by 1982 after Digital Equipment Corporation and Intel Corporation formed with XEROX the consortium called “DIX” (acronym for Digital-Intel-XEROX).

The IEEE 802.3 standard was published in 1985 (IEEE, 2012) and, as Ethernet, it covers the first two layers of the ISO/OSI Layered model:

Layer 1 “Physical Layer” – standardizes all sorts of physical medium, from copper cables in various forms to fiber-optic cables, from transmission speeds starting at 10 megabits per second to 200 gigabits per second.

Layer 2 “Data Link Layer” – standardizes two sub-layers, the “Media Access Control” (MAC), basically the “Carrier Sense Multiple Access with Collision Detection” (CSMA/CD) method devised by XEROX for Ethernet, and the “Logical Link Control” (LLC) subject of the IEEE 802.2 standard (IEEE, 1998).

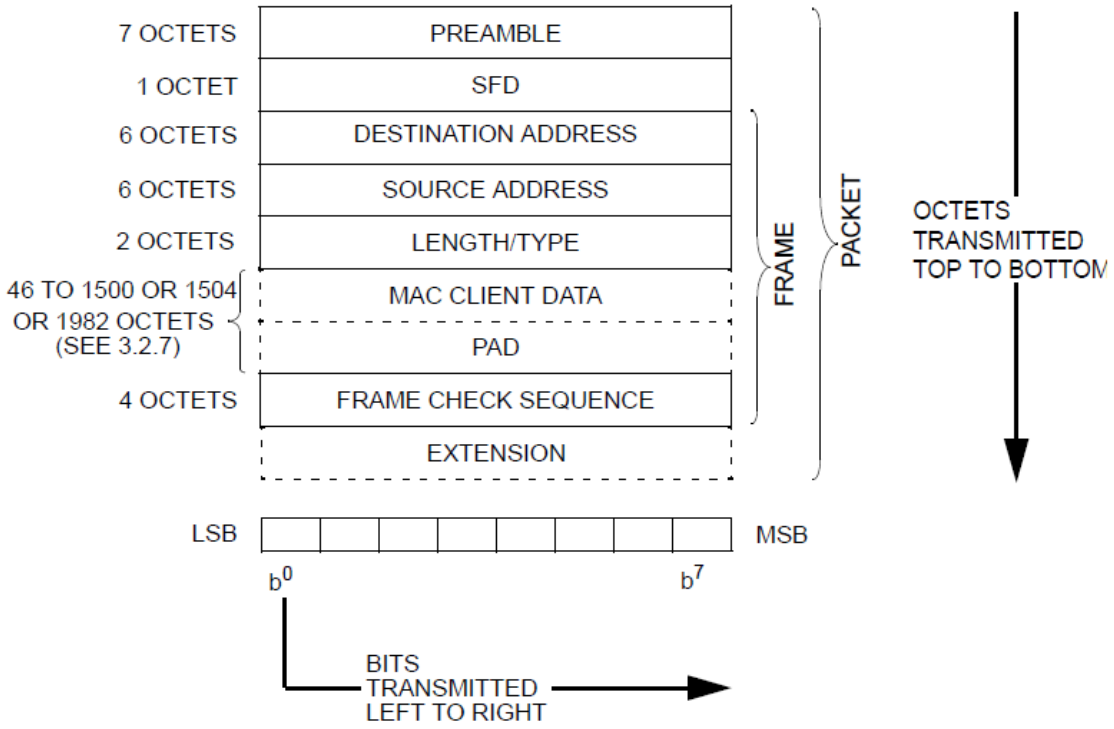
The IEEE 802.3 standard introduced a few changes in the original Ethernet data packet layout. This new layout of the IEEE 802.3 data frame was a result of the standardization committee efforts in adding certain connection services directly into the “Data Link Layer” instead of leaving it to upper layers as Ethernet does (see next section).

The first 8 bytes of the Ethernet preamble, a fixed sequence of bits used to identify the beginning of a valid data packet after an idle period in the transmission medium, was divided in the IEEE 802.3 standard into two fields: a 7-byte “Preamble” copied from the first 7 bytes of the original Ethernet preamble and a 1-byte “Start-of-Frame-Delimiter” copied from the last 1 byte of the original Ethernet preamble.

The next 12 bytes following the Preamble, however, remained the same. The first 6 bytes are used by Ethernet and IEEE 802.3 as the “Destination Address”

of the targeted MAC sub-layer, as the next 6 bytes, which are used as the “Source Address” of the sending MAC sub-layer.

Figure 3.2 – IEEE 802.3 Frame format.



Source: IEEE 802.3-1 standard (2012)

The bit transmitting order in both Ethernet and IEEE 802.3 is LSB-first and the first 2 bits to be decoded by the receiving end have a special meaning in both standards. The first bit determines whether the MAC “Destination Address” is an “individual” (unicast transmission) or “group“ (multicast transmission). The second bit determines whether the MAC Destination Address is “locally administered” or “globally administered”. A all-1s MAC “Destination Address” (hexadecimal FF-FF-FF-FF-FF-FF) is interpreted as a broadcast transmission.

It is important to point out that MAC addresses have a building rule according to IEEE, which includes a leading 3-byte field called “Organizationally Unique Identifier” (OUI). Each company manufacturing devices that can be attached to an Ethernet or IEEE 802.3 network uses its own OUI to uniquely identify each piece of equipment produced. Since OUI occupies the first 3 bytes of the MAC address, it is usually a number multiple of 4 (with a few exceptions). This is rather convenient, for it leaves untouched the first 2 LSBs which have the special use just described.

The following 2-byte field used by Ethernet as “Type” (the “Ethertype”) to define what sort of upper layer protocol was carried by the data packet was used in the IEEE 802.3 standard either as “Length” or as “Type” in a clear attempt to reconcile the intention of the IEEE standardization committee in embedding the identification of the protocol carried by the data packet into the “Data Link Layer” and the already existing large customer base Ethernet enjoyed in the beginning of the 80’s.

This “reconciliation” rule is simple, as stated in the IEEE 802.3 standard document:

a) If the value of this field is less than or equal to 1500 decimal (05DC hexadecimal), then the Length/ Type field indicates the number of MAC client data octets contained in the subsequent MAC Client Data field of the basic frame (Length interpretation).

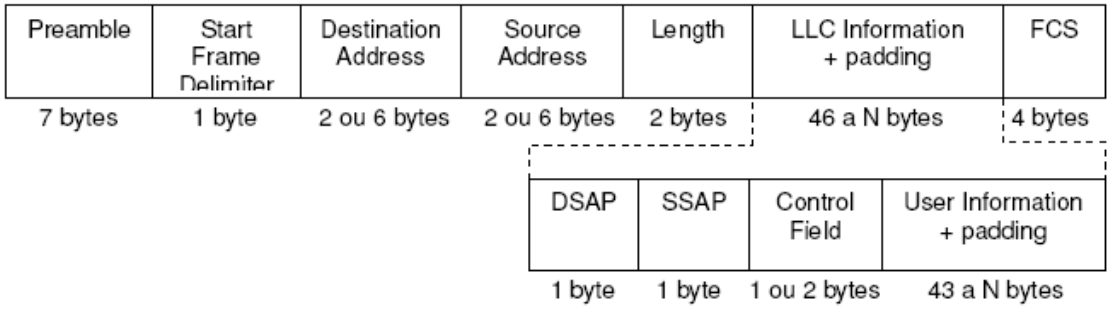
b) If the value of this field is greater than or equal to 1536 decimal (0600 hexadecimal), then the Length/Type field indicates the Ethertype of the MAC client protocol (Type interpretation).

