INTERNET-DRAFT
draft-avsolov-dtpdia-05.txt
Expires: March 1, 2005

A. Soloviev
Petrozavodsk State University

September 1, 2004

Data Transfer Protocol for Distributed Information Acquisition (DTP/DIA)

Status of This Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC 2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress".

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on March 1, 2005.

Copyright Notice

Abstract

The described in this memo protocol was developed for application in some distributed information measurement systems (IMS) at any stage of communication. It was designed for transmission of measured data from the measuring devices to the processing and storing equipment. Also it does support a common identification of the devices in distributed IMS. Due to its simplicity it can be easily implemented in firmware of any digital measuring device.

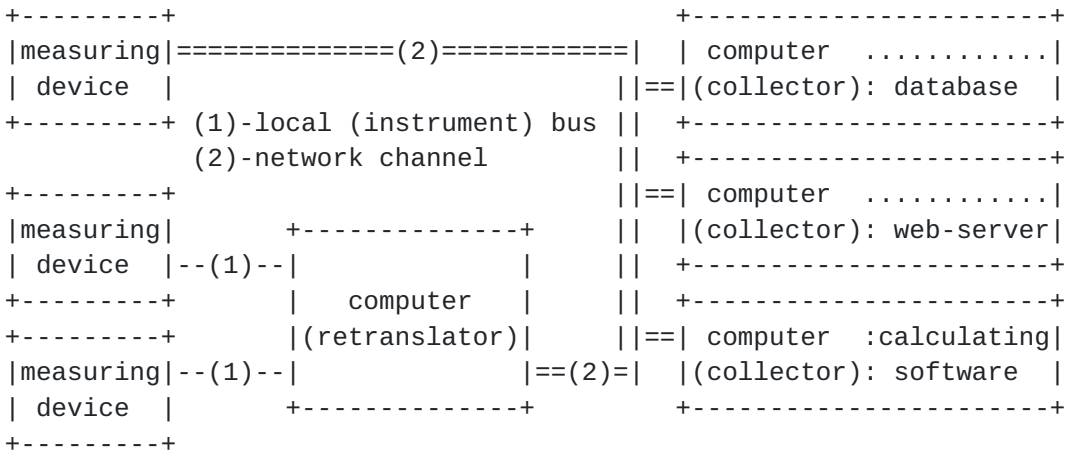
1. Introduction

The Data Transfer Protocol for Distributed Information Acquisition (DTP/DIA) was developed for application in distributed information measurement systems (IMS). DTP/DIA provides functionality of the presentation and application layers of the OSI Reference Model for the specified purposes.

1.1. Concept of Distributed IMS

IMS stands for a bundle of software and hardware that performs acquisition, processing, storing, and presentation of measured data. The systems with control function are out of scope. The simplest IMS consists of some measuring device connected to computer by means of some instrument interface (such as IEEE 488 (GPIB) or EIA/RS 232). A typical measuring device contains a sensor and a microcontroller that performs initial data acquisition and processing. In such a system the majority of IMS functions are given to a computer.

More complicated systems can be developed on the basis of CAMAC or VXI. But projects with large amount of investigated objects at far distances from each other require to install distributed IMS. Some nodes of distributed IMS should perform data transmission to nodes that process and store data. Sometimes this communication can be done by the measuring device itself if it is rather sophisticated, or this function may be performed by a computer. That is we deal with two kinds of data channels: local (1) and network (2).



In most cases the nodes of distributed IMS are considered to be the components of the open systems. So the OSI Reference Model can be applied to their communication channels. The upper layers of such systems are poorly standardized due to wide range of applications of IMS. The most of existing standards are vendor-specific. That's why it is necessary to introduce a simple protocol, suitable for both

kinds of data channels. DTP/DIA matches these requirements.

1.2. Remark about Terms

We avoid using the terms "client" and "server" in description of IMS. Instead we use the term "data source" for the object which transmits measured data and "data collector" for the node which receives measured data. If the node performs both reception and retransmission it will be named "retranslator". Obviously, in this terminology the measuring devices are "data sources", but a single computer could be either end-point "data collector" or "retranslator".

