Analysis and Simulation of Hyper Text Transfer Protocol at the Application Layer of the Internet

Naing Naing Kyaw
Department of Electronic Engineering, Technological University (Taunggyi), Myanmar

DOI: 10.29322/IJSRP.9.01.2019.p8512
http://dx.doi.org/10.29322/IJSRP.9.01.2019.p8512

Abstract- The Internet and computer networks make extensive use of protocols. This paper is mainly focused on HTTP which is one of the most important piece of application layer protocol. Application layer is the topmost layer in OSI and TCP/IP model. The general format of HTTP request and response messages in real world are analyzed and captured by Wireshark. The actions taken by the Web client and server as these messages are sent and received are simulated and monitored using Packet Tracer.

Index Terms- Analysis, Simulation, HTTP, Packet Tracer, Wireshark

I. INTRODUCTION

The Internet is a computer network that interconnects hundreds of millions of computing devices throughout the world. Standards come into play to create systems and products that can interoperate. Internet standards are developed by the Internet Engineering Task Force (IETF). The IETF standards documents are called requests for comments (RFCs). RFCs started out as general requests for comments to resolve network and protocol design problems that faced the precursor to the Internet. Different protocols are used to accomplish different communication tasks. A protocol defines the format and the order of messages exchanged between two or more communicating entities, as well as the actions taken on the transmission and/or receipt of a message or other event.

It is important to distinguish between network applications and application-layer protocols. An application-layer protocol is a very important piece of a network application. The Web is a client-server application that allows users to obtain documents from Web servers on demand. The Web application consists of many components, including a standard for document formats such as Hypertext Markup Language (HTML), Web browsers (Firefox and Microsoft Internet Explorer), Web servers (Apache, Python and Microsoft servers), and an application-layer protocol. A Web page consists of objects. An object is simply a file—such as an HTML file, a JPEG image, a Java applet, or a video clip—that is addressable by a single URL (Universal Resource Locator). Most Web pages consist of a base HTML file and The Web's application-layer protocol, HTTP, defines the format and sequence of messages exchanged between browser and Web server [4].

This paper is organized as follows. Section I is introduction about the Internet. The difference between OSI reference model and TCP/IP model is illustrated in section II and III. In section IV, overview of HTTP is explained with the request and response format. Programming with sockets in client/server interaction is presented in section V. Web pages are created by using Hypertext Markup Language is involved in section VI. Capture the format and sequence of messages exchanged by using Wireshark is described in section VII. Then, the simulation of HTTP on Packet Tracer software is introduced in section VIII and finally, discussion and conclusion are included in section IX.

II. OSI REFERENCE MODEL

Open System Interconnection Reference Model (OSI RM) promotes the development of network technology. Launched by ISO (International Organization for Standardization) in the 1980s, OSI is a seven-layer function/protocol model. Looking at the OSI model, data transmitted from a computer will be transferred from the highest layer (Layer 7) to the lowest layer (Layer 1). Figure 1 shows how data is transferred from one layer to the next in a simple network consisting of only two computers (Computer A and Computer B) connected by a network cable.

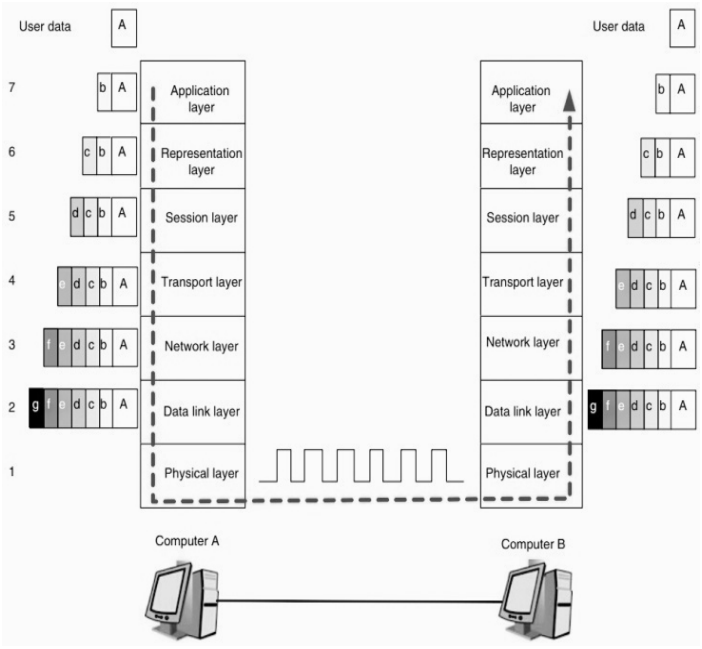

Figure 1: Data encapsulation and decapsulation process on communication terminals in the OSI model

The data is encapsulated during the transfer process until it is converted to optical/electrical signals and sent out from the physical layer. Conversely, data received by a computer will be transferred from the lowest layer to the highest layer and be

decapsulated. Figure 2 shows the hierarchical structure of the OSI model which is used in Packet Tracer [2].

