Trace File Analysis in Fibre Channel over Ethernet Environment: A Step Ahead

Pranoti Kale
Associate Professor
Department of Computer Engineering,
Bharati Vidyapeeth's College of Engineering for Women,
Pune, India

Naina Chaturvedi
U.G Student
Department of Computer Engineering,
Bharati Vidyapeeth's College of Engineering for Women,
Pune, India

Neha Vishwakarma
U.G Student
Department of Computer Engineering,
Bharati Vidyapeeth's College of Engineering for Women,
Pune, India

Priyadarshni Sivakumaran
U.G Student
Department of Computer Engineering,
Bharati Vidyapeeth's College of Engineering for Women,
Pune, India

ABSTRACT

Enlightening a convergence of LAN and SAN engenders Fibre Channel over Ethernet Technology as one of the significantly important Ethernet Enhancements. Fibre Channel over Ethernet being a storage protocol, is proposed mapping of Fibre channel frames over existing Ethernet. Wireshark, an open source traffic analyzer, captures the live packets and provides detailed description of packet in the form of Trace Files containing hexadecimal values. These hexadecimal values represent the fields of frame format for the traffic it carries. This paper enlists steps showing equivalence between contents of the trace file and the fields of Ethernet Frame Format.

We propose that in FCoE environment, similarly we can co-relate our FCoE trace file and FCoE Frame Format. This paper provides an approach which proves beneficial for all those who wish to add their contributions in the world of FCoE regarding the reserved fields in the FCoE Frame Format. Mapping of trace file values with Frame Format fields will clear understanding whether the fields represent Time stamp or the length.

General Terms

Fibre Channel over Ethernet, Wireshark

Keywords

FCoE; Fibre channel; Frame Format; SAN; Trace file Wireshark.

INTRODUCTION

Emphasizing over cost effectiveness, reliability and high-performance protocol, FCoE is recent technology which provides converged fabric with server and storage virtualization which also inherits the ability to support both fibre channel and IP based protocols over a common link layer transport.

Wireshark, a network analyzer tool functioning in multiplatform, profoundly in real time depicts out the traffic situation in network by keenly monitoring and analyzing the live traffic. Being a open source protocol analyzer, preferably used by Network administrators for troubleshooting the network problems. The captured files by it can display the encapsulation and the fields along with their meanings of different packets specified by different networking protocols. It uses pcap to capture packets, so it can only capture the packets on the types of networks that pcap supports.

In this paper we focus on trace file generated by the protocol analyzer which depicts the fields of the frame format associated with the network technology and helps in recognizing the details of the protocols used in the network.

The motive of documenting this paper is to provide a head start for the novice users and developers who wish to work in the Open-FCoE [7] and the Wireshark Open Source [9] domain.

We find this document to be appropriate for relating the trace file values with the Frame Format, regarding which substantial information is not available on the World Wide Web.

Rest of the paper is organized as follows: Section II throws light on mapping of OSI stack and FC stack then presents an overall outlook on frame formats and Wireshark trace file. Section III A explains the detail steps showing how the contents of trace file are mapped to the Ethernet frame format. Section III B explains the similar approach for mapping the contents of the trace file with FCoE frame format. Section IV presents the conclusions of the paper.

1. AN OUTLOOK ON FRAME FORMATS AND WIRESHARK TRACE FILE

2.1 Ethernet Frame:

The preamble field is of 62-bit used to allow the transmitter and receiver to synchronize their communication. Start of frame delimiter is dedicated with 2-bit used to indicate the beginning of the frame. The destination MAC address field dedicated with 48-bit is the MAC address of the machine receiving data. Source MAC is the MAC address of the machine transmitting data of 48-bit. Length field indicates the length of the entire Ethernet frame in bytes. Data field has the data inserted. FCS (Frame Check Sequence) is calculated using the Cyclic Redundancy Check.

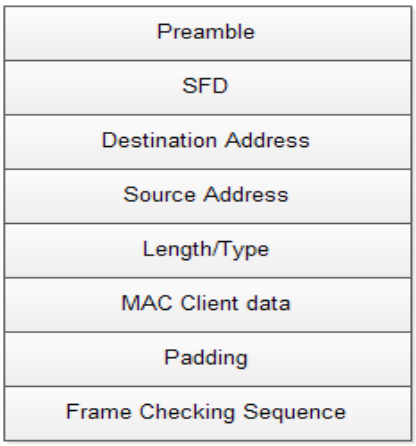


Figure 1 Ethernet Frame Format [11]

2.2 Fiber Channel Frame

Each frame begins and ends with a Frame Delimiter of 4-bytes. The Frame Header immediately follows the SOF delimiter dedicated with 24-bytes. The Frame Header is used to control link applications, control device protocol transfers, and detect missing or out of order Frames. Optional Header may contain further link control information. Payload contains the information to be transferred from source N_Port to a destination N_Port. The 4 bytes Cyclic Redundancy Check (CRC) is used to detect transmission errors.