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

2. General Features

The protocol describes data transmission in small stand-alone packets. This feature allows using DTP/DIA over both streamed (connection-oriented) and message-oriented channels.

DTP/DIA provides several data presentation forms. Each form corresponds to a specific packet type. Some forms help to avoid implementation of floating point arithmetics in firmware of the device. There are forms that allow to transmit service or control information.

DTP/DIA allows transmission of measured data accompanied with measurement error and unit measure.

In distributed IMS it is very important to distinguish the measuring devices from each other at the application layer. So the protocol includes identification mechanism. This feature helps to transmit data from several data sources over a single communication channel. For example, one can use a device with several sensors (such device is represented as multiple data sources).

We suppose to apply DTP/DIA for both network and local channels. Local channels of some types don't provide reliable data delivery. To avoid unreliability of local interfaces DTP/DIA has the following primitive features: detection of packet start (packet leading sequence), check sum and time stamp.

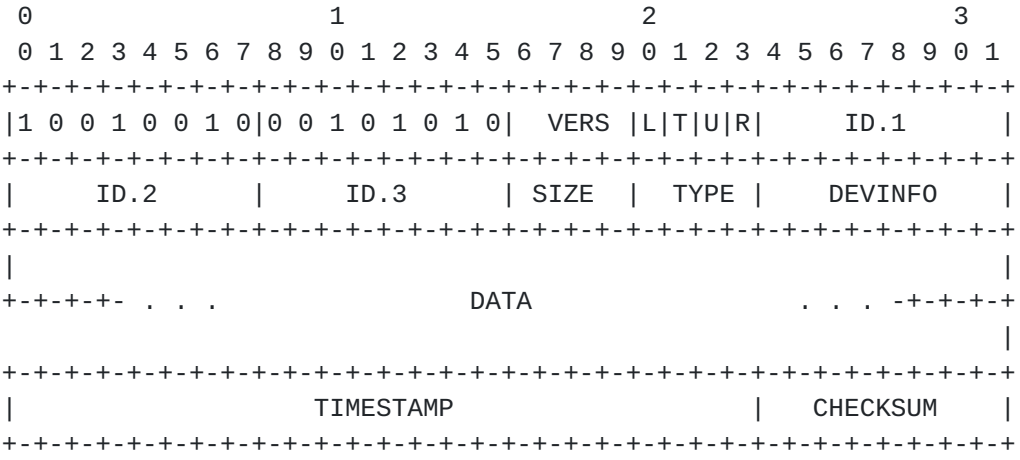
3. DTP/DIA Packet Format

DTP/DIA packet contains the fixed-sized Header block, the Data field and several optional fields. Here and further the octets numbering starts from 0.

Header block contains packet leading sequence, version field (VERS), flags field (L, T, and U bits), Data Source Identifier (ID.1, ID.2, and ID.3 fields), size field (SIZE), packet type field (TYPE), and device vendor specific data (DEVINFO). The size of the Header block is 8 octets. This block is REQUIRED.

The DATA filed may contain measuring information, unit measure mark, accuracy information, it may represent List of Inquiring Identifiers, or some special information. The DATA field may be followed by TIMESTAMP and CHECKSUM.

A simple measuring device must produce packets which contains at least Header block and a simple form of the Data field.



3.1. Packet Leading Sequence

The first two octets of DTP/DIA packet contain values 0x49 and 0x54. These values must be used by software to detect the origin of DTP/DIA packet. Collector software must ignore any data not started with the specified packet leading sequence.

3.2. Version Field

Current version code is 0.

3.3. Flags Field

The 20th bit (L) determines the kind of byte order of multi-byte fields in the packet. 1 corresponds to little-endian format, 0 corresponds to big-endian format. It's up to IMS developer to choose byte order format. Obviously, it depends on what kind of microcontroller was used in the measuring device. For example, Intel's MCS 96/296 uses little-endian byte order, but Motorola's MC68xxx uses big-endian byte order.

The 21st bit (T) determines whether TIMESTAMP field has to be ignored or not (see Section 3.10). If this bit is set (T=1) TIMESTAMP must be ignored, otherwise it requires a valid value inside.

