Construction of Distributed Sensors Network Based on DTP/DIA

Alexei A. KOROLKOV
Alexei P. MOSCHEVIKIN
Alexei V. SOLOVIEV
Department of IMS & PhE, Petrozavodsk State University
Petrozavodsk, Karelia, 185640, Russian Federation

ABSTRACT
Present paper describes one of the systems with distributed data acquisition. The presented online sensors network covers the North-west region of Russia. Sensors measure atmospheric parameters: temperature, humidity and pressure. Measured data are transferred through Internet channels to database server every 5 minutes and may be also stored locally on the side of data source. All the acquired information is represented on the web-server http://thermo.karelia.ru/eng/. The sensors net is functioning on Data Transfer Protocol for Distributed Information Acquisition (DTP/DIA), which was designed especially for these purposes. The main features of DTP/DIA are simple implementation, reliability, ability to resolve various sources of information, support of up to 16 millions unique devices. In this net DTP/DIA uses TCP/IP packet as envelope.

Keywords: distributed sensors net, humidity, microcontrollers, online temperature, pressure, TCP/IP.

INTRODUCTION
Distributed sensors network is a kind of information measurement system (IMS), which performs acquisition, processing, storing, and presentation of measured data and usually doesn't suppose control function.

Distributed sensors networks are widely used in scientific experiments, technological procedures and even in usual life. Typical examples of such networks are:

- scientific monitoring of atmospheric parameters in ecology related topics;
- monitoring and control of technological parameters in various conveyer manufacturing;
- network of meters for monitoring of power consumption;
- network of movement sensors as a part of security system in some office;
- temperature control in large hotels etc.

The simplest IMS consists of measuring device, connected to computer by means of some communication interface. Typical measuring device contains a sensor and a microcontroller, which performs initial data acquisition and processing. In such a system the majority of IMS functions are given to computer.

Projects with large amount of investigated objects far distanced from each other require installing distributed IMS. Some nodes of distributed IMS should perform data transmission to nodes which process and store data. That's why in this case developer of distributed IMS should pay great attention to choice of communication interfaces and protocols. There are few vendor-specific, application-specific and general-purpose protocols that can be used in distributed IMS. The system with several objects may be based on some instrument interfaces like IEEE 488 (GPIB) or even EIA/RS 232. In this case SCPI or TMSL are commonly used as application-level protocol. More complicated systems are usually developed on the basis of CAMAC or VXI (for compact systems) and on the basis of CAN or IEC 60870-5 (for distributed systems).

![Figure 1. Typical Distributed Sensors Network](image-url)
SCPI, DNP, UCA, ELCOM-90 or some vendor-specific protocols are not suitable without adaptation. The network described in this paper matches this case. After analysis of existing protocols, their adaptation and simplification we developed Data Transfer Protocol for Distributed Information Acquisition (DTP/DIA), which suits our purpose perfectly.

In description of DTP/DIA we use the term "data source" for the object, which transmits measured data, and "data collector" - for the node that receives measured data. If the node performs both reception and retransmission, this type of nodes is named "retranslator". Obviously, in this terminology measuring devices are "data sources", but a certain computer could be either "end-point data collector" or "retranslator" (Fig.1).

DTP/DIA was implemented in both embedded firmware (for example, for Atmel microcontrollers) and personal computer software (Distributed Information Acquisition software – DIA).

**MAIN FEATURES OF THE DTP/DIA**

The protocol describes data transmission in small standalone packets. This feature allows using DTP/DIA over both streamed (connection-oriented) and message-oriented channels. DTP/DIA packet consists of three logical blocks: Header, Measured Data and Special Data. Header and Measured Data are required. They have a fixed size. Special Data block is recommended to use and its size is variable. The maximal size of DTP/DIA packet is 60 bytes. A simple measuring device must produce packets, contained at least Header and one of the forms of Measured Data block. More sophisticated device should include Special Data block as well.

DTP/DIA provides few data presentation forms. Each form corresponds to a specific packet type. Some of the forms help to avoid the implementation of floating point arithmetic in device firmware.

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

In distributed IMS it is very important to distinguish 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). To distinguish data from various data sources DTP/DIA packet contains Data Source Identifier. To introduce a feature of structurization in distributed IMS the Identifier was divided into two parts. They correspond to high and low levels of 2-level hierarchy. Developer of IMS may use the first part to specify the group of units in distributed IMS and the second part to appoint a device number in this group. The protocol supports up to 16 millions ($2^{24}$) data sources.