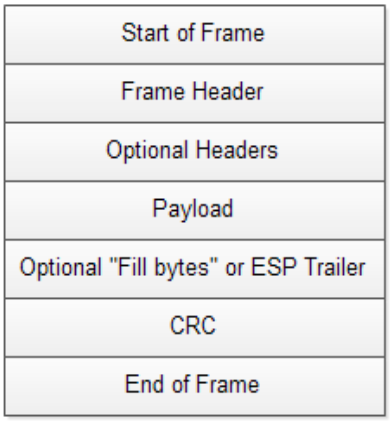
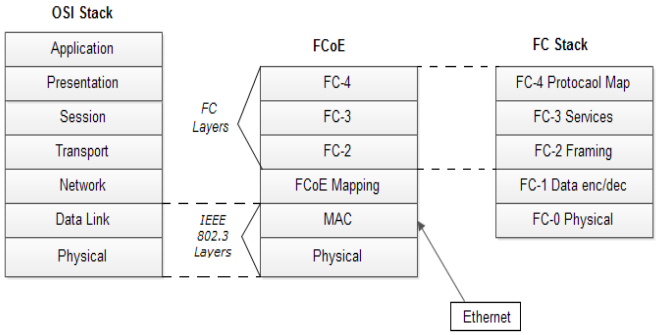


Figure 2 Fibre Channel Frame [11]

2.3 Mapping of IEEE 802.3 and FC stack

The diagram below shows the mapping between FC & FCoE. The most important layer here is FCoE mapping where the interfacing of two different protocols i.e. fibre channel & IEEE 802.3 layers is done. The FCoE Entity has an Ethernet MAC address that will be used as a source or destination address. Fibre Channel links are typically point-to-point and do not need an address at the link layer. An Ethernet network does not form an end-point to end-point connection same as FC. So FCoE need to rely on Ethernet MAC addresses to direct a frame to its correct Ethernet destination.

FCoE becomes the combination of Ethernet and the FC frame.



- Figure 3 Mapping of FC to FCoE Layers

2.4 FCoE frame format

The first 48-bits in the frame specify the Destination MAC address and next 48-bits specify the Source MAC address. The 32-bit IEEE 802.1Q Tag provides the same function as it does for virtual LANs, allowing multiple virtual networks across a single infrastructure. FCoE has its own Ether type as designated by the next 16 bits, followed by the 4-bit version field. The next 100 bits are reserved. Following the reserved field is the Start of Frame of 8-bit. The 8-bit End of Frame delimiter is followed by 24 reserved bits. The last 32-bits are dedicated to the FCS function that provides error detection for the Ethernet frame. The encapsulated Fibre channel frame consists of the original 24 byte FC header and the data being transported.

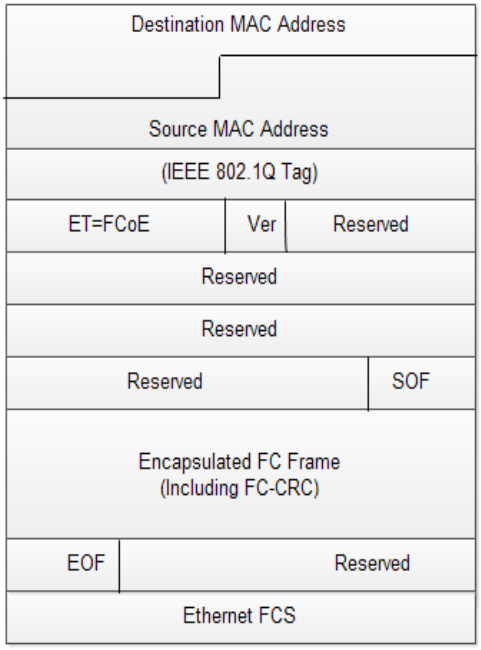


Figure 4 FCoE Frame Format [11]

2.5 Wireshark capture file details:

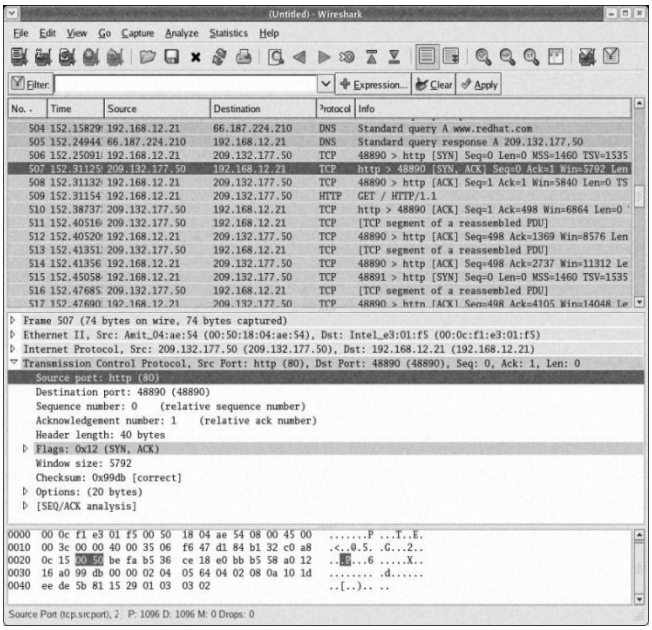


Figure 5 Wireshark capture file [9].