The 22nd bit (U) determines whether UTF-8 [10] encoding should be used for representation of textual information. If this bit is set (U=1) any text in the DATA field must be encoded with UTF-8, otherwise text is represented as common ASCII.

The 23rd bit is reserved for future use and must be 0.

3.4. Data Source Identifier

To distinguish data from various data sources DTP/DIA packet contains Data Source Identifier. To introduce a bit of structurization in distributed IMS the Identifier was divided into three parts: ID.1, ID.2, and ID.3. They correspond to high, middle, and low levels of 3-level hierarchy of IMS. The notation "AAA/BBB/CCC" will be used to specify Data Source Identifier (where AAA - ID.1, BBB - ID.2, CCC - ID.3).

It's not recommended to assign a fixed identifier for data source by device vendor. Developer of IMS should have an opportunity to change Data Source Identifier, for example, by means of special software or DIP-switches. Device vendor may apply Data Source Identification Procedure for assigning identifiers as described in Section 4.

Identifiers "0/0/0" and "255/255/255" are reserved for Data Source Identification Procedure (see Section 4).

Every retranslator SHOULD keep Data Source Identifier untouched.

3.5. Size of the Packet

The size of the packet specifies amount of 32-bit words occupied by the packet. The minimal size of DTP/DIA packet is 12 bytes (SIZE=3). The maximal size of DTP/DIA packet is 60 bytes (SIZE=15).

The values 0, 1, and 2 are illegal. Collector software should scan 7 bytes back for another packet leading sequence or discard received bytes.

3.6. Packet Types

The following packet types are declared: TYPE_FLOAT (TYPE=0), TYPE_INT1 (TYPE=1), TYPE_INT2 (TYPE=2), TYPE_INT3 (TYPE=3), TYPE_INFO (TYPE=14), TYPE_SPEC (TYPE=15). Other values are reserved for future use. Collector software should ignore the packets of reserved types.

TYPE_FLOAT, TYPE_INT1, TYPE_INT2, and TYPE_INT3 specify different data presentation forms (as described in Section 3.8).

TYPE_INFO packets are designed to provide additional textual information about data source. Octets from the 8th to the end of the packet (except for the last 32-bit word) are filled with some text (firmware version, copyrights, vendor information etc). Encoding of this text depends on flag U. Such text must be terminated by at least one octet zero and padding bytes must contain zeros as well. Maximal length of this text without trailing zeros is 47 bytes. Collector software may silently ignore such packets.

TYPE_SPEC packets have a special purpose in the Data Source Identification Procedure (see Section 4). When this packet is used in Data Source Identification Procedure, Measured Data block must be filled with zeros. In other cases Measured Data block may contain some device state information when the packet was issued by data source, or it may contain some control information when the packet was issued by collector. These values are vendor specific.

3.7. Device Vendor Specific Data

The DEVINFO field contains device vendor specific information. This value MUST NOT influence on interpretation of MEASURED DATA. It is recommended to code the model of the device in this field, so the whole Header block might be hard-coded in the firmware of the device.

3.8. DATA Field

The content of the DATA field is determined by the packet type. The size of this field must be devisible by 4. Padding bytes must contain zeroes.

TYPE_FLOAT packet (TYPE=0)

The DATA field contains a certain value of physical quantity, represented as 32-bit floating point number ("single" in terms of IEEE 754). These bits contain a sign bit (S), 8 bits of exponent

(E), and 23 bits of mantissa's fraction (M1...M23). Initial reference value is calculated in such a way (see details in [6]):

(-1)^S * 2^(E-127) * 1.M1M2M3...M23

Here we use "^" as exponentiating operator. This type supports the initial reference values in the range 1.18E-38... 3.40E+38 (in magnitude)

TYPE_INT1 packet (TYPE=1)

The DATA field contains multiplied by 10 value of physical quantity, represented as 32-bit signed integer number. This type supports the initial reference values in the range -214,748,364.8 ... 214,748,364.7

TYPE_INT2 packet (TYPE=2)

The DATA field contains multiplied by 100 value of physical quantity, represented as 32-bit signed integer number. This type supports the initial reference values in the range -21,474,836.48 ... 21,474,836.47

TYPE_INT3 packet (TYPE=3)

The DATA field contains multiplied by 1000 value of physical quantity, represented as 32-bit signed integer number. This type supports the initial reference values in the range -2,147,483.648 ... 2,147,483.647

Byte order of Measured Data block depends on the value of flag L.