The Length and Type interpretations of this field are mutually exclusive.

As a measure of how infrequent is the use of the “Length” interpretation, suffice it to say that the Internetworking Protocol known to us as IP has an “Ethertype” of 0800 hexadecimal, therefore falling in the case (b) above.

The embedding of the identification of the protocol carried by IEEE 802.3 packets when the “Type/Length” field is interpreted as “Length” is covered by the IEEE 802.2 standard addressed in the next section.

Figure 3.3 – IEEE 802.3 Frame with IEEE 802.2 LLC field.



2.4 IEEE 802.2 Logical Link Control

This IEEE 802.2 standard (IEEE, 1998) is not new. Its first supplements were published in 1993. The last and final version was published in 1998. It covers the “Logical Link Layer” (LLC) sub-layer of the “Data Link Layer” of the IEEE 802.3 standard.

As stated in the standard’s text:

“This International Standard provides a description of the peer-to-peer protocol procedures that are defined for the transfer of information and control between any pair of data link layer service access points on a LAN.”

The standard describes “service” as a means of accessing capabilities provided by upper communication layers. Using a “Service Access Point” (SAP) is how one reaches a particular “service”. A SAP can be understood as a logical construct managed by a software component belonging to an upper network layer.

The “Logical Link Control” (LLC) is defined as the upper sub-layer of the “Data Link Layer”, where “Media Access Control” is the lower sub-layer. LLC describes the connection services available to SAPs.

The data structure used in LLC is called “Protocol Data Unit” (PDU). The PDUs has following fields:

Address Fields:

Destination Service Access Point (DSAP) – Contains the destination SAP of the PDU.

Source Service Access Point (SSAP) – Contains the source SAP of the PDU.

Control Field - Designate command and response functions, may contain sequence numbers when required.

Information Field – Contains zero or more bytes of information.

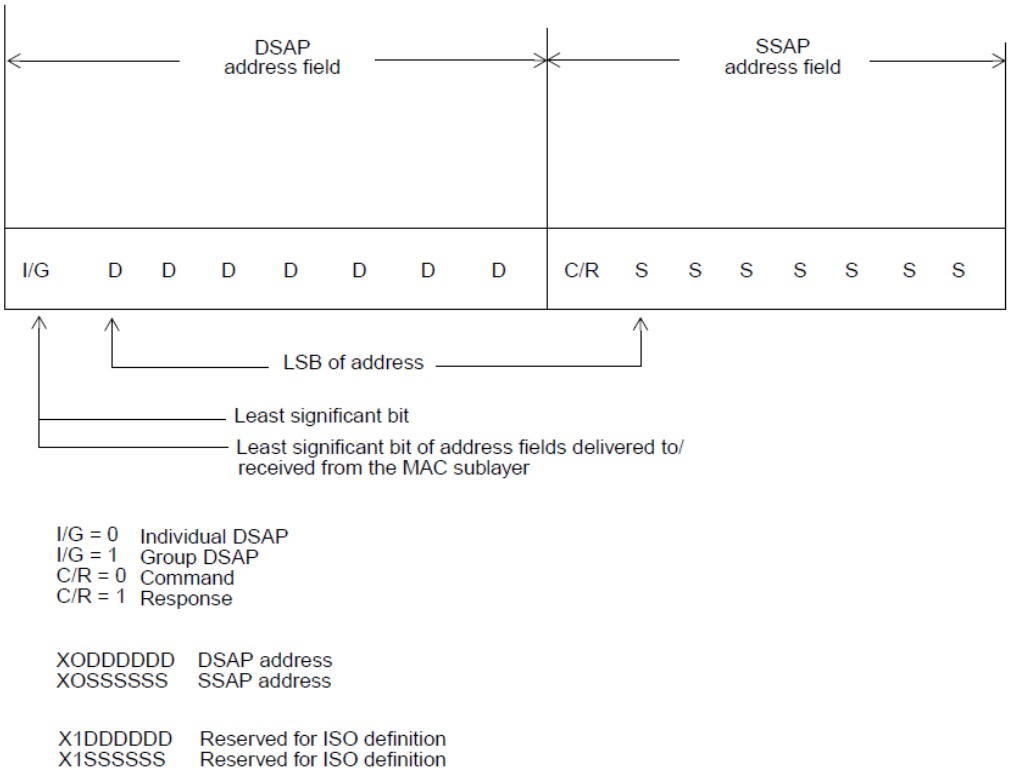
Figure 3.4 – IEEE 802.2 LLC Fields.

DSAP address	SSAP address	Control	Information
8 bits	8 bits	8 or 16 bits	M*8 bits

DSAP address	=	Destination service access point address field
SSAP address	=	Source service access point address field
Control	=	Control field [16 bits for formats that include sequence numbering, and 8 bits for formats that do not (see 5.2)]
Information	=	Information field
*	=	Multiplication
M	=	An integer value equal to or greater than 0. (Upper bound of M is a function of the medium access control methodology used.)

Source: IEEE 802.2 standard (1998)

Figure 3.5 – IEEE 802.2 Address Fields.



Source: IEEE 802.2 standard (1998)

Each SAP address field has seven bits of actual address and one least significant bit used in the DSAP address field to identify the DSAP address as either an individual (bit set to “0”) or a group address (bit set to “1”) and in the SSAP address field to identify that the LLC PDU is a command (bit set to “0”) or a response (bit set to “1”). The second least significant bit set to “1” indicates a “reserved” address.

All “1”s in the DSAP address field is said to be the “Global” DSAP address and all “0”s in the DSAP or SSAP address field is said to be the “Null” address.

In this work, both DSAP and SSAP addresses will use the least significant bit set to “0” (thus following a “0xxxxxxx” format), giving 126 “non-null” different SAP addresses (all even decimal numbers).

The standard defines three different types of services:

Type 1 Operation: PDUs shall be exchanged between two LLC layers without the need for the establishment of a data link connection.

Type 2 Operation: A data link connection shall be established between two LLC layers prior to any exchange of information-bearing PDUs.

Type 3 Operation: PDUs shall be exchanged between two LLC layers without the need for the establishment of a data link connection.

