

Analysis and structure of a system based on IFSF international communication standard for measuring the consumption of petroleum products. IFSF and TCP / IP protocol

Ekaterina Antonova Gospodinova-Zaharieva

Abstract—In 1993, retailers of petroleum products began introducing computer systems at service stations to deal with integrated pump control, payment cards, epos (electronic point of sale) terminals, etc.. International standards IFSF the U.S., Europe and around the world reap significant cost savings by implementing network devices based on open standards systems. New devices, such as car ID, may make use of existing standards IFSF. Simple IFSF design and engineering means less testing. Unified system architecture simplifies applications and interfaces. Evolution of standards IFSF avoids sudden aging. IFSF is completely compatible with TCP/IP. By Open Systems-based technologies restrictions are avoided in any particular platform or equipment manufacturer. This enables reducing to a minimum the cost of installation and maintenance. In the absence of an approach based on common standards, petrol retailers may choose to use equipment and systems from one or multiple vendors. Both approaches have serious shortcomings. Since the hierarchical architecture is not the optimal control solution due to its complexity, this work explores a computer system designed to sell liquid fuels, gas and other goods at gas and petrol stations, a system based on the international IFSF standard, and which aims to improve the data transmission speed between the different network platforms. In many cases, these platforms are not synchronized, not interconnected, and do not support unified communication standards. It is necessary to build an open, interoperable system connecting different devices into a single network, so that they can exchange data and instructions. Therefore, a peer-to-peer network has been used where the network controller could be eliminated without loss of functionality. An open standard has been provided, and also free choice for add-ons` modifications, implementation of new features, and maintenance.

Index Terms— IFSF, TCP/IP, software, interface, converter module.

I. INTRODUCTION

Every day thousands of transactions are made at the gas stations. Data from them is processed and transmitted. The IFSF International Standard is based on a LonWorks network platform and open standard systems, and is TCP / IP-compliant. Its integration options are with high transmission capacity. It is widely accepted that today's most common communication technology is TCP / IP. IFSF publishes an alternative communication IFSF Layer Protocol

Ekaterina Antonova Gospodinova-Zaharieva, Department of Electrical engineering, automation and information technology, Technical University of Sofia - Faculty of Engineering and Pedagogy-Sliven, Sliven 8800, Bulgaria.

based on TCP / IP. IT features of TCP / IP are used to ensure that IFSF applications also operate with TCP / IP. The protocol is based on an open system architecture, so all devices can communicate with each other. In this way, it is not necessary for every device of different origins to get acquainted with the technical characteristics of the other devices. The researched computer system designed to sell liquid fuels, gas and other products at gas and petrol stations is based on the international IFSF standard, personal computers, an IFSF device with a TCP / IP interface and a converter module.

It is widely accepted that today's most common communication technology is TCP / IP. Using the Lon Works network platform and the IFSF protocol, the following goals are met:

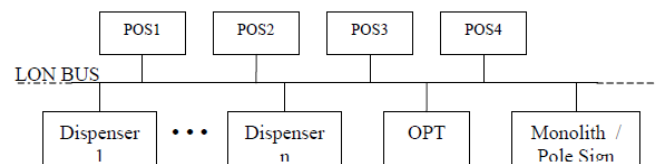
- means used to measure are synchronized;
- a unified communication standard is maintained;
- the cost for the consumers is reduced.

II. SYSTEM ARCHITECTURE OF INTERNATIONAL COMMUNICATION STANDARD IFSF

The protocol is designed to be able to support different configurations:

- a system using one controller that manages various

Figure 1 Architecture with one controller



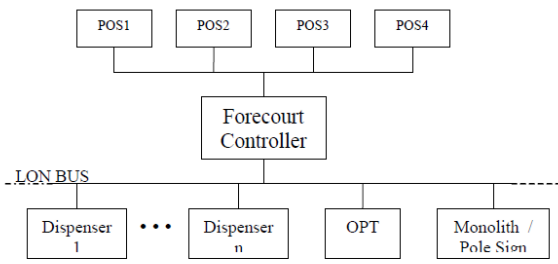
communications and is able to store distributive transactions if multiple points of use are present. It is responsible for the data distribution between them (Figure 1);

- a decentralized system in which the devices are connected directly to the bus (Figure 2);

The choice between these two solutions depends on the following rules: buffering capability in the dispenser or in the controller, or migration capability. Most existing systems use the first architecture due to its easier application. Message exchange can be described as following:

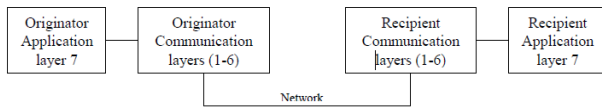
- messages between the layers: Originator Application and Originator Communication;

Analysis and structure of a system based on IFSF international communication standard for measuring the consumption of petroleum products. IFSF and TCP / IP protocol



- network messages;
- messages between recipient's communication layers and recipient application.