Retranslator may convert measured data presentation from one form to another.

If information about measurement error and unit measure are to be transmitted the measuring information should be followed by unit measure mark and accuracy data.

UNIT MEASURE MARK	PROB	ERROR

UNIT MEASURE MARK starts from the 12th octet of the packet. This field is considered to be a text notation of unit measure. It is recommended to place generally used notations, based on [9], into this field. Encoding of this text depends on flag U. Such text must be terminated by at least one octet zero. The size of this field is variable but must be divisible by 4 and padding octets must contain zeros. Maximal length of this text without trailing zeros is 43 bytes. UNIT MEASURE MARK must not be used in TYPE_INFO and TYPE_SPEC packets. UNIT MEASURE MARK is required if accuracy fields are present (but may be filled with zeros).

Accuracy fields (PROB and ERROR) start just after the trailing zeros of UNIT MEASURE MARK. PROB offset must be aligned by 4. ERROR field contains the measurement relative error and follows PROB. PROB field contains the probability of physical quantity being outside of the interval determined by measurement relative error (the complement of the reliability to 1). In TYPE_FLOAT packets accuracy fields are represented as 32-bit floating point values and occupy 8 octets. In TYPE_INTx packets they are represented as 16-bit unsigned integer numbers multiplied by 10000 and occupy 4 octets. Byte order of these values depends on the value of flag L. Accuracy fields are OPTIONAL.

3.9. TIMESTAMP

To distinguish various packets and so to avoid packet duplication TIMESTAMP may be included into the packet. TIMESTAMP stands for measurement time, represented as 24 least significant bits in seconds, elapsed since 00:00:00 UTC, January 1, 1970. Byte order of TIMESTAMP is determined by bit L in the flags field.

If collector receives two packets from one data source with the same TIMESTAMP it should discard one of these packets. This situation may happen not only because of packets duplication but when developer of IMS has implemented the opportunity to correct measured data or to make it more accurate on-the-fly. So collector software may be set up to discard the first packet with the same TIMESTAMP.

This field is OPTIONAL. In the case TIMESTAMP and CHECKSUM are absent (SIZE=3), the bit T in the flags field must be set (T=1). If CHECKSUM is included into packet the space for TIMESTAMP is reserved to provide the proper alignment. In the case time stamp information isn't required the bit T must be set (T=1), TIMESTAMP must be filled with zero and will be ignored by collector. If bit T is cleared (T=0) TIMESTAMP contains a valid value and may be treated.

3.10. CHECKSUM

CHECKSUM occupies the last octet of the packet. CHECKSUM is the remainder of division of the sum of previous bytes by 256 (modulus of 256). This field is required if the packet is larger than 12 bytes (SIZE>3). The packet with incorrect CHECKSUM must be discarded.

4. Data Source Identification Procedure

Data Source Identification Procedure is an optional mechanism that helps sharing a single transport layer connection for data transmission from several data sources. This feature must not be applied to connectionless (message-oriented) channels. So data

source can distinguish every collector by a certain transport layer connection. This mechanism consists of two events: "Offer To Identify" and "Device Request".

"Offer To Identify" corresponds to TYPE_SPEC packet with ID="255/255/255", sent by data source equipment. This packet may be sent not only just right after connection establishment but at any moment (for example, when hardware configuration of data sources has changed). Expected reaction of collector is "Device Request" packet. If collector ignores "Offer To Identify" (doesn't reply for specified time) the action of data source must be one of the following:

- 0..3 retries of "Offer To Identify" then it must break the transport layer connection;

- 0..3 retries of "Offer To Identify" then it must start transmission of data packets of data source with the least existing identifier.

