Engineering and Technology Research 5(1): 013-017, January 2021
DOI: 10.15413/etr.2021.0114

ISSN: 2682-5716

Research Paper

Analysis and structure of the international communication standard IFSF: IFSF device architecture design with TCP/IP interface

Accepted 7th December, 2021

ABSTRACT

The oil industry has always had problems in its communications in regard to the various protocols or interfaces used by intelligent system manufacturers. Patented protocols limit the choice to certain producers, who often cannot meet the changing needs of oil companies and computer systems. It is often necessary to convert protocols, thus increasing equipment prices and maintenance costs, and delaying projects. IFSF (International Forecourt Standards Forum) is designed to meet the requirements of the OSI (Open Systems Interconnection Basic Reference Model) model, which has its own protocol for working with layer 7 and technical common implementation for layers 1 to 6. All existing standards have been adopted or adapted. The protocol is based on an open system architecture so that the devices can communicate with each other. In this way, it is not necessary for each device of different origin to be acquainted with the technical characteristics of the other devices. The communication layer is specified by IFSF separately for LonWorks and TCP / IP. Both options use established IT technologies, with some IFSF implementation recommendations.

Key words: Communication standard, interface, protocol, intelligent systems, protocol standard, TSP\IP, IFSF, oil.

Ekaterina Antonova Gospodinova

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

E-mail: ekaterina_gospodiniova@abv.bg.

INTRODUCTION

The international IFSF standard was established in 1993, when the oil industry began to introduce computer systems into gas station architecture to integrate control devices, card payment systems and other IT applications (eg. distribution, accounting and sales statistics). It was created by a European organization, including international oil companies, which supports the development and use of international standards in the oil industry for the interoperability of gas station systems and devices (Willig, 2013; Wollschlaeger, 2014). The oil industry has always had problems in its communications in regard to the various protocols or interfaces used by system manufacturers. Patented protocols limit the choice to certain producers, who often cannot meet the changing needs of oil companies and computer systems (NSYS and FAFNIR, 2020).

Existing devices can commission IFSF using a protocol converter or PCD, a small computer that accepts the IFSF protocol and communicates with other devices using its proprietary protocol. The IFSF has two independent layers: a device request protocol layer and a communication protocol layer. The request protocol is independent of the communication layer. The specification of the communication protocol makes the connection to the transport layer (Technologies and Communications, 2011).

IFSF standards specify messages sent and received from each device type. Messages are designed to be extensible by defining individual fields by type and length. The messages are grouped in different databases for the logical part of each device. Each type of device (dispenser, tank level meter, etc.) defines its own set of databases and fields

(Schneider Automation, 2019).

Solving the existing problems will significantly simplify the design, reliability, installation and operation of level measuring systems and will improve the operation of control systems and fuel level. The aim is to design an architecture of an IFSF device with a TCP / IP interface for connection to a network technology platform for an automated leveling system and the ISFS standard (Waft, 2019).

COMPONENTS

The components selected to provide ISO (International Organization for Standardization) communication layers 1 to 6 use different physical layers: RS485, power lines, optical fibers, radio waves, infrared light, etc. The highest 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 a domain - level one. The second level of addressing is the subnet. A subnet is a logical grouping of nodes from one or more channels for transmitting messages. There may be up to 255 subnets per domain. The third level of addressing is the node. There may be up to 127 subnet nodes. Group addressing of facilities is not used (Volz, 2018).

The protocol offers four main communication services: ACKD, REQUEST / RESPONSE, UNACKD_RPT, UNACKD.

The IFSF network built-in service is ACKD. It sends a message to all nodes, and individual confirmations from each receiving node are expected to be returned. If the confirmations are not received, the transaction is performed again. The number of attempts and the waiting time are variable and are determined together with the network parameters by the application level (layer 7 of the OSI model). The parameters and their values are determined by the IFSF as follows: the priority layer first determines whether the message has priority over other messages and in this way improves the response time of critical messages. After the first attempt, it is determined how many times the node will repeat without receiving confirmation. The physical layer 1 is in free topology at 78 Kbits. The protocol uses P-resistant CSMA (Carrier-sense multiple access-like Ethernet) and thus does not require other bus management. The CSMA embedded chip uses three processors and because only two of them are used to control communications in the bus, the later never interferes with the transfer of messages to the application layer. Explicit messages are expected for layers 1 to 6. The duration of such messages is determined by the Max_Block_Length data element. The block length range is 32 to 228, and BL (Block number) is included in confirmation messages (Schneider and LonMark, 2012, 2020).