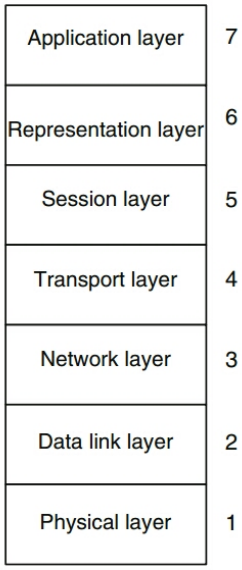

Figure 2: Hierarchical structure of the OSI model

III. TCP/IP MODEL

Originating from ARPANET (Advanced Research Projects Agency Network) and enhanced by the IETF, the TCP/IP model is formed from two important protocols, the Transmission Control Protocol (TCP) and Internet Protocol (IP). To download a 2000 byte song on the Internet, the user's Web browser first sends a request. Before the song is sent, it will be encapsulated through the layers on the Web server. The application layer will add an HTTP header to the original song data to form an HTTP datagram.

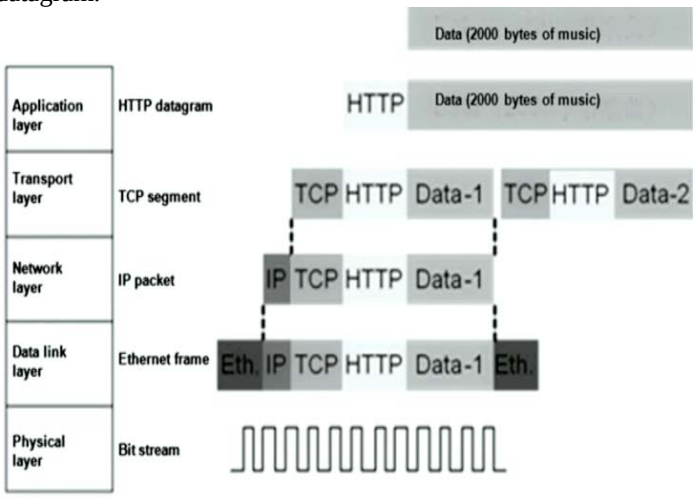

Figure 3: Data encapsulation process in the TCP/IP model

Because the HTTP datagram is too long it will be segmented into two parts in the transport layer, with a TCP header added to the front of each part to form two TCP segments. At the network layer an IP header will be added to each TCP segment to form an IP packet. When the IP packet reaches the data link layer, assuming the layer is using Ethernet technology, it will add an Ethernet frame header and trailer to the IP packet, forming an Ethernet frame. Finally, the physical layer will convert these Ethernet frames into a bit stream as shown in Figure 3. Figure 4 shows the hierarchical structure of the TCP/IP model which is used in Wireshark [2].

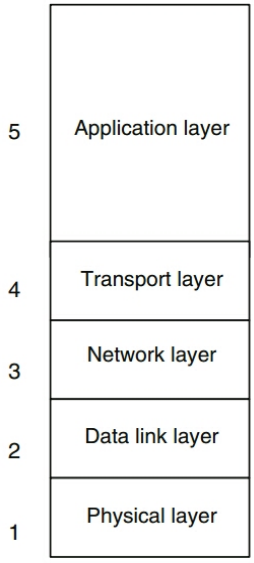

Figure 4: Hierarchical structure of the TCP/IP model

IV. OVERVIEW OF HTTP

The Hyper Text Transfer Protocol (HTTP) is the main protocol used by World Wide Web for communication. HTTP, the Web's application-layer protocol, is at the heart of the Web. HTTP is implemented in two programs: a client program and a server program. The client program and server program, executing on different end systems, talk to each other by exchanging HTTP messages. HTTP defines the structure of these messages and how the client and server exchange the messages. HTTP defines how Web clients request Web pages from Web servers and how servers transfer Web pages to clients. When a user requests a Web page (for example, clicks on a hyperlink), the browser sends HTTP request messages for the objects in the page to the server. The server receives the requests and responds with HTTP response messages that contain the objects. The general idea of the interaction between client and server is illustrated in Figure 5 [4].

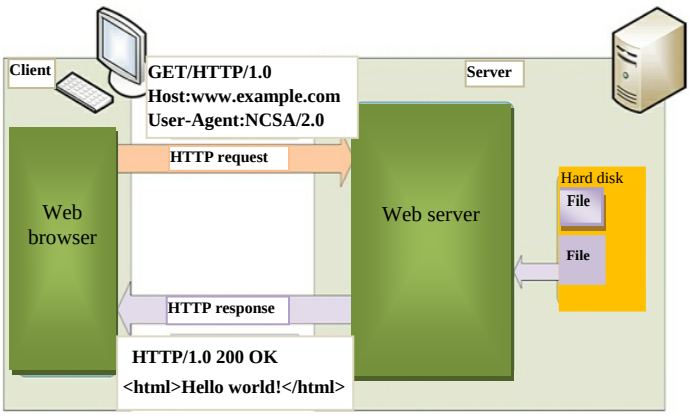

Figure 5: HTTP request response behavior

HTTP communication usually takes place over TCP/IP connections. HTTP is a clear text protocol and it is not secure. The default port is TCP 80. HTTPS is a similar protocol that enables encryption for added security. The default port is TCP 443. This allows for a more secure form of data transfer [2]. HTTPS is HTTP using a Secure Socket Layer (SSL) [10].