The above diagram shows the capture file of the Wireshark software. The Wireshark snapshot shown in the diagram basically consists of three main sections. First section is "Packet List" pane, which consist of various fields giving details of the captured data. Each line displayed in the packet list corresponds to one packet in the capture file. If you select a line in this pane, more details will be displayed in the "Packet Details" and "Packet Bytes" panes.

The default columns detailing the packet captured in "Packet List" pane are:

- **No**. - The number of the packet in the capture file.
- **Time** -The timestamp of the packet. There are various time formats present in Wireshark.
- **Source** - The address where this packet is coming from.
- **Destination** - The address where this packet is going to.
- **Protocol** - The protocol name in a short (perhaps abbreviated) version.
- **Info** - Additional information about the packet.

The Second important section is "Packet Details" pane, which shows the details of each captured packet in more depth. It gives details about the protocols used and protocol fields of the selected packet in the "Packet List" pane. The Third important section of the Window is "Packet Bytes" pane. This section consists also consist of the same details as shown in the "Packet Details" pane but in hexadecimal style, called as trace file. In the next section, how the contents of the trace file represent the fields of the Ethernet frame format is described. The trace file values also give the fields of the protocol header structure.

2. MAPPING OF TRACE FILE

3.1 Mapping contents of the trace file with Ethernet Frame format

The first step is to install Wireshark which can be obtained from [9]. Considering the Ethernet technology and Windows platform we can easily use Wireshark to capture the data packets.

1. Go to command prompt and get inside Wireshark Directory.

2. Execute `dir *.exe`

This command will display all the executable files present in Wireshark. The files present are capinfos.exe, dumpcap.exe, editcap.exe, mergecap.exe, text2pcap.exe, tshark.exe, uninstall.exe, WinPcap_4_0_1.exe, Wireshark.exe. These are the libraries and executables that capture the live traffic in the network and give detailed analysis.

The library file WinPcap_4_0_1.exe captures the traffic and hands it to Wireshark.exe. tshark.exe (textshark) captures the data from the network.

3. Execute tshark.exe –D

This command provides the list of interfaces we have in our machine. All the external and internal interfaces are listed on executing the command.

Note: Use tshark.exe –h to view the commands.

4. tshark -i <interface no.> and generate traffic i.e. execute ping command on another command prompt.

This shows the details of the traffic i.e. source IP address, destination IP address, protocol and information. The interface no. of the technology used can be obtained from the list provided after the execution of command 3.

5. tshark –i <interface no> -x

Generate traffic i.e. execute ping command on another command prompt. This command gives the contents of the trace file with the information given by command 4.

For all the captured data, a sequence of values is given. These are the ones contained in the trace file and they are the actual data sent through network.

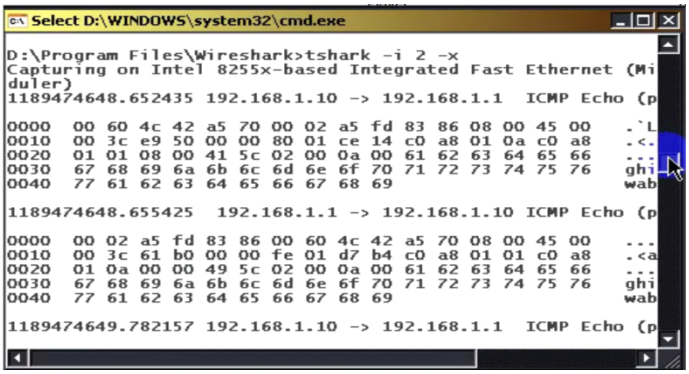


Figure 6 Trace file values after executing tshark –i <interface no> -x [6].

The figure shows the result of the command tshark –i <interface no> -x.

6. Here the technology used is Ethernet. Consider the frame format of Ethernet as shown in section II A. 6-bytes are allocated for the destination MAC address. Now considering

the values of the trace file, the first six hexadecimal values represent the destination MAC address.
You can cross check by executing arp –a command on other command prompt which gives the destination IP address and physical address. In the similar fashion the next 6 hexadecimal values give the source MAC address.