IFSF DEVICE ARCHITECTURE WITH TCP / IP INTERFACE

There are four main components:

• IFSF application that will remain the same whether the communication is LON or TCP/IP;
• The IP stack is the interface to the network. It implements the various IP protocols and provides a connection management service that separates IP addresses from other protocol stacks;
• DHCP server, which is used to distribute IP addresses to devices on a network. It may be part of an IFSF device, or it may be a separate device;
• IFSF to IP converter module. It has the responsibility to look like an IFSF interface to the current IFSF application, receive all IFSF messages, place them in IP datagrams, and send them to remote devices over a local network. The module has three main purposes - to send and receive messages through a proxy, to maintain a list of all active connections in the local network and packets of all data and control messages in TCP flows for the network module.

SEQUENCE OF IFSF AND TCP / IP COMMUNICATION

Before each IFSF communication can begin, the DHCP server must be configured with its own IP address and range of IP addresses to be accessible to clients. All other devices must have their node numbers when installed in the LNA (A low-noise amplifier) address. When communication begins, each TCP/IP stack will request an IP address from the DHCP server using UDP. Some devices may use static addresses. If so, each device must be able to be reprogrammed on the network as required. It is recommended that static IP addressing not be used. The proxy server will create a "well known" port in the IP stack so that incoming messages can be received. Each application will send a periodic message to the IFSF interface. Upon receipt, it will register the application, use the IP stack to obtain a contact address with the TCP interface and enter its LNA/IP card. Unique port addresses will be allocated for each IFSF request. Thus, all configuration tasks are completed to enable all hosted IFSF applications to receive and send messages.

To send messages, the IFSF interface will route periodic messages from each application that hosts the proxy server. These messages will be transmitted using UDP datagrams, and all incoming will be viewed and recorded where necessary.

IP SERVICES

Internet Protocol (IP) has several types of services that need to be routed to receive data, although these services only apply to frameworks that pass through a router to reach their destination. The ARP protocol is the mechanism that distributes IP addresses. It is not used for IFSF message transfer, so there is no setup option. The ICMP protocol reports errors to diagnose network problems, but it is also not used to transfer IFSF messages. The TCP protocol must be set to a maximum size that matches the size of the buffer for receiving messages on the local host. The local host buffer must be able to handle multiple messages in order to use the environment. A good built-in mechanism can save data without overflowing the buffer, causing retransmissions. This will also avoid resending the link segments that are so short that they lead to inefficient bandwidth usage. The UDP protocol does not yet have messaging options to be configured. DHCP also has no messaging options, but there is optional information that can be provided to the customer when applying for an IP address. DNS server address and TFTP are not options.

NETWORKING AND GRANTING ACCESS

The method for sending messages between IFSF devices is solved and described. To connect a device to an IFSF computer for on-site or off-site communications requires some kind of name resolution mechanism. This is usually accomplished through a host 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 host file are the only compatibility options. To use a host file, the host name to the IP address card must be manually placed in the receiving file. The change in the network environment must be reflected in each host file on each machine. For small networks, this is an easy task, but the larger the network, the harder it becomes. DNS is the solution for medium and large networks. It is difficult to set up a DNS server when there is only one point in place that needs to be applied to each device on the network.

RESEARCH AND IMPLEMENTATION OF THE SYSTEM

This configuration uses a dispenser with a card reader controlled by a dual-control device. The allocator has two independent IFSF applications, control allocators from a controlled card reader. The IFSF protocol converter is shown and bound by the innermost dotted line. This is a collection of applications that are responsible for implementing IFSF interaction with the protocol stack.

Figure 1 shows the moment after switching on the two controllers. The proxy server opens a UDP port for sending and receiving remote periodic messages. Connection controllers open a TCP port for receiving connection requests for each local IFSF device. Entries are made in a local LNA / IP table for each local application.

The dispenser sends an IFSF request message to controller1. If the converter module determines that there is no TCP connection between controller1 and the allocator application, it instructs the connection controller to send a TCP connection request message in the following steps:

1. source - IP = 192.1.1.1
2. track - IP = 192.1.1.21
3. source Port = 1111
4. track port = from LNA to get IP table → TCP
5. the message acknowledges the connection and identifies the controller port
6. the converter module adds this port address to the correct LNA / IP table and sends a TCP message containing an IFSF message.
7. source - IP = 192.1.1.1
8. track - IP = 192.1.1.21
9. source Port = 1111
10. source - Port = 2222
11. communication of the IFSF application.