The HTTP specifications include the definitions of the HTTP message formats. There are two types of HTTP messages, request messages and response messages, both of which are discussed below. First, a typical HTTP request message is provided:

GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr

The general format of a request message is shown in Figure 6 and response message is shown in Figure 7.

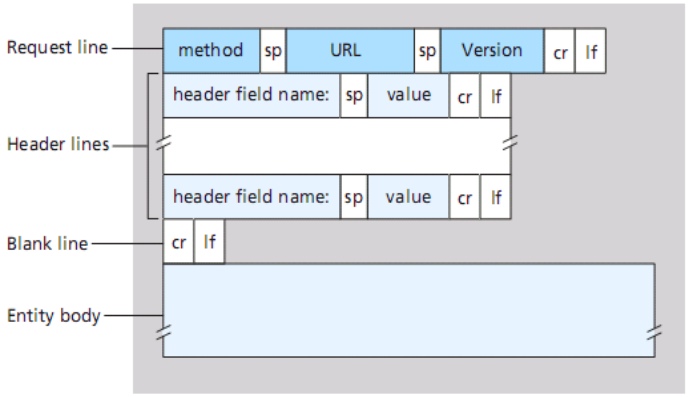

Figure 6: General Format of HTTP Request Message

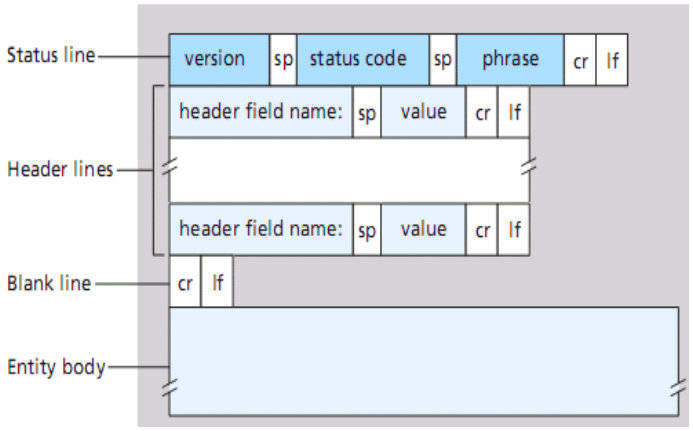

Figure 7: General format of HTTP response message

After that, a typical HTTP response message is provided as follows:[4]

HTTP/1.1 200 OK
Connection: close
Date: Tue, 09 Aug 2011 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Tue, 09 Aug 2011 15:11:03GMT
Content-Length: 6821
Content-Type: text/html
(data data data data data ...)

V. PROGRAMMING WITH SOCKETS

Network programming in any programming language can begin with sockets. A network socket is a virtual end point where entities can perform inter-process communication. For example, one process sitting in a computer, exchanges data with another process sitting on the same or another computer. The first process which initiates the communication is the client and the latter one is the server. Python has quite an easy way to start with the socket interface.

A flow of client/server interaction is shown in Figure 8. This will give the usage of socket API (Application Programming Interface). In the interaction between a typical client and a server, the server process has to work a bit more. After creating a socket object, the server process binds that socket to a particular IP address and port. After the successful binding, the server process will start listening for a new client connection.

For a valid client session, the server process can accept the request of the client process. At this point, the connection between the server and the client has been established. Then the client/server enters into the request/response loop. The client process sends data to the server process, and the server process processes the data and returns a response to the client. When the client process finishes, it exits by closing down the connection. At that moment, the server process probably goes back to the listening state [3].