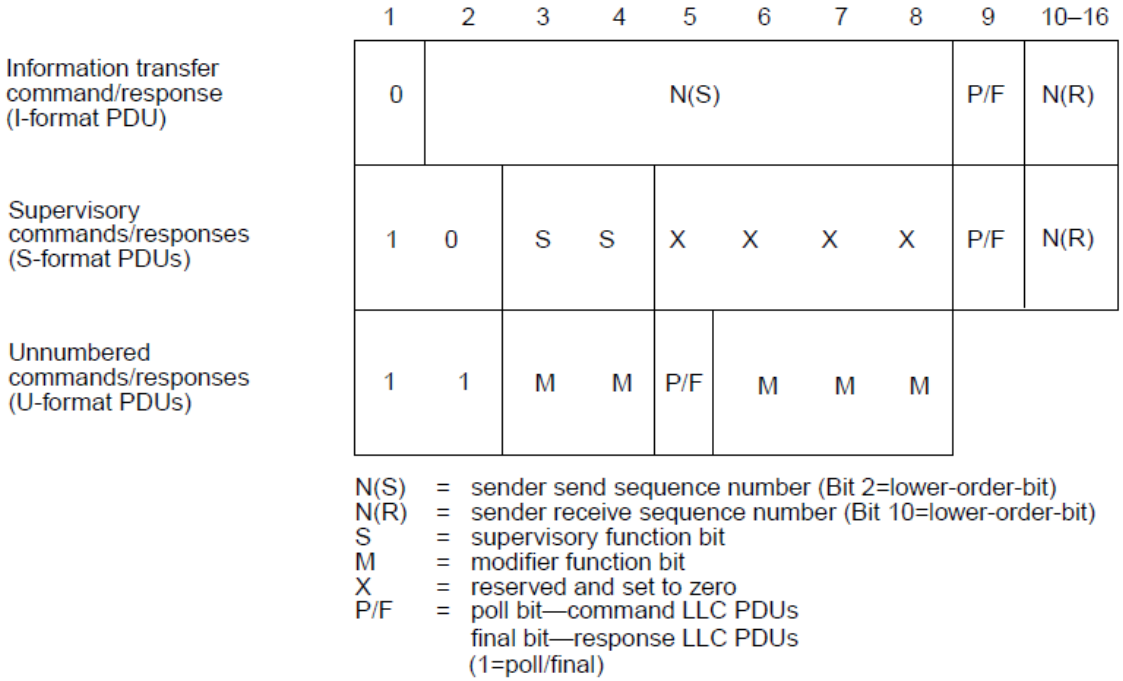
There are three different Control Field formats:

Information transfer format: The I-format PDU shall be used to perform numbered information transfer in Type 2 operation.

Supervisory format: The S-format PDU shall be used to perform data link supervisory control functions in Type 2 operation.

Unnumbered format: The U-format PDUs shall be used in Type 1, Type 2, or Type 3 operation to provide additional data link control functions and to provide unsequenced information transfer.

Figure 3.6 – IEEE 802.2 PDU Control Field.



Source: IEEE 802.2 standard (1998)

To this work, only Type 1 Operation and Unnumbered Command/Response (U-format PDUs) will be relevant.

There are three types of U-format Commands and Responses PDUs in Type 1 Operation:

Unnumbered information (UI) Command

The UI command PDU shall be used to send information to one or more LLCs. There is no LLC response PDU to the UI command PDU.

Exchange identification (XID) Command/Response

The XID command PDU shall be used to convey the types of LLC services supported (for all LLC services. The XID response PDU shall be used to reply to an XID command PDU at the earliest opportunity

Test (TEST) Command/Response

The TEST command PDU shall be used to cause the destination LLC to respond with the TEST response PDU at the earliest opportunity, thus performing a basic test of the LLC to LLC transmission path.

Figure 3.7 – Type 1 operation command control field bit assignments.

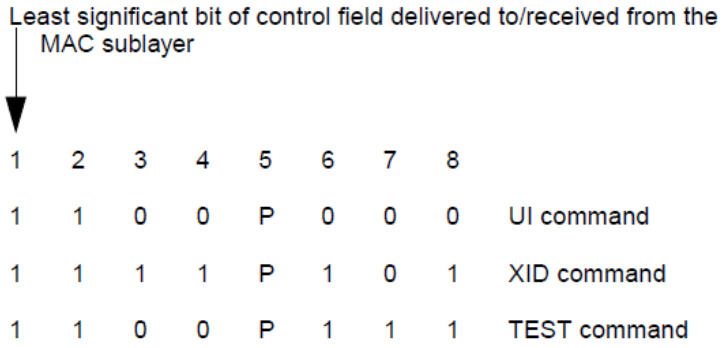
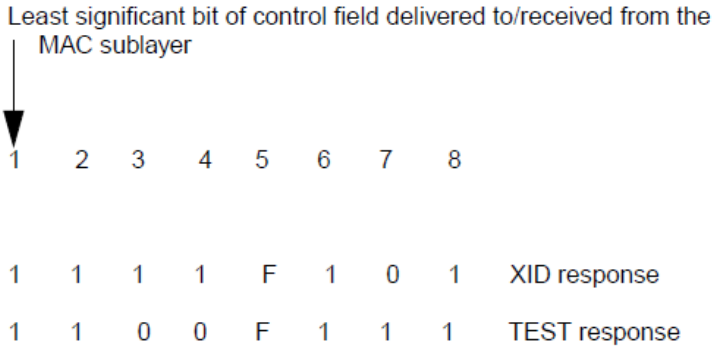


Figure 3.8 – Type 1 operation response control field bit assignments.



Source: IEEE 802.2 standard (1998)

To this work, only UI and TEST PDUs will be relevant.

The only difference between Type 1 Operation Command and Response PDUs is on the fifth LSB bit of the Control Field named “Poll/Final” (P/F) bit. On a XID or TEST Command and Response PDUs, the P/F bit shall be set to “F” (“1”) and on a UI Command it shall be set to “P” (“0”).

In the 802.2 standard there is an extension called “Subnetwork Access Protocol” (SNAP) which was created to provide to upper layer protocols the same Ethertype field used in Ethernet.

The SNAP header consists of the 3-byte “Organizationally Unique Identifier” (OUI) followed by a 2-byte Protocol ID. If the OUI is hexadecimal 000000, the protocol ID is the Ethernet type (Ethertype) field value for the protocol running on top of SNAP.

The SNAP header is to be found only in UI PDUs which have both DSAP and SSAP fields filled with hexadecimal AA and Control Field set to hexadecimal 03 (the first two LSB bits set as required for U-format PDU and P/F bit set to “0”).

The historic reason for this apparently unnecessary complication, which in fact reduces the number of usable data bytes in the IEEE 802.3 network packet, was that in its design the LLC “Service Access Point” (SAP) is just 7 bits long, allowing for at most 128 different combinations. As vendors started registering more and more different communication protocols, it became clear to IEEE that soon 8 bits would not be sufficient. As a result, the SAP hexadecimal AA was reserved and the SNAP extension created.