Figure 2 Decentralized system



From compatibility perspective, only the network message should be considered, but for clarity the relationship between application layer 7 and message layers 1-6 (Figure 3) is also taken into consideration.

Figure. 3 Layers from 1 to 7

III. COMPONENTS

The selected design components ensure that the ISO communication layers from 1 to 6 are able to use different physical layers: RS485, power lines, fiber optics, radio waves, infrared light and more. The top level of the network hierarchy is the domain. If different applications share messages, they can be divided into different areas. All devices in a gas station are installed in domain 1. Second-level addressing is a subnet. A subnet is a logical grouping of nodes from one or more channels. There may be up to 255 domain subnets. The third level of addressing is the node. There may be up to 127 nodes on the subnet. Group addressing of facilities is not used.

To avoid specifying the contents of messages when the application is dependent, it is designed to use a read / write mechanism. Six main messages were used to access all data in the system: Read Messages, Answer Messages, Write Messages, Unsolicited Data Messages with Acknowledge, Unsolicited Data Messages without Acknowledge, Acknowledge Messages.

IV. ADDRESSING

The platform is built in such a way that each device can be localized and define its own address, meaning: subnet and number of nodes (S, N) by:

- built-in switches or similar mechanical solution;
- an address provided by the x-processor application;
- subnet (S), number of nodes (N): 2 bytes.

The subnet address (S) is used to indicate the device type.

• S: Distributors, controllers, printers, card readers, card access, bank access, poles, tank level meter, carwash, monitoring equipment, fuel supply control, payment terminals, network server, user interface, security system, etc.

• N: All devices belonging to the same subnet are identified

by a unique node address (N). The values are 1-127.

S and N must be applied to the application layer. Therefore, the incoming message reaches only to the application layer, and then the data is processed for the particular device.

V. IFSF AND TCP / IP PROTOCOL

The distribution of the transport of IFSF messages using a TCP / IP protocol of any size, is done by using the distribution of other embedded applications and is limited to a maximum of 228 bytes to prevent the buffer from being overflowed. The selected IP packet does not buffer small messages to avoid delay. Also, changes are not made on the IP addresses in the serving flag, because they cannot be managed by different network routers and IP stacks. When IP is a streaming protocol, a message may not start immediately after the end of another. The RFC discovers its start and end in order to correctly execute the IFSF request. Applying additional information to an IFSF request to transfer one message to another is not allowed. The length field is used. It is in the same position for all IFSF messages. In positioning in recovery mode, the sending node retransmits messages in their entirety. Appropriate access control are passwords, dial-up backups, firewalls, etc. that ensure network security. The use of an authentication mechanism is recommended, because sensitive network messages are routed to unsecured connections, and this ensures that the message code is real at the end of the link.

VI. IFSF DEVICE ARCHITECTURE WITH TCP / IP INTERFACE

It contains four main components:

- IFSF application that will remain the same whether the communication is LON or TCP / IP.
- IP stack is the network interface. It implements the various IP protocols and provides a connection management service separating IP addresses from other protocol stacks.
- DHCP server that is used to distribute IP addresses to devices on a single network. It may be part of an IFSF device, or a separate device.
- IFSF to IP converter module. It has the task of looking like an IFSF interface to the current IFSF application, accepting all IFSF messages, placing them in IP datagrams, and sending them to remote devices over a LAN. The module has three main objectives: to send and receive proxy messages, maintain a list of all active local network connections and packets of all data and control messages in TCP feeds for the network module.

VII. CONVERTER MODULE

The converter module consists of three function blocks (Fig. 4):

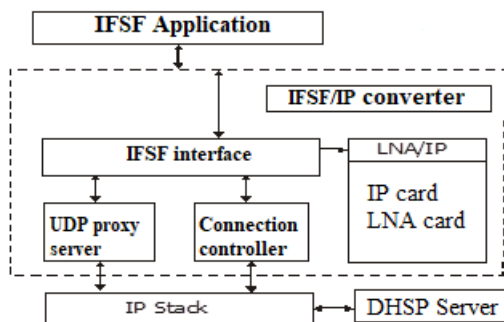
- IFSF interface. This module is responsible for the interaction between the application of IFSF and other IP communication services. It supports a table with all LNAs and their respective IP addresses (a combination of IP address, protocol, and port.) This module will route all periodic messages to a proxy server and all other messages to the link administrator;
- UDP. This module is responsible for partitioning the local periodic request queries and broadcasts them using UDP datagrams. Incoming periodic messages come via this module and are sent via the IFSF interface.