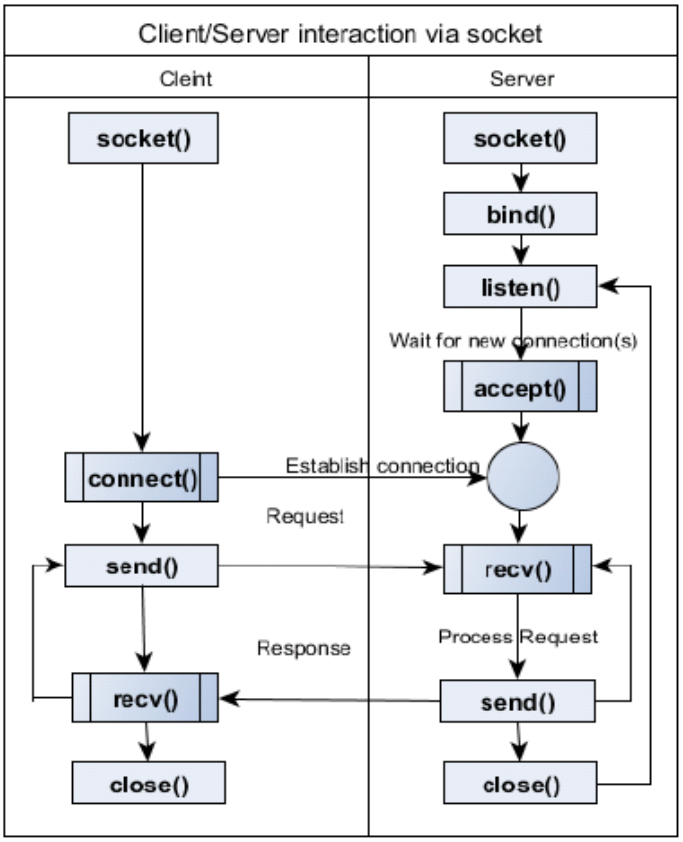

Figure 8: Client/Server interaction through socket

VI. HTML

HTML or Hypertext Markup Language is a formatting language that programmers and developers use to create documents on the Web. HTML5 is being developed as the next

major revision of HTML. This can now be used for new functions that can benefit developers and Internet users.

The latest edition HTML5 has enhanced features for programmers such as <video>, <audio> and <canvas> elements. The users view a Web page written in HTML in a Web browser such as Internet Explorer, Mozilla Firefox or Google Chrome. The HTML5 language has specific rules that allow placement and format of text, graphics, video and audio on a Web page.

Programmers use these programming tags or elements to produce web pages in unique and creative ways. Tags such as <section>, <article>, <header> enable the creator to make a more efficient and intelligent Web page [9].

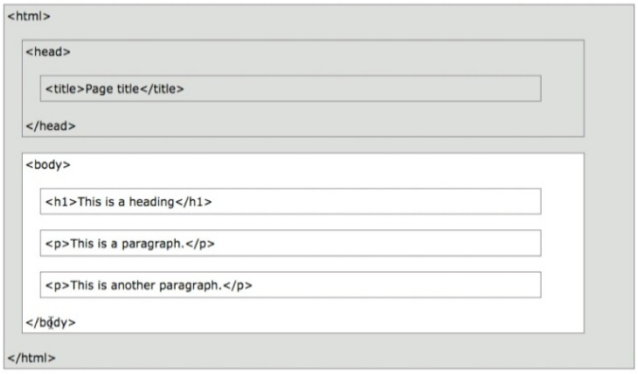

Figure 9: HTML page structure

Browsers do not display the HTML tags, but use them to render the content of the page. A visualization of a HTML page structure is shown in Figure 9.

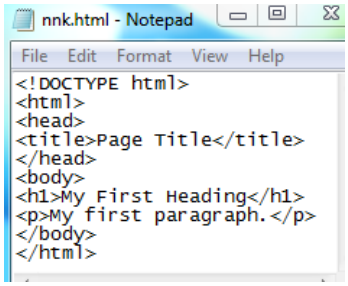

Figure10: HTML code example

Figure10 shows a small HTML code for a simple page. The code may be explained as follows:

- The<! DOCTYPE html>declaration defines the document type to be HTML5
- The <html>element is the root element of an HTML page
- The <head> element contains meta information about the document
- The <title> element specifies a title for the document
- The <body> element contains the visible page content
- The <h1> element defines a large heading
- The <p> element defines a paragraph

The code may be written in any editor such as notepad or word pad. The document should be saved as .html and then opened from any Web browser such as Internet Explorer, Google Chrome or Firefox. Figure 11 shows the result of applying the previous code [1].

Figure 11: HTML code result on Web browser

VII. WIRESHARK

Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible. A network packet analyzer can be thought of as a measuring device used to examine what's going on inside a network cable [8].

Wireshark is one of the best open source packet analyzers available today. Some intended purposes of Wireshark are as follows:

- Network administrators use it to troubleshoot network problems
- Network security engineers use it to examine security problems
- QA (Quality Assurance) engineers use it to verify network applications
- Developers use it to debug protocol implementations
- People use it to learn network protocol internals

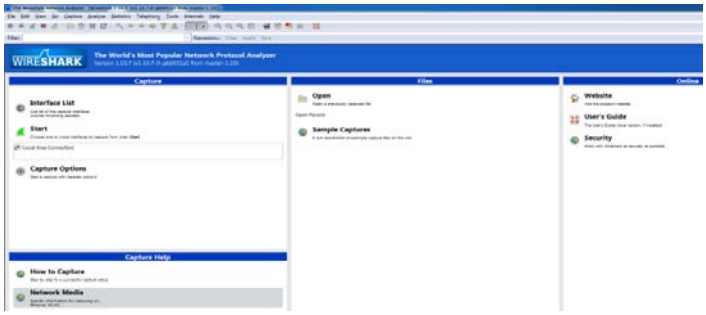

Figure 12: Wireshark network protocol analyzer

When Wireshark software is running, the home page screen is depicted as shown in Figure12.