Now if you consider the next field of Ethernet frame i.e. length or type, it gives the type of protocol used. That means the next hexadecimal numbers denotes the fields of the protocol header structure. Here in this case the protocol is IP obtained from Ethernet frame. Therefore the next few hexadecimal values denote the fields of the IP header structure.
Figure [6] gives fields in the IP header structure. The values counting till the header length would represent the IP structure. From these values, the last 4 hexadecimal values indicate the destination address.

Calculating in the similar fashion we get the protocol used in the IP header structure. In the above case it is ICMP. So the remaining hexadecimal values represent the contents of the ICMP header structure.

Here we come to know the importance of the values in trace file which gives the actual data sent through the network. This data can be used for troubleshooting errors, network security issues.

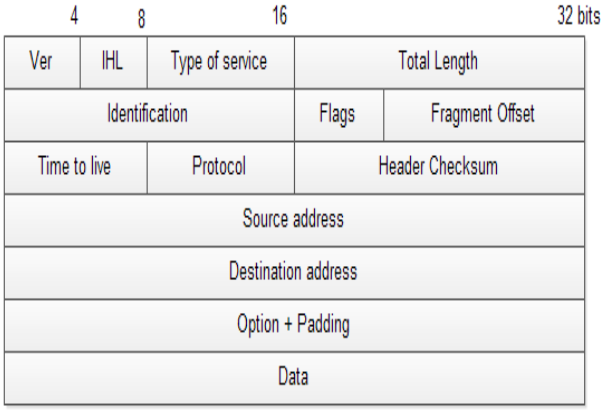


Figure 6 IP header structure [11]

3.2 Similar approach towards FCoE frame format

As we have seen the information of the traffic, Protocols and protocol fields can be derived from the trace files in Ethernet technology, in similar way we can derive the same information in Fibre Channel over Ethernet (FCoE) technology.

For implementing this we have to create FCoE initiator and target setup, which can be done with the help of scripts available in the paper [5] or the open source reference [11]. This includes various steps like Kernel compilation, building bzImage, making and installing the required modules etc. finally followed with setting of initiator and target.

The latest version of Wireshark should be installed from the Wireshark repositories. After Wireshark is installed the same steps can be followed as in the case described in section III A for finding out the details of the live data in FCoE.

Brocade and Cisco have put forth their views regarding what the reserved fields in the FCoE frame format indicate. Analyzing the trace file will help researchers to find out what data the reserved fields of the FCoE Frame format indicate. The work regarding the same is under progress at our level.

3. CONCLUSIONS

Industry leading vendors including, Brocade, Cisco, EMC, Emulex, IBM, Intel, Nuova, QLogic, and Sun Microsystems supports FCoE. Brocade and Cisco have proposed their different views regarding the content of the reserved fields of FCoE frame format. This paper proposes an approach to relate the trace file values with the Frame format of Fibre channel over Ethernet. We hope that this approach proves beneficial for all those who wish to add their contributions in the world of FCoE regarding the reserved fields in the Frame Format.

4. ACKNOWLEDGMENT

The approach would not have seen the light of the day without support of Dr. Anupam Bhide, who motivated us to initiate work in this area. We are also grateful to the Open-FCoE and Wireshark community who made us available such great open source technologies.

Last but not the least; we thank all the authors of various research papers and articles who have provided us directions in this domain

5. REFERENCES

[1] R. Love, “Linux Kernel Development”, Third Edition, pp. 2-20, 2010, Addison-Wesley, ISBN: 978-0-672-32946-3

[2] Claudio DeSanti and Joy Jiang. FCoE in Perspective. In Proceedings of International Conference on Advanced Infocomm technology, 2008.

[3] Joy Jiang and Claudio DeSanti. The role of FCoE in I/O Consolidation. In Proceedings of International Conference on Advanced Infocomm technology, 2008.

[4] Michael Ko, Daniel Eisenhauer, and Renato Recio. A Case for Convergence Enhanced Ethernet: Requirements and Applications. In ICC, pages 5702–5707, 2008

[5] Fibre Channel over Ethernet-A Head Start reference, Pranoti Kale, Ashwin Tumma, Harshada Kshirsagar, Pooja Ramrakhyani, Tejashri Vinode.

[6] Introduction to Wireshark, Hakim, Video reference: *http://www.youtube.com/watch?v=jzkUuc5jK8Q*

[7] Official Website of Open-FCoE, *http://www.open-fcoe.org/open-fcoe.*

[8] The Linux Kernel Mailing List: *http://lkml.org/.*

[9] Official Website of Wireshark *http://www.wireshark.org*

[10] Official Website for Protocols *http://www.protocols.com.*

[11] Steps for installation of the initiator and the target of FCoE: *http://www.mail-archive.com/devel@open-fcoe.org/msg03299.html.*