TCP/IP CONNECTION ARCHITECTURE

The implementation allows the use of two different TCP protocol architectures (used to transfer IFSF messages), representing an oriented connection. The two possible TCP connection architectures created between IFSF devices mainly contain a TCP/IP module and an address table (Figure 2). The table is used to translate the outgoing IFSF TCP connection message. The structure of the address table varies depending on the architecture. When using TCP protocol there is only one connection between two hosts, regardless of the number of IFSF devices located on each host. Therefore, the address table needs to be more complex. In fact, the address table structure is a relational database consisting of two tables:

- IFSF table for devices;
- IFSF/TCP connection table.

The table for TCP connections contains:

1. IP address of the partner host;
2. Port - contains the port number of the application that runs on the partner host;
3. TCP SOCKET FD - contains a file descriptor of the local TCP connection - this is a reference to a TCP connection that should be used to transfer IFSF messages for a specific device;
4. IFSF address - contains the IFSF address of a special IFSF receiving device.

Multiple TCP connections between two hosts have a structured address table, and this architecture is quite

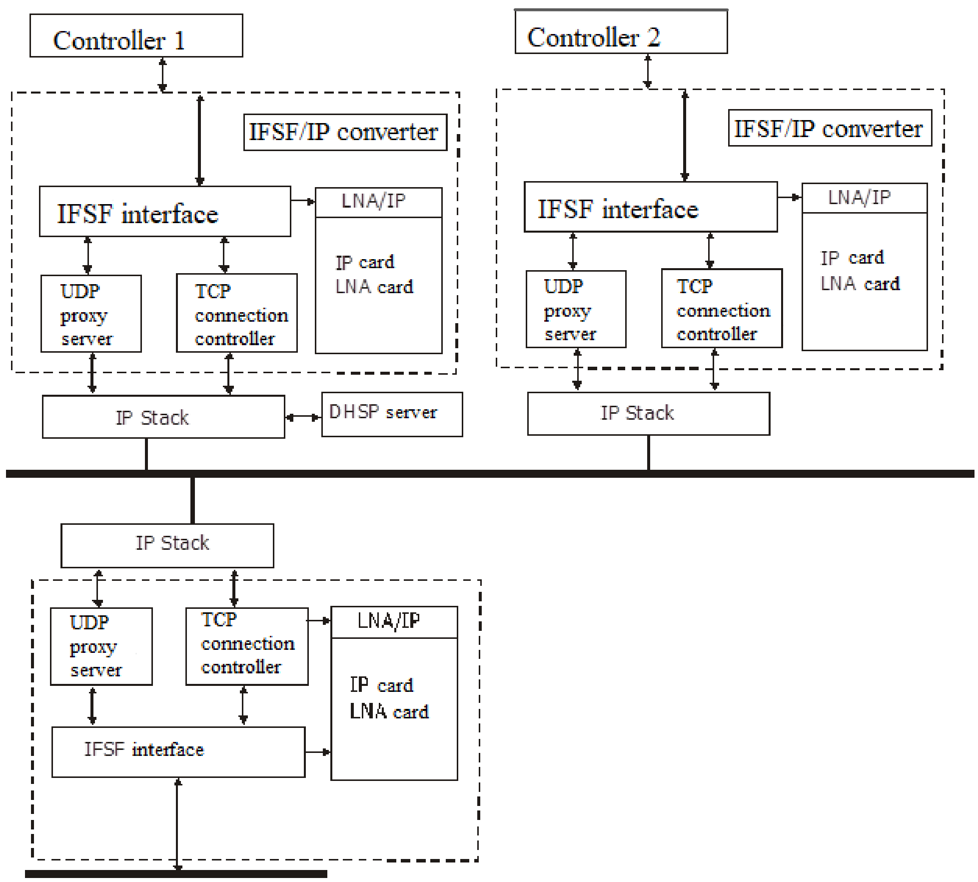

Figure 1: Moment after switching on the two controllers.