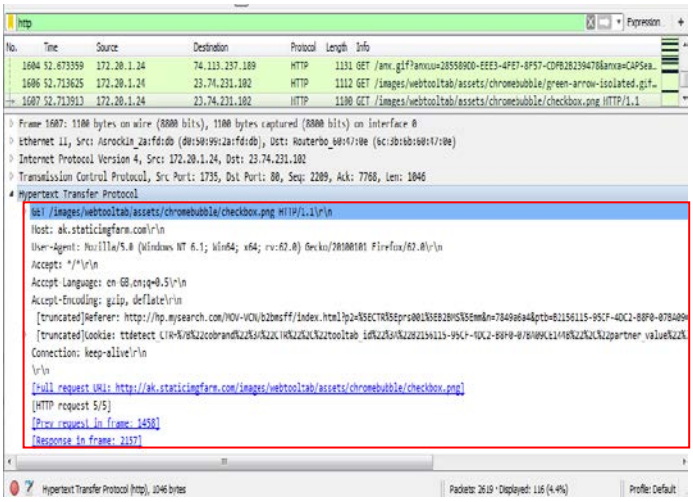

Figure13: HTTP request

As shown in Figure 13, when 172.20.1.24 send a request to 23.74.231.102 , the request is sent over transport layer, network layer and data link layer of the client and it can be seen above the red box. Inside the box represents the HTTP request message format in real world environment. So Wireshark is based on the TCP/IP model. The transport layer use Transmission Control Protocol with source port 1735 and destination port 80. Port 80 is the default port for HTTP protocol. In network layer Internet Protocol Version 4 with source address 172.20.1.24 and destination address 23.74.231.102 are also included.

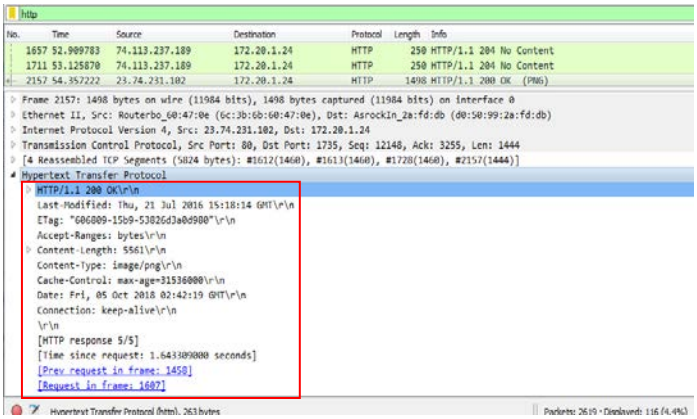

Figure 14: HTTP response

When 23.74.231.102 send a response to 172.20.1.24, this response is sent over transport layer, network layer and data link layer of the server and it can be seen above the red box. Inside the box represents the HTTP response message format in real world environment. Again the transport layer use Transmission Control Protocol but this time source port is 80 and destination port is 1735. Similarly, in network layer Internet Protocol Version 4 with source address 23.74.231.102 and destination address 172.20.1.24 are included as shown in the above Figure 14.

VIII. PACKET TRACER

Packet Tracer is a protocol simulator developed by Dennis Frezzo and his team at Cisco Systems. Packet Tracer (PT) is a powerful and dynamic tool that displays the various protocols used in networking. Cisco Packet Tracer provides two operating modes to visualize the behavior of a network, real-time mode and simulation mode.

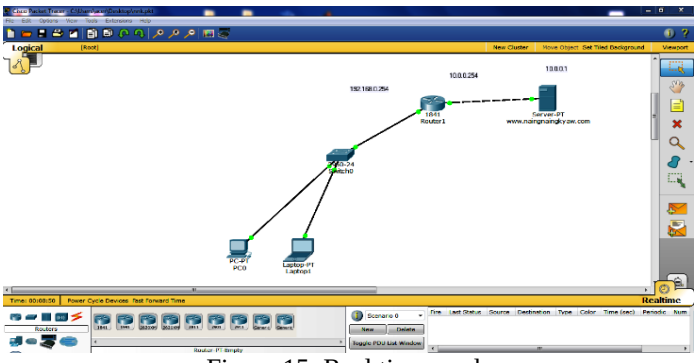

Figure 15: Real time mode

In real-time mode as shown in Figure 15, the network behaves as real devices do, with immediate real-time response for all network activities. The real-time mode gives students a viable alternative to real equipment and allows them to gain configuration practice before working with real equipment.