- Communication controller. This block is responsible for managing TCP connections. Each IFSF message is processed by this interface. All data and control messages will be wrapped in TCP and sent to the appropriate address. The appropriate address is determined by taking the LNA information from the IFSF message and found at the corresponding LNA and IP mapping address. The receiving station will accept the message and download the TCP shell. When an IFSF message is sent the request from this interface sends an IFSF message, and the controller checks if there is a TCP connection to the required IFSF device. If this is not the case, a connection setup request is generated and will be sent to the IFSF network application address.

Figure 4 Converter module

VIII. SEQUENCE OF THE IFSF AND TCP / IP COMMUNICATION

Before each IFSF communication starts, the DHCP server must be configured with its own IP address and the IP address range must be made available to clients. All other devices must have their nodes at installation in the LNA address. When communication begins, each TCP / IP stack will request an IP



address from the DHCP server using UDP. The proxy server will create a "well known" port in the IP stack so that incoming messages can be received. Each application will periodically send a message to the IFSF interface. Upon receiving, it will register the application, use the IP stack to create a contact address with the TCP interface, and will enter its LNA / IP card. For each IFSF request, a unique port address will be allocated. When this is complete for all network IFSF applications, the table will have an identifier for socket addresses. The periodic message will be broadcast via UDP to the "well known" gateway containing the contact addresses for this request. Thus, all configuration tasks have been completed to allow all network IFSF applications to receive and send messages. In order to send messages, the IFSF will route the periodic messages from any application that is a proxy network node. These messages will be broadcast using UDP datagrams, and all incoming messages will be viewed and recorded. They will then be accepted by all IFSF applications sent from this interface. For explicit messages, when such IFSF message is received from an application, the connection administrator will verify that the connection to the required device has been created and, if so, will send the associated socket message. If not, it will receive an address from the LNA / IP table and will send the message.

IX. NETWORKING AND ACCESS AUTHORIZATION

To connect an IFSF device to a computer on or off the workspace, permission is required to resolve its name. This is accomplished through a network node file or DNS, or both. In a strict NT environment, WINS is used, but WINS is limited to NT machines only. In an environment where different types of operating systems are used, DNS and network file are the only compatibility options. To use a network file, the network node name to the IP address card must be manually placed in the receiving file. The change in a network environment must be reflected in each network file on each machine. For small networks, this is feasible, but the bigger the network, the harder it is. DNS is the solution for this medium-sized network.

X. INPUT / OUTPUT SYSTEM

The device has a card reader controlled by another two-controller device. The distributor has two independent IFSF applications, one control distributor from another controlled reader card. The link controller opens a TCP port to receive connection requests for each local IFSF device. The recordings are made in a local LNA / IP table for each local application. The IFSF request of the device sends a message to the controller. If the converter detects that there is no TCP connection between the controller and the distributor application, it instructs another controller to send a TCP message with a connection request.

XI. CONCLUSION

Hierarchical architecture is not the optimal control solution. It is unnecessarily complicated. Therefore, a peer-to-peer network is used where the network controller can be eliminated without loss of functionality. Since the structure does not consist of multiple control layers, the devices are directly connected to the individual channels. This architecture does not limit the flow of information between the devices, executes management algorithms, and increases the utility of the system. The assembly time is also reduced. The network management software coordinates the transfer of information. In this system, the communication controller is intended to optimize and simplify communications with respect to the different protocols or interfaces used by the data transmission system manufacturers. The system is simplified. With one device, the use of two different protocol messaging architectures containing a TCP / IP module and address table is allowed. Only a small part of the system resources is consumed, an unlimited number of devices can be connected to the network, it is designed to meet the requirements, it is OSI-compliant and has its own protocol for adopting Layer 7, as well as a technical implementation for Layers 1 and 2 through 6. All existing standards are adopted or adapted. The system architecture of the controller allows all devices to communicate with each other.

ACKNOWLEDGMENT

Thanks to eng. Dimitar Bozhilov and ELL LTD for their cooperation and invaluable help!

REFERENCES

- [1] ISO/IEC 12207:1995, Information Technology – Software life cycle processes, 2015
- [2] IEEE Standard for Software Test Documentation, IEEE Std. 829 – 1998, 2013
- [3] Dorf Chard C., Modern Control System,2010.
- [4] Keating D.A., Sensors and Interfacing.
- [5] Technical documentation, LonMark International, 217
- [6] Schneider Automation, Modbus Messaging on TCP/IP Implementation Guide, 2010
- [7] Erickson, Kelvin, and Hedrick, John, Plant Wide Process Control, Wiley & Sons , 2012
- [8] <http://www.nsys-bg.com>
- [9] www.isfs.org
- [10]http://www.stitcs.com/en/LonWorks/IFSF_Management_Overview
- [11] www.conferencenet.org
- [12] www.bloomberg.com
- [13] <ftp://ftp.prosoft.ru/pub/Hardware/belden/Catalogs/LonWorks>
- [14] [www. LonMark International](http://www.LonMarkInternational.com)
- [15] www.echelon.com/support