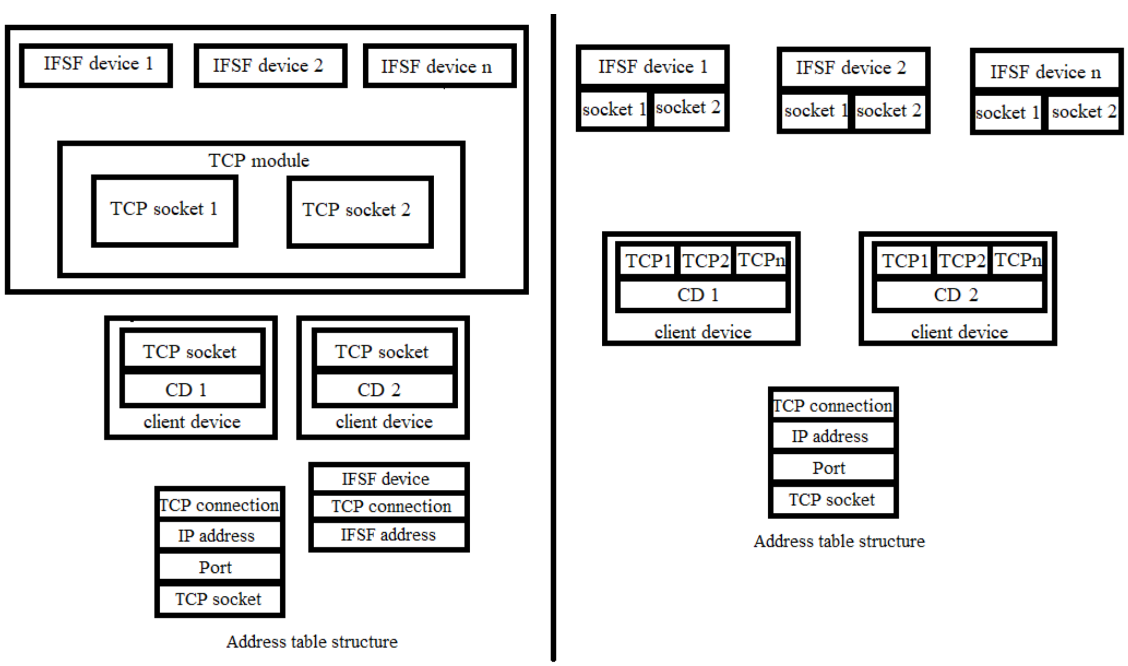

Figure 2: TCP connection between two hosts.

simple, as there is a special TCP connection between the two IFSF devices that must transfer IFSF messages.

To provide IFSF functionality when using the TCP / IP protocol of fixed IFSF periodic messages, they must be extended with additional data. All data is transmitted to the network in bytes.

CONCLUSIONS

In the designed system with one dual-control device, it is allowed to use two different protocol architectures for message transfer that contain TCP/IP module and address table. Very few system resources are consumed and an unlimited number of devices can be connected to the network. There is no need to convert protocols, thus reducing the cost of equipment and maintenance costs. The system is designed to meet the requirements compatible with the OSI model and has its own protocol for working with layer 7 and technical common implementation for layers 1 to 6. The system architecture allows all devices to communicate with each other. Devices of different origin can be connected without knowing the technical characteristics of the other devices. The communication layer is specified by the IFSF for TCP/IP.

REFERENCES

SchneiderAutomation, ModbusMessagingon TCP/IP ImplementationGuide, May 2019.

Technical documentation FAFNIR, VISY-X, VISY-Stick and VISY-Command, 2020.

Technical documentation NSYS, VISY-X, VISY-Stick and VISY-Command, 2020.

Technical Documentation, LonMark International, 2020.

Technologies and Communications, OSI The Model - Network Layers, http://techs-mobile.blogspot.bg/2011/03/osi.html.

Volz M (2018). Quo vadis Layer 7. The Industrial Ethernet Book, Issue 5, spring 2018.

Waft A (2019). Tunneling of Component Network Data over IP Channels, 2019.

Willig AA (2013). MAC Protocol and a Scheduling Approach as Elements of a Lower Layers Architecture in Wireless Industrial LANs. Proceedings of WFCS '97 (IEEE Int. Works. OnFactoryCommunication Systems), Barcelona, Spain. October, 2013.

Wollschlaeger M (2014) Framework for Web integration of factory communication systems, in IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), AntibesJuan-les-Pins, France, October 2014, pp. 261-265.

“VXIbusConsortium”, http://www.vxibus.org.

“PXI Systems Alliance”, http://www.pxisa.org.