2.5 Internet Protocol (IP)

In May 1974, the Institute of Electrical and Electronics Engineers (IEEE) published a paper entitled "A Protocol for Packet Network Intercommunication" The paper's authors, Vinton Cerf and Robert Kahn, described an network protocol for sharing resources using packet switching among network nodes. A central control component of this model was the "Transmission Control Program" that incorporated both connection-oriented links and datagram services between nodes. The monolithic “Transmission Control Program” was later divided into a modular architecture consisting of the “Transmission Control Protocol” (TCP) at the Transport Layer and the “Internet Protocol” (IP) at the Network Layer. This layered model became known as the United States’ Department of Defense (DoD) “Internet Model and Internet protocol suite”, and informally as “TCP/IP”.

The “Advanced Research Projects Agency Network” (ARPANET) initially founded by the Advanced Research Projects Agency (ARPA) of the US DoD was first network to implement the TCP/IP protocol suite.

IP versions 0 to 3 were experimental versions developed between 1977 and 1979. The protocol version in use today is version “4” (IPv4) introduced in September of 1978 and “4” is the protocol version number carried in every IP packet connecting computers, mobile phones and other communication devices all over the world.

The successor to IPv4 is IPv6. Its most prominent difference from version 4 is the size of the addresses. While IPv4 uses 32 bits for addressing, yielding 4.3 billion (4.3×10^{9}) different addresses, IPv6 uses 128-bit addresses providing 3.4×10^{38} different addresses.

Another Transport Layer protocol operating over IP is worth mentioning: the “User Datagram Protocol” (UDP) which was introduced in 1980. Unlike TCP, which implements a reliable connection-oriented communication between network nodes, UDP is suitable for purposes where error checking and recovery are either not necessary or are performed at the Application Layer. UDP avoids the overhead of such processing in the protocol stack. Time-sensitive applications often use UDP because dropping packets is preferable over waiting for packet retransmission, which may not be an option in systems operating under real-time constraints.

3 DIGITAL DATA BUSES

This section lists three Digital Data Buses (DDB) which became an inspiration for this work: ARINC-429, ARINC-664 Part 7 (AFDX™) and SpaceWire.

Each one deserves an entire book due to relevance of each one of them in the aerospace industry. However, in order to be succinct, besides some historic background, only a few important characteristics will be enumerated such as topology, physical medium, medium access control, data encoding, transmission data frame format and others deserving special attention. The careful reader should turn to the references for more details on each DDB.

3.1 ARINC-429

"ARINC" stands for Aeronautical Radio Incorporated and "429" is the associated number to the specification document named "Mark 33 – Digital Information Transfer System (DITS)". The ARINC-429 specification (ARINC, 2001) was officially published in 1978 and usually referred to as a "unidirectional", "multi-drop" serial bus.

ARINC-429 allows "point-to-point" communication with the addition of an important feature: it allows for multiple receivers (up to 20) for one transmitter (hence the "multi-drop" attribute).

ARINC-429 physical medium is a 78Ω twisted-pair shielded copper cable. Data transmissions use Bipolar-Return-to-Zero (BPRZ) encoding (transition from high-to-low or low-to-high voltage levels at half bit-time). To protect data transmissions from interference, ARINC-429 uses two wires mirror-imaging voltage levels on them between -10V and +10V. Allowed transmission speeds are 12.5kbps or 100kbps with 4 bit-times bus idle (at 0 Volts) between two consecutive data frames.

Since ARINC-429 allows for only one transmitter on the physical medium, it does not require any access control to it. If Node A needs to communicate with Node B it uses one cable; if Node B needs to communicate back with Node A, it must use a second cable (hence the "unidirectional" attribute). Normally,

devices communicating over ARINC-429 have separate circuitries for transmitting and receiving.

Data frames are 32 bits long with up to 19 bits of data, 1 bit of odd-parity and 12 bits overhead, including an 8-bit frame identifier (the “Label”).

Mostly because of its simplicity and reliability, communication links following the ARINC-429 specification are in current use, unmodified since its formal publication.

3.2 ARINC-664 Part 7 (AFDX™)

The 7th part of ARINC specification “664” received the title “AVIONICS FULL-DUPLEX SWITCHED ETHERNET”. It describes what was called “Determinist Networks” within the more general concept of “Aircraft Data Networks” (ADN) introduced by the ARINC-664 specification (ARINC, 2009), which now has 8 parts:

Part 1 - Systems Concepts and Overview

Part 2 - Ethernet Physical and Data-Link Layer Specifications

Part 3 - Internet-based Protocols and Services

Part 4 - Internet-based Address Structure and Assigned Numbers

Part 5 - Network Domain Characteristics and Functional Elements

Part 6 - Reserved;

Part 7 - Deterministic Networks

Part 8 - Upper Layer Protocol Services

The Part 7 of the ARINC-664 specification was developed around a concept created inside AIRBUS called “AFDX™” for the A380 project. “AFDX” is now a brand name belonging to EADS which restricts its use in commercial products.

According to the text, the Part 7 describes a special case of what the ARINC-664 specification calls “profiled networks”, which in turn is a special case of

“compliant networks” (refer to ARINC-664 Part 1). The networks which are “compliant” with ARINC-664 are IEEE 802.3 and IP. The term “profiled” refers to some restrictions imposed to both IEEE 802.3 and IP networks, for instance, network addresses shall be fixed in each specific installation.

The term “SWITCHED” in the title of the ARINC-664 Part 7 (A664-P7) specification suggests that this type of network requires a switching hardware; therefore it follows a “star” topology which can be combined into a “multi-star” topology. Nodes in an A664-P7 network are called “End-Systems” (ES).

Also according to the text, the most important feature of A664-P7 networks is “Quality-of-Service”, in particular timely delivery of data. To achieve this goal, several special elements were introduced, modifying how data packets are assembled and delivered throughout the network.

The first important element introduced is called “Virtual Link” (the “VL”). The VL is a unidirectional logical communication link with guaranteed bandwidth. Being “unidirectional” has its consequences in an A664-P7 network: if ES A needs to communicate to ES B it needs one VL and if ES B needs to communicate back to ES A it needs a second VL.

It is interesting to note that this “unidirectional” characteristic of the VL is implemented in a full-duplex IEEE 802.3 physical medium. This means that an End-System can simultaneous transmit and receive data using a single cable but using two VLs. If one compares this situation with the one described in the previous section, he or she will immediately find similarities – at least from a logical perspective – with two nodes communication via ARINC-429. No surprises here, because of one of the key issues driving AIRBUS toward AFDX was precisely the virtualization of an ARINC-429 “point-to-point” network on an IEEE 802.3 infrastructure. With benefits, such as electrical cabling simplification and a 103-fold increase in transmission speed.

The A664-P7 standard itself recognizes this by saying: “In a system with many end points, point-to-point wiring is a major overhead. Ethernet networks can offer significant advantages and a suitable model for a deterministic network can be obtained through emulating the point-to-point connectivity”

The “multi-drop” attribute of ARINC-429 data bus is usually realized by splitting cables or working on cable connectors. The same “multi-drop” attribute applies to A664-P7 networks, however through another important element: the “A664-P7 Switch”. The technical specifications of this type of equipment were derived from those found in Ethernet Switches, but with special attention to the restrictions imposed by the ARINC-664 standard.

