CS305 · Computer Networking

Assignment 1

何泽安 (He Zean)
12011323
Mar. 12, 2022

Q. 1

The first point is, there are many participations in the network, take the application layer as an example, where developers (in different companies) need to work together to maintain and adhere to several protocols that all devices support and can understand (e.g. both Chrome and Edge should be able to render the same .http file correctly) – without uniform standards, cross-software interaction will become far more difficult than it is today.

Secondly, the network is considered as an extremely complex system, that people have to layering and highly abstracting the system. The protocols here can be viewed as interfaces in Java, which means people can easily change the implementation of a specific problem / layer, only needs to maintain its functions behave same, as expected and specified in the protocols. More detailed explanation see Protocol Layering *(Computer Networking: A Top-Down Approach, 6 ed., pp. 49)*.

Q. 2

Application Layer	Resides network applications & their protocols (letting developer easier to use the services provided by lower layers).
Transport Layer	Provides *logic communication* for application processes running on different hosts (a.k.a. transports application-layer messages between application endpoints).
Network Layer	Moves network-layer packets (datagrams) from one host to another.
Link Layer	Routes a datagram through a series of routers between the source and destination.
Physical Layer	Move the individual bits within the frame from one node to the next.

Q. 3

Router Physical layer, link layer, network layer.

Link-layer Switch Physical layer, link layer.

Host Physical layer, link layer, network layer, transport layer, application layer.

Q. 4

(a) $d_{prop} = \frac{m}{s}\text{sec}$

(b) $d_{trans} = \frac{L}{R}\text{sec}$

(c) $d_{nodal} = \cancel{d_{proc}} + \cancel{d_{queue}} + d_{trans} + d_{prop} = (\frac{m}{s} + \frac{L}{R})\text{sec}$

(d) It was just pushed (transmitted) into the link, a.k.a. the very starting place of the link between two hosts.

(e) In the link, on its way to Host B. Host A $\overset{(Ls/R)\text{m}}{\longrightarrow}$ first bit $\longrightarrow$ Host B

(f) The first bit has already been received by Host B.

(g)

$$d_{prop} = m/s$$

$$d_{trans} = L/R$$

$$m = Ls/R = \frac{2.5 \cdot 10^8 \text{m/s} \times 120\text{bit}}{56 \cdot 10^3 \text{bit/s}} = 535.7\text{km}$$

Q. 5

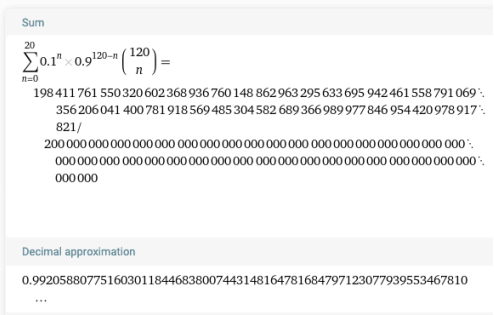
(a) No matter the user uses the resource or not, circuit switching must divide the link (FDM, TDM... whatever) reserve for all users all the time.

$$\frac{3\text{Mbps}}{150\text{kbps}} = 20\text{max support \#users}$$

(b) $p = 0.1$, as the mentioned in the question.

(c) $p_n = C_{120}^n \, 0.1^n \, 0.9^{120-n}$

(d) $p_{21+} = 1 - \Sigma_{n=0}^{20} \, C_{120}^n \, 0.1^n \, 0.9^{120-n} \approx 0.008$



Q. 6

As protocols for webpage and email, they are less care about the transmission efficiency, since they are no that time-sensitive, and in general, the file size is not very large. This make the "only" benefit of UDP (rather than TCP, its high efficiency and low payload for server) seems not that important for these scenarios. In another hand, these content must be highly assured correctness – once a bit gets wrong, the whole content may change its meaning or cannot be correctly parsed or rendered, which relies on the error control mechanism of TCP.

Q. 7

(a) F **(b)** T **(c)** F **(d)** F **(e)** F

Q. 8

```
GET /cs453/index.html HTTP/1.1
Host: gaia.cs.umass.edu
User-Agent: Mozilla/5.0 (Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gecko/20040804 Netscape/7.2 (ax)
Accept:ext/xml, application/xml, application/xhtml+xml, text/html;q=0.9, text/plain;q=0.8, image/png,*/*;q=0
Accept-Language: en-us, en;q=0.5
AcceptEncoding: zip, deflate
Accept-Charset: ISO -8859-1, utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection:keep-alive
<cr><lf>
```

(a) The host is *gaia.cs.umass.edu*, while the file path is */cs453/index.html*. Also we know it uses http instead of https, since wireshark can only parse the https-packages to the TLS layer. Summary:

```
http://gaia.cs.umass.edu/cs453/index.html
```

(b) Trivially, HTTP 1.1, see the HTTP/1.1 header.

(c) Persistent, see the header *Connection: keep-alive*

(d) The application layer doesn't include this information, you need to check the network layer.

(e) The browser "pretends that" it is *Mozilla/5.0*. Note that Mozilla is actually no more existing, however, for historical reasons such as market competition, many browsers now use the Mozilla agent name[1]. Actually this browser is Netscape browser running on Windows XP[2].

The reason for putting the browser type in the message is that, different browsers use different render engine, and may / may not support several frameworks. Therefore, the developers of websites often need to prepare multiple versions for the same object (written in different framework, etc.), and give the user a proper version base on the agent-name.

Q. 9

```
HTTP/1.1 200 OK
Date: Tue, 07 Mar 2008 12:39:45GMT
Server: Apache/2.0.52 (Fedora)
Last-Modified: Sat, 10 Dec2005 18:27:46 GMT
ETag: " 526c3-f22-a88a4c80"
AcceptRanges: bytes
Content-Length: 3874
Keep-Alive: timeout=max=100
Connection: Keep-Alive
Content-Type: text/html; charset= ISO-8859-1

<!doctype html public " //w3c//dtd html 4.0 transitional//en" >
<html>
 <head>
  <meta http-equiv=" Content-Type"  content=" text/html; charset=iso-8859-1" >
```

[1]为什么所有主要浏览器的 User-Agent 都是 Mozilla/x.0 开头？ (https://www.zhihu.com/question/19553117)

[2]https://user-agents.net/string/mozilla-5-0-windows-u-windows-nt-5-1-en-us-rv-1-7-2-gecko-20040804-netscape-7-2-ax

```
16      <meta name=" GENERATOR"  content=" Mozilla/4.79 [en] (Windows NT 5.0; U) Netscape]" >
17       <title>CMPSCI 453 / 591 / NTU-ST550ASpring 2005 homepage</title>
18    </head>
19  <!-- ... -->^^I
```

(a) Yes, the server successfully handled the request and responded with status code *200 OK*. As the *Date* header mentioned, the document reply was provided at Sat, 10 Dec2005 18:27:46 GMT.

(b) Sat, 10 Dec2005 18:27:46 GMT. See the *Last-Modified* header.

(c) 3874 bytes. See the *Content-Length* header, also note the header *AcceptRanges: bytes*.

(d) **<!doc**

Yes, see the *Connection* header which indicates the client to keep-alive.