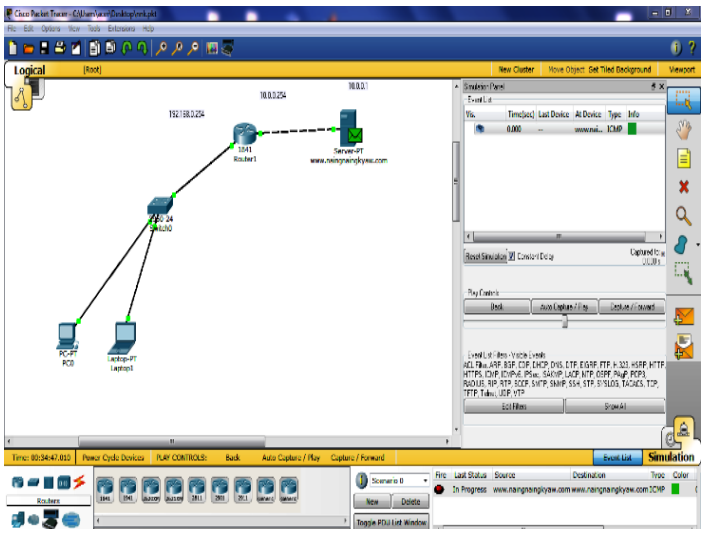

Figure 16: Pinging to Web server

In simulation mode as shown in Figure16, the user can see and control time intervals, the inner workings of data transfer, and the propagation of data across a network. Students can visually analyze a packet and its' details from the simulation mode. This helps students understand the fundamental concepts behind network operations [4].

Cisco Packet Tracer has two workspaces: logical and physical. The logical allows users to build logical network topologies by placing, connecting, and clustering virtual network devices. The physical workspace provides a graphical physical dimension of the logical network, giving a sense of scale and placement in how network devices such as routers, switches, and hosts would look in a real environment.

The physical view also provides geographic representations of networks, including multiple cities, buildings, and wiring closets [7]. The physical view of main wiring and wiring closets are depicted in Figure 17 and 18 respectively.

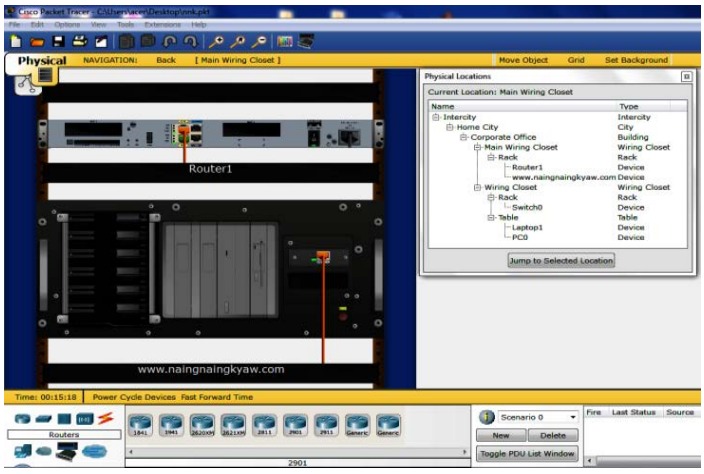

Figure 17: Physical view of main wiring closet

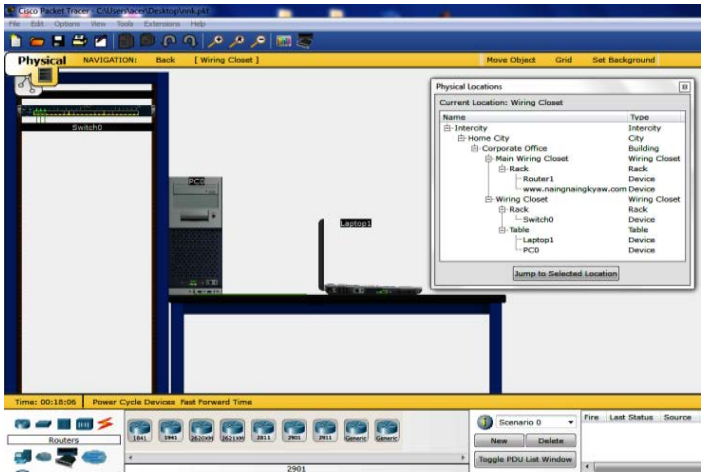

Figure 18: Physical view of wiring closet

In the above Figures, the real devices such as router, server, switch, desktop and laptop can be seen in physical nature.