Another consequence of the “multi-drop” attribute of A644-P7 networks is that VLs must support multicast transmissions. This is realized my using special classes of IEEE 802.3 and IP network addresses built around VLs.

The A664-P7 Switch can provide “Traffic Policing” as any commercially available Ethernet Switch, and policing is essential to the deterministic nature of A664-P7 networks. However, a A664-P7 Switch is not allowed to “auto-discover” routing paths for network data packets as any Ethernet Switch does: routing paths must be statically configured for each VL and made effective at A664-P7 Switch power-on.

While A664-P7 Switches are expected to do “Traffic Policing” on incoming network traffic, another element is required to secure bandwidth to VLs. An End-System that transmits data on A664-P7 networks need to provide “Traffic Shaping” on each VL, that is, no VL is allowed to transmit more than it is expected to.

On A664-P7 networks, two parameters are used for defining the allowed bandwidth of a VL:

Lmax: the maximum packet size a VL can transmit expressed in bytes;

Bandwidth Allocation Gap (BAG): the minimum amount of time separating two consecutive data packets transmitted on the VL expressed in milliseconds.

The bandwidth for a VL is defined by the quantity (Lmax+20)/BAG.

It is the responsibility of each node in an A664-P7 network to perform “Traffic Shaping” on each VL it uses. However, an effect called “transmission jitter” is

observed and needs to be dealt with on each A664-P7 End-System. VLs are streams of data that share the same transmission port on an A664-P7 End-System; therefore the transmission carried out on one VL may suffer interference from transmissions from other VLs, since data packets line up for reaching the physical medium.

This “transmission jitter” is the maximum amount of time one expects to affect the BAG of a particular VL. If the maximum and the minimum amount of time observed on a VL separating two consecutive data packet transmissions are BAG plus “X” and BAG minus “Y” milliseconds respectively, the “transmission jitter” is the quantity “X plus Y” for that particular VL.

The “transmission jitter” is an important quantity for the “Traffic Policing” function performed by the A664-P7 Switch. If an A664-P7 ES is responsible for “shaping” network traffic on each VL, the A664-P7 Switch is responsible for “policing” the incoming traffic in for each VL. Without “transmission jitter”, policing is simple and it is sufficient to verify that the bandwidth associated to a VL is not exceeded. With “transmission jitter”, a VL is allowed an “overdraft” to compensate for oscillations in the data packet transmission period represented by the parameter BAG.

“Traffic Shaping” and “Traffic Policing” working together should give A664-P7 networks its “determinist” behavior, although it would be more appropriate to describe this quality as “bounded data delivery”. After all, what analytic methods permit in A664-P7 networks is the estimation of a “bound” for the arrival pattern of network data frames.

A third important element in A664-P7 networks is the introduction of the concept of “ports” through the use of the “User Datagram Protocol” (UDP) over IP. A “port” is a virtual construct that allows data exchange between applications running in different ES. A664-P7 networks take advantage that UDP, a Transport Layer protocol which already defines “ports” for the same purpose.

Since A664-P7 use IEEE 802.3 and IP, fragmentation of data packets is supported. On IP, fragmentation is governed by the quantity “Maximum Transfer Unit” (MTU) expressed in bytes: any data packet with size bigger than MTU is

split in two or more fragments reassembled at the receiving end. With IP over IEEE 802.3, the value of MTU is 1500 bytes, on A664-P& networks; MTU is equal to the parameter Lmax for each VL. That is, on an A664-P7 network, a data packet with size bigger than Lmax for a particular VL is split in two or more fragments. According to the ARINC-664 Part 7 standard, the maximum packet size allowed is 8,192 bytes.

Data encoding on A664-P7 networks follows the IEEE 802.3 paradigm at 100 megabits per second (“Manchester” encoding also called “Phase Encoding” - PE). Data frames follow the UDP/IP over IEEE 802.3 paradigm with special rules for assembling IEEE 802.3 and IP destination and source addresses. And one important, 1-byte sized exception: A664-P7 data frames are numbered from “0” to “255” using a field located at the very end of each data packet called “Sequence Number” (SN). Because of this SN field, the maximum payload size of an A664-P7 data packet is one byte less than that of a “normal” UDP/IP over IEEE 802.3 data packet.

The SN field is the resource chosen for implementing two special layers into the otherwise IEEE 802.3/Ethernet standard compliant A664-P7network. These are called “Redundancy Management” (RM) and “Integrity Checking” (IC). Data packet transmissions on A664-P7 networks occur using two redundant physical links which transport two identical copies of each data packet. The IC layer checks whether data packets have consecutive SN and the RC layer discards the second copy once it receives and validates the first copy.

The ARINC-664 Part 7 specification became the “de facto” standard for large avionics networks since its formal publication in 2005.

3.3 Spacewire

In its own words, “SpaceWire links are full-duplex, point-to-point, serial data communication links” (ECSS, 2008).

At first, one could believe that being declared “point-to-point”, SpaceWire restricts the topologies it can be cover. However, it actually supports “multi-star”

topologies with the introduction of “routing switches” and an associated routing protocol.

The SpaceWire standard is divided into “clauses”, six of them dedicated to a protocol level:

Clause 5 (Physical Level) covers cables, connectors, cable assemblies and printed circuit board tracks.

Clause 6 (Signal Level) deals principally with electrical characteristics, and coding and signal timing.

Clause 7 (Character Level) describes how data and control characters are encoded.

Clause 8 (Exchange Level) presents the way in which a SpaceWire link operates including link initialization, normal operation, error detection and error recovery.

Clause 9 (Packet Level) describes the way in which data is encapsulated in packets for transfer across a SpaceWire network.

Clause 10 (Network Level) deals with the structure and operation of a SpaceWire network.

SpaceWire was designed for moving large amounts of data reliably between two electronic units installed in a spacecraft. It provides mechanisms for securing link stability and link recovery following detection of an error condition and also a mechanism for finding alternate data traffic routes to overcome occasional link congestion. It also provides flow control on both transmitting and receiving sides of each node.

The packet structure is very simple: it defines a header which contains the routing information, a payload and an end-of-packet marker.

Data inside the packet is encapsulated as “Characters”. They can be either 10-bit “Data Characters” or 4-bit “Control Characters”. One particularly important “Control Character” is the “Flow Control Token” (FCT), used in regulating traffic between two nodes.

If one node is prepared for receiving data from other node, that is, it has enough memory space on its receiver electronics for admitting data characters, it must transmit a packet containing an FCT. Receiving an FCT authorizes the transmitting node to send 8 characters and sending an FCT sets the receiving node to expect 8 characters. The transmitting node keeps a credit count of how many characters it is allowed to send and the receiving node likewise keeps a credit count of how many characters it has allowed to receive. Each time the transmitting node sends a data character, it decrements the transmit credit count by one. Each time the receiving node receives a data character, it decrements the receive credit count by one. The standard specifies that the maximum number of outstanding data characters on either the transmitting or receiving side is 56.