Another important part of the DTP/DIA identification mechanism is an optional feature called the **Data Source Identification Procedure**, which helps sharing a single transport layer connection for data transmission from several data sources. This feature is not supposed to 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". They represent a kind of handshaking. **Also Data Source Identification Procedure** may be applied for dynamic assigning identifiers.

DTP/DIA may be applied for both network and local channels. Few types of local channels 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.

DTP/DIA provides neither authentication mechanism, nor security issues. To produce security-based applications developer of IMS should provide authentication and data protection on the transport layer (by means of SSL [2] or TLS [1]).

Detailed DTP/DIA specification can be found in Drafts Directory of IETF: www.ietf.org/ietf/1id-abstracts.txt

**DIA SOFTWARE**

Distributed Information Acquisition Software fully implements the described protocol. It was developed and tested on several platforms: Windows 95/98/Me, Windows NT/2000/XP, Linux and FreeBSD. Linux version can be linked statically and placed with the Linux kernel on a floppy disk. So it can be used in diskless low-performance systems (floppy required only). The main part of the program is provided with open sources under the terms of the GNU Lesser General Public License (http://www.gnu.org).

The program deals with objects of two kinds: **Device** and **Channel**. **Device** objects represent a particular data source and store current measured information (date of measurement, measurement result, accuracy, units etc). **Channel** objects hides the method of data receiving and transmitting from other parts of the program. Such object implements input/output procedures for a specific communication resource (serial port, TCP-socket, UDP-socket, file, database, etc). Although it is not limited by
the protocol, the current release of the DIA software doesn't support bidirectional Channel objects, i.e. any Channel object must either transmit data or receive it. Each Device object may be connected with few transmitting or receiving channels. Transmitting channels implements the “retranslator” function in the terms of DTP/DIA. Moreover the data storing function is the transmitting function as well. That is when we suppose to store measuring information in a file or database we assign a certain transmitting channel to corresponding object, which “transmits” data to file or database.

The open-source interface of objects interconnection and the modular structure of the program allow developing a module for any communication resource. The program can use new modules without recompiling. At the moment we have implemented the following communication modules:

- RS232 module performs communication via serial line.
- FILE module allows receiving data from file. It is supposed that new data arrives when file modification time is changed. This module can be flexibly configured to support different file formats.
- MYSQL module provides the opportunity to store data in MySQL database.
- TCP/UDP module supports receiving and transmitting data over Internet/Intranet channels ([5], [6]). When the program transmits data by means of this module it acts as retranslator in the terms of DTP/DIA. This module can be configured to use SOCKS5 proxy [3].
- SSL module performs secure receiving and transmitting measuring information. It provides some kind of authentication. At the stage of SSL handshaking both communicating sides should provide certificates, which are signed by trusted certification authority. Otherwise the connection establishment won't be authorized and SSL handshaking will fail.
- CTRL module implements some control interface, which allows creation and removing of both kinds of objects in the run-time. This feature provides the opportunity to configure the distributed IMS dynamically. By means of this module the operator may watch the state of all the objects in the running program.
- SMTP module implements a simple SMTP client according to [4]. It is supposed to inform the operator about long network time-outs. By means of various SMTP gates this information may be delivered to SMS, pager or ICQ.

DTP/DIA IN MICROCONTROLLER FIRMWARE

The realization of DTP/DIA in microcontroller built-in software is rather unsophisticated due to simplicity of the protocol. Binary code of packet formation and transmission procedure does not exceed 100 bytes in case of hardware UART in MCU (microcontroller unit).

One of the examples of DTP/DIA implementation is temperature meter, based on Atmel MCU AT90S2313, which has the following useful features:

- 2K bytes of in-system programmable flash memory;
- 128 bytes of SRAM;
- programmable watchdog timer;
- full duplex UART;
- low power consumption.

Block scheme of device is presented in Fig.2.

Figure 2. Block Scheme of Digital Temperature Meter

Digital sensor DS1820 (Dallas Semiconductors) is connected to MCU through 1-wire interface in slave mode. MCU puts various commands on data line and sensor replies. The whole cycle of data acquisition includes: sensor reset, temperature registering procedure, another sensor reset, temperature read procedure, processing read data, transmitting the data and sleep period (aimed at power saving). The stage of processing is necessary when digital format of obtained data does not comply with the certain field specification in DTP/DIA. For example, the most common format of DTP/DIA packet assumes the transmitting real physical value multiplied by 10 to escape fractional part (usually, it is sufficient to know value with accuracy of ±0.1).