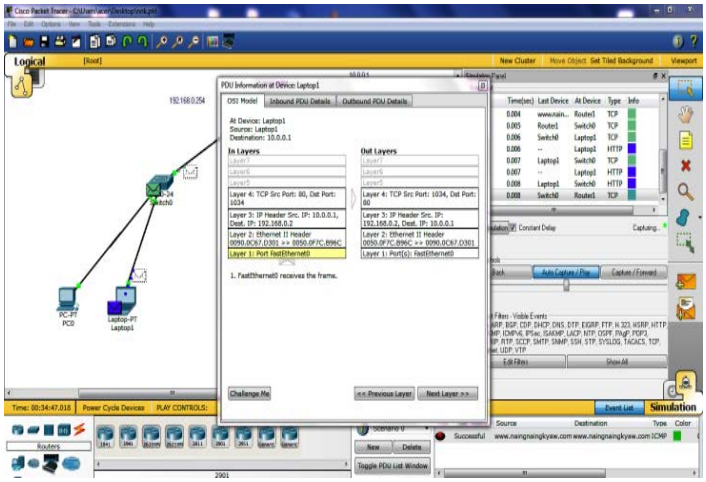

Figure 19: Logical view of TCP connection establish

In the logical environment, one router, one switch, one web server, one desktop and one laptop computers are used to investigate the working principles of HTTP over TCP or UDP. When a Web browser request Web page from the Web server the TCP connection is established as shown in Figure 19.

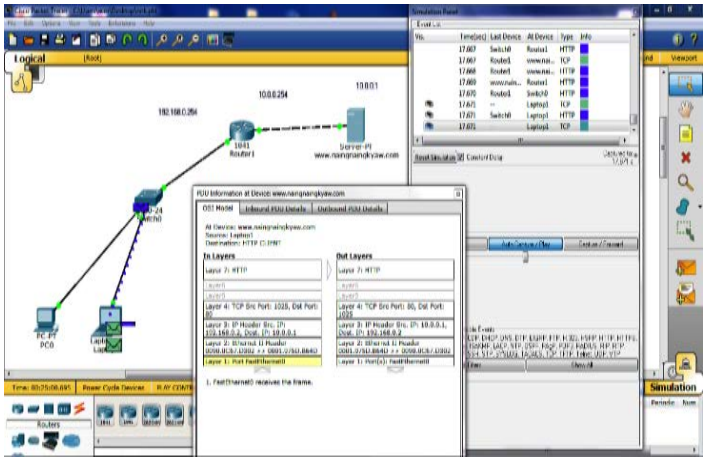

Figure 20: HTTP request and response in OSI model

Packet tracer uses the OSI seven layer protocol model and HTTP works on the application layer. Functions of HTTP request and response made on TCP are illustrated in Figure 20.

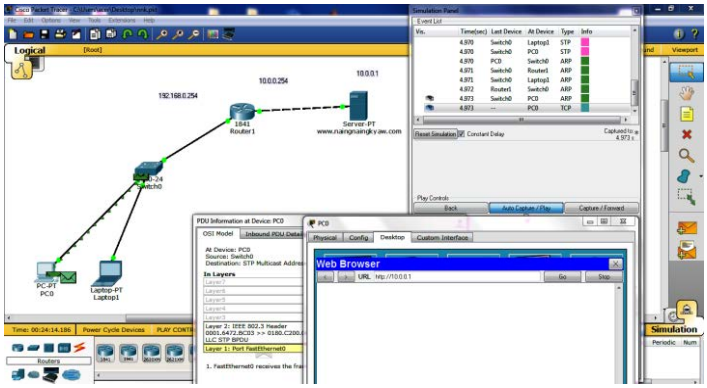

Figure 21: Web browser browse a Web page

When the user requests a Web page from the PC, it start works on OSI model as shown in Figure 21.

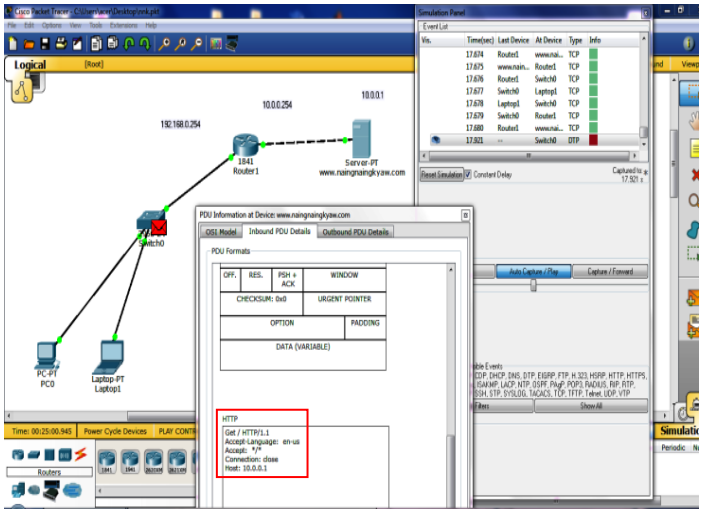

Figure 22: HTTP request format

Figure 22 shows the HTTP request format in the red box when the laptop sent a request to the Web server.