Routing in SpaceWire deserves special attention due to its clever implementation. To support “multi-star” topologies by cascading routing switches, enough routing information is inserted in the beginning of the data packet as a sequence of 8-bit fields informing the switch output port whereto the data packet should be forwarded. As the data packet crosses a routing switch, the first leading character is removed and only remaining characters are forwarded to the output port.

Another important feature in SpaceWire is what the standard calls “wormhole routing”, described in the text as follows:

“*As soon as the header for a packet is received the switch determines the output port to route the packet to by checking the destination address. If the requested output port is free then the packet is routed immediately to that output port. That output port is now marked as busy until the last character of the packet has passed through the switch*”

This mechanism is not new and a similar approach called “cut-through” was used in the first commercially available Ethernet switches (CISCO, 2019).

Broadcast and multicast are also supported by the standard, but these forms of packet distribution are treated as particular cases in the routing switch programming, unlike IEEE 802.3/Ethernet which use special network addresses for the same purpose.

SpaceWire physical medium operates with “Low Voltage Differential Signaling” (LVDS) using a low voltage swing (from -400mV to +400mV over 1.2V level). Data encoding is “Data-Strobe” (DS) with one line for Data and one line for Strobe. The data is transmitted “NRZ style” (high voltage level is interpreted as “1” and low voltage level is interpreted as “0”) and the strobe signal changes state whenever the data remains constant from one data bit time to the next.

SpaceWire cables comprise four twisted pair wires with a separate shield around each twisted pair and an overall shield. The standard provides detailed information not only about the cable construction, but also about connector types and other wiring requirements.

Supported data transmission speeds range from 2 megabits per second to 400 megabits per second, what places SpaceWire on the top of the list of DDBs for this particular attribute.

3.4 And Many Others

It would be unfair not to list other DDBs which were developed by the industry as the industry needed them (summary adapted from Wikipedia):

3.4.1 MIL-1553B

MIL-STD-1553 (MIL STANDARD, 2019) was first published as a U.S. Air Force standard in 1973, and first was used on the F-16 Falcon fighter aircraft. It was originally designed as an avionic data bus for use with military avionics, but has also become commonly used in spacecraft on-board data handling (OBDH) subsystems.

3.4.2 RS-232

In telecommunications, RS-232, “Recommended Standard” 232 (EIA STANDARD, 1969), refers to a standard originally introduced in 1960 for serial communication transmission of data. It formally defines signals connecting between a DTE (Data Terminal Equipment) such as a computer terminal, and a DCE (Data Communication Equipment), such as a modem.

3.4.3 RS-422

RS-422 is a technical standard originated by the Electronic Industries Alliance (TIA/EIA STANDARD, 1994) that specifies electrical characteristics of a differential signaling that can transmit data at rates as high as 10 Mbit/s, or may be sent on cables as long as 1,500 meters. Some systems directly interconnect using RS-422 signals, or RS-422 converters may be used to extend the range of RS-232 connections.

3.4.4 RS-485

RS-485 (EIA STANDARD, 1983) supports inexpensive local networks and multidrop communications links, using the same differential signaling over twisted pair as RS-422. These characteristics make RS-485 useful in industrial control systems and similar applications.

3.4.5 CAN

A Controller Area Network (CAN bus) is a robust vehicle bus standard designed to allow microcontrollers and devices to communicate with each other in applications without a host computer (ROBERT BOSCH, 1991). It is a message-based protocol, designed originally for multiplex electrical wiring within automobiles to save on copper, but is also be used in many other contexts. Development of the CAN bus started in 1983 at Robert Bosch GmbH. The protocol was officially released in 1986 at the Society of Automotive Engineers (SAE) conference in Detroit, Michigan. Bosch published several versions of the CAN specification and the latest is CAN 2.0 published in 1991.

3.4.6 ARINC-629

The ARINC-629 computer bus was introduced in May 1995 and was first used on the Boeing 777 (SAE-ITC STANDARD, 2019). The ARINC-629 bus operates as a multiple-source, multiple-sink system, where each terminal can transmit data to, and receive data from, every other terminal on the data bus. While some people expected that the Boeing 777 would be the first and last aircraft to use ARINC-629 data bus, it is also used on the Boeing 737 MAX and Airbus A330 and A340.

3.4.7 TTP

The Time-Triggered Protocol (TTP) is an open computer network protocol for control systems (TTTECH, 2003). It was designed as a time-triggered field bus for vehicles and industrial applications and standardized in 2011 as SAE AS6003 (TTP Communication Protocol). TTP was originally designed at the Vienna University of Technology in the early 1980s. In 1998 TTTech Computertechnik AG took over the development of TTP, providing software and hardware products.

3.4.8 TT Ethernet

The Time-Triggered Ethernet (SAE AS6802) standard defines a fault-tolerant synchronization strategy for building and maintaining synchronized time in Ethernet networks, and outlines mechanisms required for synchronous time-triggered packet switching for critical integrated applications, such as integrated modular avionics architectures (TTTECH, 2008).

TT Ethernet technology implemented by TTTech was used in the Orion Multipurpose Crew Vehicle (MPCV), a NASA spacecraft designed to take a crew of up to six astronauts to destinations beyond Low Earth Orbit including the Moon and Mars. The brains of the Orion spacecraft (NASA, 2019) is the Vehicle Management Computer (VMC), a single electronics unit consisting of four independent modules that deliver the processing capability for the Orion spacecraft and communicate with the other avionics of the Orion spacecraft via redundant Ethernet connections using the TTEthernet Network Interface Controllers and network switches.

3.4.9 FlexRay

FlexRay is an automotive network communications protocol developed by the FlexRay Consortium to govern on-board automotive computing (FLEXRAY, 2005). It was designed to be faster and more reliable than CAN and TTP. The FlexRay consortium disbanded in 2009, but the FlexRay standard is now a set of ISO standards, ISO 17458-1 to 17458-5.

4 EMBEDDED NETWORKS

Networks designed to be installed in a completely segregated environment such as in aerospace vehicles present particularities that differentiate them from ordinary commercial networks.

Embedded networks serve the purpose of connecting functions hosted by electronics modules that interact to serve a greater purpose, for instance, providing global communication or Earth climate survey.

These systems need to have their behavior predicted during their design, therefore networks need to present a level of deterministic behavior while connecting functional elements of such systems.

These elements can be data producers, data consumers or both, and the relations between them are in general defined quite early in the system design phase.

Further, certain functions which give an embedded network the desired deterministic behavior are created and deployed at network elements as needed.

The next two sections explore the two aspects of embedded networks and their relevance to the design of the system and of the network serving it.

4.1 Data Producers and Data Consumers

In a complex and high integrated distributed processing system, in particular those present in aerospace vehicles, it is essential for a proper design to identify how engineering data flows from one part of the system to other parts of the system.