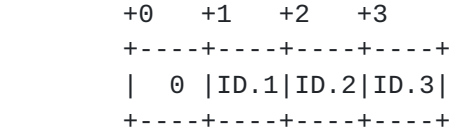
If collector requests both non-existing and existing data sources the equipment action must be one of the following:

- 0..3 retries of "Offer To Identify" then it must break the transport layer connection;

- 0..3 retries of "Offer To Identify" then it must ignore the non-existing data sources request;

If collector requests non-existing data source only the equipment may break the transport layer connection.

"Device Request" corresponds to TYPE_SPEC packet with ID="0/0/0", sent by collector. This packet must contain List of Inquiring Identifiers in the DATA field. The size of this list is variable but divisible by 4 (the size of sequence). The list starts from the 12th octet and lasts to the end of the packet. Each Data Source Identifier in this list occupies 3 last octets in each sequence. Leading byte in each sequence is zero.



List of Inquiring Identifiers may contain up to 11 identifiers of the data sources, expected to send their data through this communication channel. If collector needs to request more than 11 data sources it must distribute their identifiers to several "Device Request"

packets. If data source accepts "Device Request" it starts sending data packets. Collector may send "Device Request" packet at any time of communication (not only when it answers to "Offer To Identify").

The difference between stand-alone "Device Request" and the sequence "Offer To Identify" - "Device Request" is that every time data source equipment sends "Offer To Identify" packet it clears the stack of requested identifiers for the corresponding collector, but when data source receives the next "Device Request" it adds identifiers from this packet to current stack of requested identifiers.

Also Data Source Identification Procedure may be applied for assigning identifiers. In this case the first identifier requested by collector should be used by data source as its own Data Source Identifier.

Retranslator may act as collector or as data source in this Data Source Identification Procedure.

5. Protocol Number

IANA has assigned TCP and UDP port number 3489 for DTP/DIA.

6. Security Considerations

DTP/DIA is definitely insecure. To produce security based applications developer of IMS should provide authentication and data protection on the transport layer (by means of SSL [5] or TLS [4].

References

[1] ANSI, "Coded Character Set - 7-Bit American Standard Code for Information Interchange", ANSI X3.4-1986.

[2] Bradner S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, 1997.

[3] Bradner S., "The Internet Standards Process - Revision 3", BCP 9, RFC 2026, 1996.

[4] Dierks T., Allen C., "The TLS Protocol Version 1.0", RFC 2246, 1999.

[5] Frier A., Karlton P., Kocher P., "The SSL 3.0 Protocol", Netscape Communications Corp., 1996.

[6] IEEE, "IEEE Standard for Binary Floating-Point Arithmetics", IEEE 754-1985.

[7] Postel J., "Transmission Control Protocol", STD 7, RFC 793, 1981.

[8] Postel J., "User Datagram Protocol", STD 6, RFC 768, 1980.

[9] "Symbols Units and Nomenclature in Physics". Document IUPAP-25, 1987.

[10] Yergeau F., "UTF-8, a transformation format of ISO 10646", RFC2279, 1998.

Acknowledgements

The author gratefully acknowledges the supervision of A. Moschevikin, associate professor, PhD, and valuable advice of A. Korolkov.

Author's Address

Alexei V. Soloviev
Chair of IMS & Physical Electronics
Petrozavodsk State University
Lenin St. 33
185910 Petrozavodsk, Karelia
RUSSIA

Phone: +7-8142-711021
E-mail: avsolov @ lab127.karelia.ru

Full Copyright Statement

Appendix: Glossary of Acronyms

CAMAC	- Computer Automated Measurement And Control
EIA/RS	- Electronic Industries Association Recommended Standard
GPIB	- General Purpose Interface Bus
IANA	- Internet Assigned Numbers Authority
IEC/CEI	- International Electrotechnical Commission
IEEE	- Institute of Electrical and Electronics Engineers
IETF	- Internet Engineering Task Force
IMS	- Information Measurement System
ISO	- International Organization for Standardization
OSI/RM	- Open Systems Interconnection Reference Model
SSL	- Secure Sockets Layer
TCP	- Transmission Control Protocol
TLS	- Transport Layer Security
UDP	- User Datagram Protocol
UTC	- Universal Coordinated Time
VXI	- VME bus eXtension for Instruments