

# Assignment 1

Songtuan Lin u6162630

April 7, 2019

## Question 1

For simplicity, I will assume the *HTTP Request* message the server received request a Web Page.

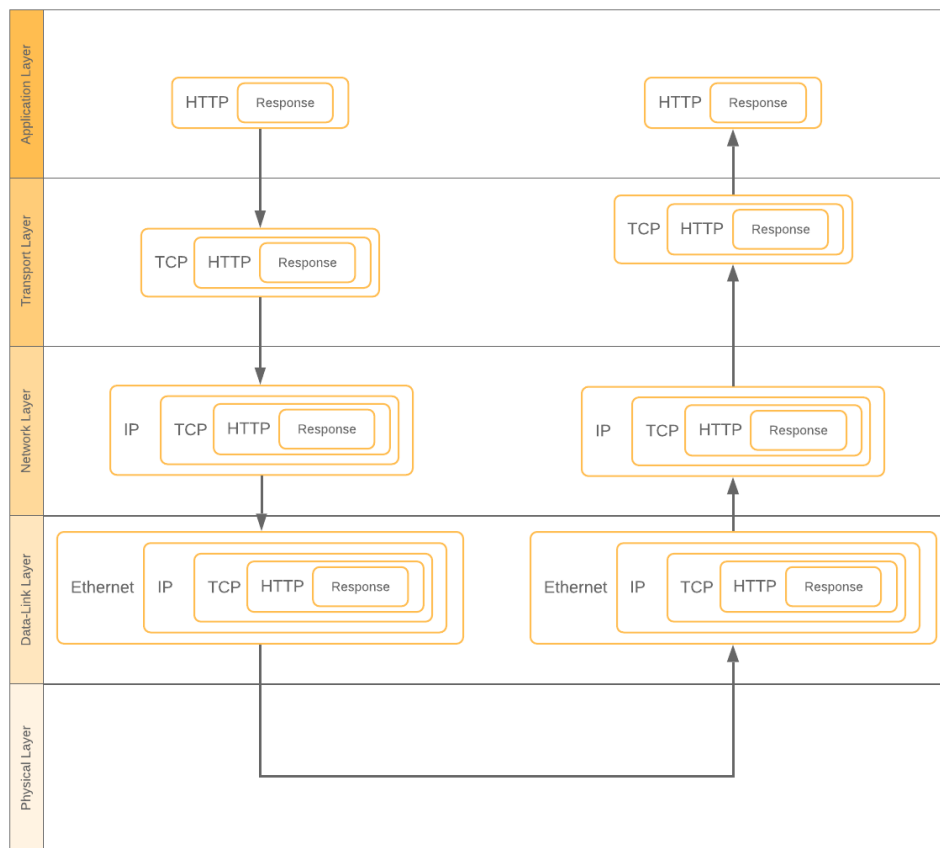


Figure 1: Transmission through Layers

By comparing Figure 1 with the original Figure, it can be seen that the HTTP Response message will go through the same Network Layer as the HTTP Request message. The only difference shown in these two figures is the *Request* field and *Response* field. Particularly, in this example, the client *request* a Web Page and the server *response* the corresponding html file. Indeed, this simple observation implies a general rule for the communication through computer network:

The message delivery within computer network must go through a series of fixed stages. Each stage will accomplish some specific functions and provide them to be used by another stages.

Technical, each stage correspond to one Layer, hence, form the modern *Layered-Architecture* as illustrated in Figure 1. Additionally, in order to maintain the generality of the message within Computer Network, which means the message can be decoded by different host, further rules that specify how each layer accomplish their function have been made. These layer-specific rules are called *protocol*, e.g. HTTP in Application Layer, TCP in Transport Layer, etc. Generally, each protocol need to add some extra information to the original message

in order to accomplish their corresponding function. This additional information is called *head*. e.g. HTTP head, TCP head, etc. In the following part, I will briefly explain the function and some common protocols corresponding to each layer:

- **Application Layer:** Application Layer produce the message that will be delivered and transfer it to next Transport Layer. Some popular Application Layer protocols include: HTTP, FTP, etc.
- **Transport Layer:** Transport Layer accomplish the preparation tasks before the delivery, e.g. specify the source and target application of the message, maintain the connection between client and server. The common Transport Layer protocols include: TCP and UDP.
- **Network Layer:** Network Layer determine the route of message delivery, which is one of the most important function. The common Network Layer protocols include OSPF, BGP, etc.
- **Data-Link Layer:** Data-Link Layer further determine the route based on the route calculated by Network Layer. The common protocols in Data-Link Layer include Ethernet, WIFI protocol family, etc.
- **Physical Layer:** Physical Layer is responsible for the actual transmission.