For instance, positioning data produced by a sensor installed in a satellite which tracks the Sun needs to flow to the energy supply subsystem which is responsible for moving the solar panels for optimal electric power generation. In a “fly-by-wire” flight control system present in modern aircraft, data must flow from pressure sensors calibrated for indicating altitude and airspeed, from accelerometers calibrated for indicating body acceleration, from the engines and from the pilot command inceptors to a central computer which is

responsible for properly moving flight control surfaces ensuring a smooth flight path.

The communication paths connecting parts of a distributed system are the result of an analysis identifying Data Producers and Data Consumers.

The important questions that need to be answered are:

Which information is required for the system to operate as designed?

Which parts of the system produce what information?

Which parts of the system consume what information?

Once Data Producers and Data Consumers are connected, a basic system topology emerges. It may indicate that “point-to-point”, “star” or a mix of the two topologies may seem more appropriate. However, other aspects of the communication infrastructure need to be addressed, such as physical distance between transmitters and receivers and any timing requirements that may affect how well Data Consumers process received data. These two aspects and perhaps others may alter the initial perception of the most suitable network topology for a given system and may limit the choice of the physical transmission medium.

The format in which data is produced and consumed is also very relevant. Sensors most commonly convert a physical quantity, such as air pressure, into a voltage level which can be calibrated to express a measure of altitude in meters or feet. Modern sensors can provide digital data, but it is not uncommon that their output also need some form of calibration. Further, if a system using a sensor for producing pressure altitude in meters needs to send this data to a system which consumes pressure altitude in feet (such as the Multi-function Display in the airplane cockpit), it must be converted before it is consumed. If mathematical operations are required for data formatting, care must be taken not to deteriorate the resolution required for the proper use of the data.

4.2 Essential Services in Embedded Networks

Letting devices communicate over a network in a complex and highly integrated processing environment onboard aerospace vehicles is quite an engineering challenge.

Unlike a network in a household where any configuration is almost never required, every aspect of the data exchange between any two participants in such embedded network has to be identified and documented. For this task, it is usual to produce "Interface Control Documents" (ICD) describing messages being transmitted by one software application in one network node and being received by an application (or applications) in one or more network nodes.

In general, networks connecting devices in aerospace vehicles have neither spurious messages nor unplanned communication paths: everything is pre-planned and rigorously tested before entering operation.

Certain pieces of software, such as Attitude Control in satellites or Flight Controls in aircraft are very sensitive to unplanned data transport delays while crossing a communication channel. In such cases, system designers strive to ensure communication determinism, that is, the behavior of the network when in operation can be predicted while the whole system is still in its design phase.

Networking in closed environments such as in aerospace vehicles involves aspects other than simply transmitting and receiving binary data. For instance, a system designer may want to restrict the amount of data one communication is supposed to carry per unit of time, or may want to make sure that one particular message goes to one node and not to any other node. These and other design concerns have driven the implementation of certain services present in complex networking scenarios, in particular those found in modern aerospace vehicles.

These services can be listed serving different network layers:

PHYSICAL LAYER

- Data Encoding
- Data Decoding

DATA LINK LAYER

- Media Access Control
- Data Validation
- Data Destination Validation
- Data Source Validation

Routing

Traffic Shaping

Traffic Policing

NETWORK LAYER

Data Validation

Fragmentation

Defragmentation

Routing

TRANSPORT LAYER

Data Validation

Application Destination Validation

Source Application Validation

Error Recovery

Data Encoding and Data Decoding at the Physical Layer can represent more or less binary data transmitted per a complete sine wave, while different Media Access Control strategies at the Data Link Layer may represent more or less transport delay in case of a failure accessing the physical medium.

Different checksum algorithms may represent higher or lower statistical probability of accepting corrupted data as valid at the Data Link layer, and different message routing implementation may introduce more or less transport delay when data has to be retransmitted to another network node. One must note that Data Validation is not the sole privilege of the Data Link Layer.

At the Data Link Layer it is also possible to protect a communication path from a misbehaving node by implementing Traffic Shaping and Traffic Policing, that is, “shaping” or constraining outgoing traffic and “policing” or forbidding incoming traffic according to some mathematical rule.

Fragmentation and Defragmentation (or reassembly) are usual at the Network Layer, because Transport Layer protocols tend to be agnostic of the limitations

imposed by the physical medium with respect to the quantity of data transmitted in a single operation.

Some form of Error Recovery is more common at the Transport Layer, whereby any inconsistency found processing the received data is communicated back to the transmitting node. At the Transport Layer is also where the upper layer protocols using the Internet Protocol (IP) are identified, for instance, Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). However, source and destination “Service Access Points” (SAP) as described in the IEEE 802.2 standard are available at the Logical Link Control (LLC) part of this standard’s Data Link Layer.

Depending on a design decision, these services can be allocated at different hardware and software elements involved in data communication across a network. For instance, a commercial Ethernet network card implements all the above services of the Physical Link Layer, but restricts itself to Destination Validation (it makes sure that the data is destined to its physical network address and discards it otherwise) and Data Validation (it makes sure that the received data is checked against the data checksum present in the last 96 bits of the received frame).

One node in a more complex network such as those following the ARINC-664 Part 7 standard must implement Traffic Shaping, as the standard dictates that a communication channel has a restricted bandwidth by design. By the same standard, which implements a “star” topology, a network switch must implement Traffic Policing for protecting the network from a node that does not perform the required Traffic Shaping properly.

4.3 High-Level Requirements for Embedded Network Protocols

Historically, network protocols were developed as a response to a real life need.

The original Ethernet was conceived inside XEROX Palo Alto Research Center (PARC) laboratory because there was a need for connecting workstations to a very expensive high-speed laser printer. The development of the Transmission Control Protocol (TCP) and the Internet Protocol (IP) was sponsored by the Department of Defense (DoD) of the United States within the scope of a project

conducted by the Defense Advanced Research Projects Agency (DARPA) for connecting geographically separated networks.

Design of networks expected to be installed in aerospace vehicles may benefit from the fact that nodes are just meters apart in an aircraft and confined in a less than one cubic meter space in a satellite, if we consider only installation issues. However, characteristics such as reliability and flexibility must be offered by embedded networks similarly to any other high-scale network.

The following high-level requirements derived from design concerns should be present in any embedded network protocol:

STATICALLY CONFIGURED

> The configuration of the protocol layers shall be statically defined and shall not change while the network is in operation.

FLEXIBILITY

> The services expected to be performed at the protocol layers shall be allocated on network elements where they can best preserve data integrity without penalizing the end-to-end transport delay experienced while crossing the network from a source to a destination.

RELIABILITY

> In the absence of any physical or electromagnetic interference, the protocol layers shall preserve data integrity as data flows from one source to one or more destinations.

MULTICASTING

> The protocol layers shall permit one-to-one as well as one-to-many communication paths.

FORWARDING

> The protocol layers shall permit a node to forward data to a destination other than the node itself.

ERROR DETECTION

> Each protocol layer shall provide a means of detecting errors when validating data received from the protocol layer immediately below.

FLOW CONTROL

There shall be a form of limiting the data flow going out of or coming in to any network element at a protocol layer level according to a fixed design parameter.