Each DS1820 has unique 64-bit hardware number. It partially forms Data Source Identifier field in DTP/DIA packets. Due to this feature few sensors may be connected on the 1-wire interface to one MCU. Microcontroller distinguishes various sources, so any distributed network of sensors may be constructed both with use of local and network channels.

ONLINE SENSORS NET

On the basis of described protocol the online net of air temperature, pressure and humidity sensors was created (web version see at http://thermo.karelia.ru/eng/). It
covers the territory of North-west Russia. Also one of the sensors is distanced at several thousand kilometers from Petrozavodsk and is situated in Mezhdurechensk (South Siberia). Geographical scheme of Karelian region of net is presented in Fig.3.

Three letter abbreviations on the map stand for different towns in Karelia: PAA – Paanayarvi, LOU – Louhi, SEG – Segezha, PTZ – Petrozavodsk, SOR – Sortavala. Arrows near the names of towns designate the current rise or fall of air temperature in respect to previous hour value.

All sensors yield data every 5 minutes and transmit them through Internet channels to the collector server situated in Petrozavodsk. Graphs (3-day, 3-week, 3-month and 1-year periods) with time dependence of air temperature are renewed every half hour. Collector, database and web-server software are running on one computer. The main difference between various sensors is the way of data transfer and physical connection of sensors to data transfer devices.

Almost all temperature sensors are connected to network in various ways. LOU sensor is connected to COM port of computer running Windows NT, SEG – Windows 98. Both are hidden behind firewall. Sortavala's one is connected to router's COM port, which is mapped to certain TCP port (feature of Cisco Systems hardware). All computers except PTZ and Mezhdurechensk (MER) sensors have Internet accessible IP addresses. PTZ and MER have local IP addresses and are translated by NATs. Moreover, the first PTZ sensor was constructed not on DS1820, but on MAX6577, connected to LPT port of computer. So data transmission is carried out through FILE channel. PTZ computer runs Linux, and MER computer runs FreeBSD (see Fig.4).

Collector software stores temperature data from various towns in corresponding log files. Web-server software and additional scripts build graphs, banners, and other required for certain web-design objects in HTML-pages.

In Petrozavodsk segment of the sensors network the ability of DIA SSL module to provide authentication and secure data transmission is being tested. SSL [2] supports peer's identity authentication by means of asymmetric (public keys) cryptography. On the basis of OpenSSL library we established private certification authority (CA) and issued X.509 certificates [7] for nodes of the network. Each node have own certificated signed by our CA and local copy of CA's self-signed certificate. During SSL-handshaking a node presents its own certificate to peer and requests peer's certificate. The node and the peer check that presented certificates were signed by our CA certificate unless connection will be broken. If the node and the peer were successfully authenticated further transmitted data are enciphered by some symmetric crypto algorithm.

In Petrozavodsk segment there are meters of not only temperature, but also of humidity and pressure. Due to universality of DTP/DIA the same network channels are used for transmission of various types of physical quantities. Any packet may include unit measure, confidence interval and accuracy. It is up to collector

![Figure 3. Karelian Map with Temperature](http://www.rikt.ru)

![Figure 4. Structure of the Presented Distributed Sensors Network](http://thermo.karelia.ru)
software to determine the source and physical quantity sent.

ACKNOWLEDGMENT

We'd like to express gratitude to all our colleagues at the department of information measurement systems and physical electronics of Petrozavodsk State University for support in this project.

The research described in this publication was made possible in part by Award No. PZ-013-02 of the U.S. Civilian Research & Development Foundation for the Independent States of the Former Soviet Union (CRDF) and of Ministry of Education of Russian Federation.

GLOSSARY OF ACRONYMS

CAMAC Computer Automated Measurement And Control
DNP Distributed Network Protocol
EIA/RS Electronic Industries Association Recommended Standard
FDDI Fiber Distributed Data Interface
GPIB General Purpose Interface Bus
HTML HyperText Mark-up Language
IANA Internet Assigned Numbers Authority
IEC International Electrotechnical Commission
IEEE Institute of Electrical and Electronics Engineers
IETF Internet Engineering Task Force
IMS Information Measurement System
NAT Network Address Translation
OSI/RM Open Systems Interconnection Reference Model
PPP Point-to-Point Protocol
SCPI Standard Commands for Programmable Instruments
SMTP Simple Mail Transfer Protocol
SSL Secure Sockets Layer
TCP Transmission Control Protocol
TLS Transport Layer Security
TMSL Test & Measurement Systems Language
UART Universal Asynchronous Receiver and Transmitter
UCA Utility Communication Architecture
UDP User Datagram Protocol
VXI VME bus eXtension for Instruments

REFERENCES