The above explanation may be too abstract. Indeed, we can think Computer Network as a real world package delivery system, hence, we can have a concrete understand about Layer-Architecture by going through an example: Assume a package delivery company have a package that want to be delivered from Australia to China. The entire delivery process, from Computer Network perspective, can be expressed as:

1. The Application Layer get a package from customer that need to be delivered.
2. The Transport Layer confirm this package delivery order.
3. The Network Layer determine the route of delivery. Particularly, it determine which country will be in the delivery path, e.g. Australia → Indian → China
4. The Data-Link Layer determine the route that how package will be delivered within country.
5. The physical Layer carry the package and do the actual transmission.

The above discussion give a theory explanation about Layer-Architecture. Indeed, by using software tools like Wireshark and cURL, we can do some practical experiments to see how Layered-Architecture model work. These practical experiments will be demonstrated in the rest question within .this assignment.

## Question 2

---

In this question, I will use cURL to inspect HTTP protocol. This experiment can illustrate the format of HTTP Request and Response message. In order to complete this question, firstly, I need to configure my web server following the setting up instruction described in the question. Particularly, the web server I used is a droplet from Digital-Ocean with IP address: 188.166.167.192. After I have deployed my web server, I can send a HTTP request through cURL by using following command:

```
curl -v http://188.166.167.192:8000
```

The output of this cURL command is shown in Figure 2, In which, there are several important content:

1. HTTP Request message, which consist of the lines start with 'i'. Each line, according to RFC(HTTP), have the following purpose:
  - **Head Line:** This first line specify the request method and the HTTP protocol version.
  - **Host:** Specify the IP address or domain name of the server.
  - **User-Agent:** Specify the application that sent this request message along with its' version. In this example, the request is come from cURL tool.
  - **Accept:** Specify the media types which are acceptable for the client. i.e. video stream and audio.
2. HTTP Response message, which consist of the lines start with 'I'. The meaning of each line is:
  - **Head Line:** Serve the same purpose as that in Response message
  - **Server:** Specify the server application
  - **Date:** The date this response was sent
  - **Content-Type:** Specify the file type that this response message will send

```

$ curl -v http://188.166.167.192:8000
* Expire in 0 ms for 6 (transfer 0xdf38c0)
* Trying 188.166.167.192...
* TCP_NODELAY set
* Expire in 200 ms for 4 (transfer 0xdf38c0)
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %         Dload  Upload  Total   Spent    Left     Speed
connected to 188.166.167.192 (188.166.167.192) port 8000 (#0)
> GET / HTTP/1.1
> Host: 188.166.167.192:8000
> User-Agent: curl/7.64.0
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Server: SimpleHTTP/0.6 Python/3.6.7
< Date: Sat, 06 Apr 2019 04:28:40 GMT
< Content-type: text/html
< Content-Length: 141
< Last-Modified: Sat, 06 Apr 2019 03:55:40 GMT
<
{ [141 bytes data]
100  141  100  141    0     0   192      0 --:--:-- --:--:-- --:--:--  193K
HTML>
<HEAD>
<TITLE>NIS2019 Assignment Page</TITLE>
</HEAD>
<BODY>
<H1>NIS2019 Assignment Test Page for u6162630</H1>
<HR>
</BODY>
</HTML>

* closing connection 0

```

Figure 2: Output of cURL

- **Content-Length:** Specify the length of the file that will be transformed
  - **Last-Modified:** Specify the last date that the file which are willing to be sent is modified.
3. HTML file, which locate in the last part of the output. This HTML file is the last part of HTTP Response message and correspond to the Web Page requested by clients.

Instead of using cURL to send HTTP Request, we can use simple Web Browser to retrieve the same Web Page, which will be shown in Question 3.

## Question 3

The HTML file that been created in web server can be retrieved through web browser by typing the following command:

http://188.166.167.192:8000

The result that displayed in web browser is:

In this experiment, the web browser send a HTTP Request and receive a HTTP Response from web server. The transmission of HTTP Request and HTTP Response can be illustrated by Figure 1 and follow the general discussion of Layer-Architecture Model in Question 1:

1. **Application Layer:** The web browser in client side(Application) will generate a message called *packet*, which consist of a HTTP head(the definition of head has been given in Question 1) and a request message. This packet will be passed to Transport Layer to have further modification.
2. **Transport Layer:** Transport Layer will choose TCP protocol to establish the connection between client and server. Hence, the three-way handshake is required and the TCP head will be added to the original packet.The TCP head append with Application Layer packet is called segment.
3. **Network Layer:** Network Layer will calculate the best route for this transmission and add a IP head to the segment, which produce a Network Layer packet.
4. **Data-Link Layer:** Data-Link Layer also determine the route, particularly, it determine the transmission route within Local Area Network(LAN). In order to do so, Data-Link Layer need add Ethernet head to Network Layer packet, which produce a frame.
5. **Physical Layer:** Physical Layer take the frame produced by Data-Link Layer and transport it to the server side.

After the server receive the HTTP Request. This message will go through the same Network Layer but in inverse order. Differ from the client side, each Layer in server side will remove the corresponding head. After the sever has processed the request, it will send a HTTP Response through Layer-Architecture Model and reach client. After that, the web browser can finally display the web page.