Other concerns related to the Physical Layer such as coding efficiency (number of significant binary digits transmitted per unit of time) and transmission rate (raw binary digits transmitted per unit of time) are not listed because they exceed the scope of this work, but they are not less important in the implementation of any network.

5 SPECIFICATION OF A NEW LAYER-2 PROTOCOL AND SERVICES

Among the Physical (Layer 1) and Data Link Layers (Layer 2) implementations developed in the last few decades, Ethernet and its IEEE standardization 802.3 is by far the most frequently used. In any household wireless access points route network traffic to commercial Internet service providers over Ethernet. In the factory floor, several implementations (check references) allow automated manufacturing of consumer electronics and cars. In commercial and military aircraft, Ethernet is present since the ARINC-664 standard parts 1 and 2 were published (check year).

Ethernet can be used in different network topologies, from its initial design as a “shared bus” to its current and by far most frequent “star” shaped, mix of these two and “point-to-point”, even when the latter seems limited to network maintenance scenarios.

However, Ethernet implementations used in aerospace vehicles also imply in using other Network (Layer 3) and Transport Layers (Layer 4), being the most frequent the Internet Protocol (IP) as the Network Layer and User Datagram Protocol (UDP) or Transmission Control Protocol (TCP).

The reason is that Ethernet, being a Data Link Layer protocol, does not provide a means of linking two application instances running in different network nodes. For that, a virtual construct needs to be defined and supported by associated services. In UDP and TCP over IP, this virtual construct is named “port”.

Therefore, for connecting a Data Producer to a Data Consumer in an embedded network such those present in modern aerospace onboard electronics apparently requires a Layer 4 protocol and associated services to transmit and receive data.

This means processing three network layers before being able to access data needed by an application for its continuing operation, which in time-critical situations, such as in controlling flight, may represent simply consuming extra processing time with no actual work being done.

To better serve time-critical applications, shortening the processing time required for extracting relevant data from a network transmission is a more than welcome characteristic of a network protocol.

In fact, the network standard IEEE 802.2 “Logical Link Control” (LLC) provides precisely this feature by specifying “Service Access Points “SAP” at the Data Link Layer (Layer 2).

The specification of a new Layer 2 protocol and associated services proposed in this work takes advantage of the IEEE 802.2 LLC protocol specification and will cover the construction of protocol frames, the behavior of the associated services at higher communication layers and how these are deployed in the network nodes in “point-to-point” and “star” network topologies.

6 NEXT STEPS

The next steps in the development of this new Data Link Layer (Layer 2) protocol specification involve:

- Definition of the new Protocol Data Unit (PDU);
- Definition of the associated services (data validation, traffic shaping, traffic policing, routing, transmission source and destination validation);
- Definition of methods for estimating PDU forwarding latency.

No laboratory experimentation is expected as a result of this work, due to lack of resources for building the required hardware and software environment.

BIBLIOGRAFIC REFERENCES

ARINC (Aeronautical Radio Incorporated). **ARINC specification 429**: Mark 33 Digital Information Transfer System (DITS) – Part 1 Functional Description, Electrical Interface, Label Assignments and Word Formats, 2001.

ARINC (Aeronautical Radio Incorporated). **ARINC specification 653-1**: Avionics Application Software Standard Interface, 2003.

ARINC (Aeronautical Radio Incorporated). **ARINC specification 664**: Aircraft Data Networks – Part 7, Deterministic Networks, 2009.

CISCO, **Cut-Through and Store-and-Forward Ethernet Switching for Low-Latency Environments**. Available at: https://www.cisco.com/c/en/us/products/collateral/switches/nexus-5020-switch/white_paper_c11-465436.html. Accessed on 02 sep. 2019.

EDISON TECH CENTER, **The V2 Rocket - how it works, guidance**. Available at: https://www.youtube.com/watch?v=Ph-npS29n9Q. Accessed on 02 sep. 2019.

EIA STANDARD, **RS-232-C Interface between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange,** Electronic Industries Association Engineering Department, Washington, USA, 1969.

EIA STANDARD, **RS-485 Electrical Characteristics of Generators and Receivers for Use in Balanced Multipoint Systems**, 1983.

ECSS (European Cooperation For Space Standardization), **SpaceWire** – Links, nodes, routers and networks, E-ST-50-12C Working Group. Noordwijk, The Netherlands, 2008.

FLEXRAY, **FlexRay Communications System - Protocol Specification - Version 2.1-Revision A**, December 2005.

IEEE (Institute Of Electrical And Electronics Engineers). **IEEE standard 802.3**: IEEE Standard for Ethernet. New York, NY, USA, 2012.

IEEE (Institute Of Electrical And Electronics Engineers). **IEEE standard 802.2**: Part 2: Logical Link Control. New York, NY, USA, 1998.

ISO (International Standards Organization), **ISO/IEC 7498-1:1994 Information technology - Open Systems Interconnection - Basic Reference Model**. Available at: https://www.iso.org/standard/20269.html. Accessed on 02 set. 2019.

LEXICO, **Definition of communication in English**. Available at: https://www.lexico.com/en/definition/communication. Accessed on 02 sep. 2019.

LEXICO, **Definition of topology in English**. Available at: https://www.lexico.com/en/definition/topology. Accessed on: 02 sep. 2019.

MERRIAN-WEBSTER, **Definition of communication**. Available at: https://www.merriam-webster.com/dictionary/communication. Accessed on 02-sep. 2019.

MIL STANDARD, **Digital Time Division Command/Response Multiplex Data Bus**. Available at: https://quicksearch.dla.mil/qsDocDetails.aspx?ident_number=275874. Acesso 02 sep. 2019.

NASA (National Space Agency), **Orion Spacecraft**. Available at: https://www.nasa.gov/exploration/systems/orion/index.html. Accessed on 02 sep. 2019.

ROBERT BOSCH, **CAN Specification Version 2.0**, 1991.

SAE AEROSPACE STANDARD, **Digital Time Division Command/Response Multiplex Data Bus AS15531**.
Available at: https://www.sae.org/standards/content/as15531/. Accessed on 02 sep. 2019.

SAE-ITC STANDARD, **ARINC-629 Multi-Transmitter Data Bus Parts 1 and 2**. Disponível em https://www.aviation-ia.com/sae-search/content/629. Accessed on 02 sep. 2019.

TIA/EIA STANDARD, **RS-422-B Electrical Characteristics of Balanced Voltage Digital Interface Circuits**, May 1994.

TTTECH, **Time-Triggered Protocol TTP/C High-Level Specification Document Protocol Version 1.1**, November 2003.

TTTECH, **TTEthernet Specification**, November 2008.

WIKIPEDIA, **V-2 rocket**, Available at:https://en.wikipedia.org/wiki/V-2_rocket. Accessed on 02 sep. 2019.

WIKIPEDIA, **Apollo Guidance Computer**. Available at: https://en.wikipedia.org/wiki/Apollo_Guidance_Computer. Accessed on 02 sep. 2019.