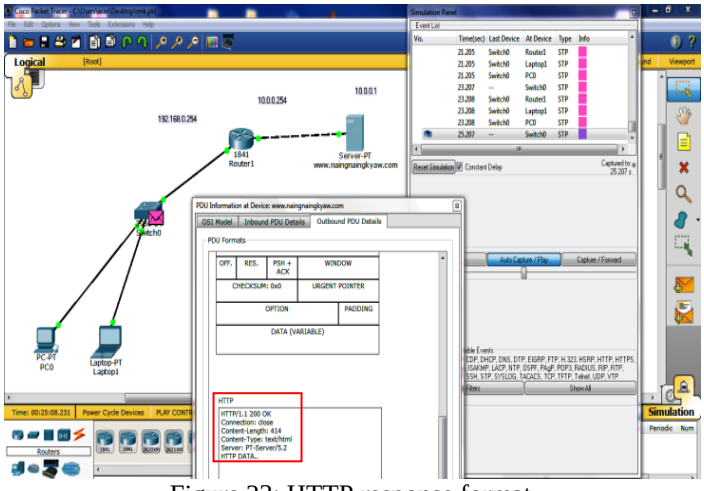

Figure 23: HTTP response format

Figure 23 shows the HTTP response format in the red box when the Web server sent a response to the laptop.

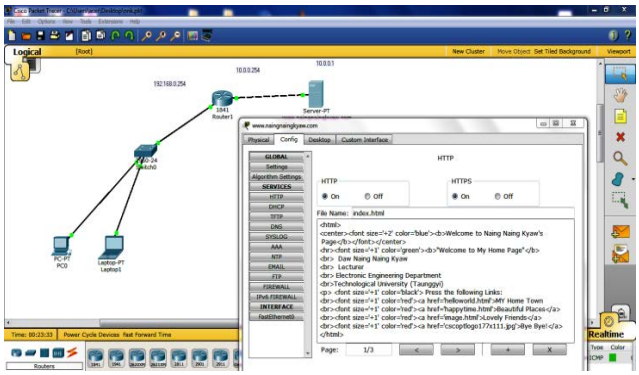

Figure 24: Programming with HTML

Figure 24 shows how the Web page is created with HTML language and the result of Web page can be monitored in the following simulation process of the Packet Tracer as in Figure 25.

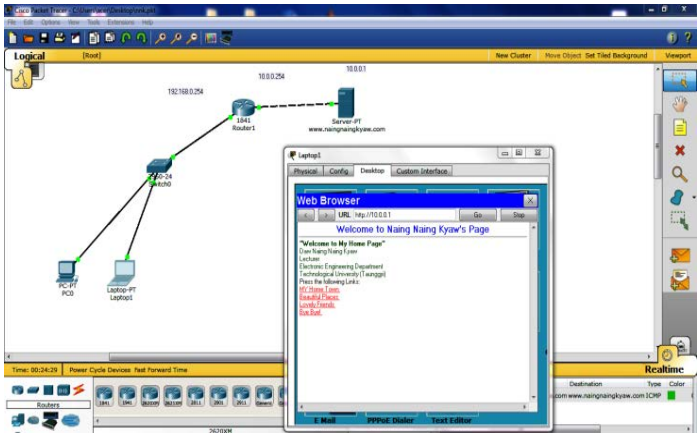

Figure 25: Web page result

IX. DISCUSSION AND CONCLUSION

HTTP functions as a request-response protocol in the client-server computing model. The client submits an HTTP request message to the server. The server which provides resources such as HTML files and other content, returns a response message to the client. Socket programming is also introduced because most of the application layer protocols like HTTP, FTP, SMTP and POP3 make use of sockets to establish connection between client and server and then for exchanging data. In this paper, analysis of HTTP request and response protocol message formats is captured with Wireshark. The simulation of this protocol is done via Packet Tracer.

ACKNOWLEDGMENT

The author wishes to thank her parents for their inspiring determination, supports and encouragement to attain her attention without any trouble throughout her life. The author would also like to express her thanks to all teachers who taught her everything from childhood till now.

REFERENCES

[1] "HTML Introduction," w3schools.com, 20.9.2016. [Online]. Available: http://www.w3schools.com/html/htmlintro.asp.
[2] "HCNA Networking Study Guide": Huawei Technologies Co., Ltd. Shenzhen, China, 2016.
[3] Dr. M. O. Faruque Sarker, "Learning Python Network Programming," June 2015.
[4] James F. Kurose, Keith W. Ross. "Computer networking : a top - down approach," 6th edition, 2015.
[5] Airi and Anderson, "Cisco Packet Tracer as a teaching and learning tool for computer networks in DWU", 2015.
[6] CC-HTTPARC.pdf. 2014. [Online]. Available: https://oa.mov.gov
[7] Cisco Packet Tracer Datasheet, 2010.
[8] Wireshark User's Guide: Version 2.9.0.
[9] http://en.wikipedia.org/wiki/HTML5
[10] httpvshttps.pdf.[Online]. Available: http://msdn.Microsoft.com/enus/library/aa767735%28v= vs.85%29.aspx

AUTHOR

Naing Naing Kyaw – Lecturer, Department of Electronic Engineering, Technological University (Taunggyi), Myanmar. Email: naingnaingkyaw.tgi@gmail.com