**Chapter 2 Review Questions**

**SECTION 2.1**

**R1. List five nonproprietary Internet applications and the application-layer protocols that they use.**

The Web: HTTP; file transfer: FTP; remote login: Telnet; e-mail: SMTP; BitTorrent file sharing: BitTorrent protocol

**R2. What is the difference between network architecture and application architecture?**

Network architecture refers to the organization of the communication process into layers (e.g., the five-layer Internet architecture). Application architecture, on the other hand, is designed by an application developer and dictates the broad structure of the application (e.g., client-server or P2P).

**Network Architecture** → The overall structure and layers that define how data moves from one device to another (e.g., OSI Model, TCP/IP Model).  
**Application Architecture** → The structure and rules for how a specific application works and exchanges data between client and server (e.g., HTTP, FTP, SMTP).

📌 In one line:

* **Network Architecture** → The design of the “road” for sending data.
* **Application Architecture** → The “rules” for running a specific app.

**R3. For a communication session between a pair of processes, which process is**

**the client and which is the server?**

The process which initiates the communication is the client; the process that waits to be contacted is the server.

**R4. For a P2P file-sharing application, do you agree with the statement, “There is no notion of client and server sides of a communication session”? Why or why not?**

No. In a P2P file-sharing application, the peer that is receiving a file is typically the client and the peer that is sending the file is typically the server.

**R5. What information is used by a process running on one host to identify a process running on another host?**

The IP address of the destination host and the port number of the socket in the destination process.

**R6. Suppose you wanted to do a transaction from a remote client to a server as**

**fast as possible. Would you use UDP or TCP? Why?**

You would use UDP. With UDP, the transaction can be completed in one roundtrip time (RTT) - the client sends the transaction request into a UDP socket, and the server sends the reply back to the client's UDP socket. With TCP, a minimum of two RTTs are needed - one to set-up the TCP connection, and another for the client to send the request, and for the server to send back the reply.

**R7. Referring to Figure 2.4, we see that none of the applications listed in Figure**

**2.4 requires both no data loss and timing. Can you conceive of an application**

**that requires no data loss and that is also highly time-sensitive?**

One such example is remote word processing, for example, with Google docs. However, because Google docs runs over the Internet (using TCP), timing guarantees are not provided.

**R8. List the four broad classes of services that a transport protocol can provide.**

**For each of the service classes, indicate if either UDP or TCP (or both) provides such a service.**

a) Reliable data transfer TCP provides a reliable byte-stream between client and server but UDP does not.

b) A guarantee that a certain value for throughput will be maintained Neither

c) A guarantee that data will be delivered within a specified amount of time Neither

d) Confidentiality (via encryption) Neither

**R9. Recall that TCP can be enhanced with TLS to provide process-to-process**

**security services, including encryption. Does TLS operate at the transport**

**layer or the application layer? If the application developer wants TCP to be**

**enhanced with TLS, what does the developer have to do?**

SSL operates at the application layer. The SSL socket takes unencrypted data from the application layer, encrypts it and then passes it to the TCP socket. If the application developer wants TCP to be enhanced with SSL, she has to include the SSL code in the application.

প্রশ্নটা basically দুইটা জিনিস জানতে চাইছে –

1. **TLS আসলে কোন layer-এ কাজ করে** — Transport layer নাকি Application layer?
   * TCP তো transport layer-এর protocol। TLS (Transport Layer Security) নাম শুনে মনে হতে পারে transport layer-এ কাজ করে, কিন্তু আসলে TLS OSI model-এর **application layer আর transport layer-এর মাঝে** কাজ করে। Networking-এর দৃষ্টিকোণ থেকে TLS-কে সাধারণত application layer-এর অংশ ধরা হয়, কারণ অ্যাপ্লিকেশন লেভেলেই TLS handshake, certificate verification ইত্যাদি হয়।
2. **TCP-কে TLS দিয়ে enhance করতে চাইলে ডেভেলপারকে কী করতে হবে** —
   * মানে, developer যদি চান TCP communication encrypted হোক, তাহলে তাকে অ্যাপ্লিকেশনের কোডে TLS/SSL library (যেমন OpenSSL, Java Secure Socket Extension, Python-এর ssl module ইত্যাদি) ব্যবহার করতে হবে।
   * ডেভেলপারকে TCP socket-এর ওপর TLS "wrap" করতে হয়, যাতে ডাটা পাঠানোর আগে encrypt হয় আর রিসিভ করার পরে decrypt হয়।
   * অর্থাৎ, এটি TCP-এর ভিতরের কাজ নয়, বরং TCP connection establish হওয়ার পরে application-level code TLS ব্যবহার করে secure channel তৈরি করে।

সারকথা:

* TLS → application layer-এ operate করে (transport layer-এর ঠিক উপরে)।
* TCP-এর সাথে TLS ব্যবহার করতে হলে ডেভেলপারকে অ্যাপ্লিকেশনে TLS library integrate করতে হবে।

**SECTIONS 2.2–2.5**

**R10. What is meant by a handshaking protocol?**

A protocol uses handshaking if the two communicating entities first exchange control packets before sending data to each other. SMTP uses handshaking at the application layer whereas HTTP does not.

**R11. Why do HTTP, SMTP, and IMAP run on top of TCP rather than on UDP?**

The applications associated with those protocols require that all application data be received in the correct order and without gaps. TCP provides this service whereas UDP does not.

কারণগুলো একে একে বলি —

**1. Reliable data transfer দরকার**

* TCP গ্যারান্টি দেয় যে ডাটা ঠিকভাবে, সঠিক ক্রমে পৌঁছাবে (error-free, in-order delivery)।
* HTTP, SMTP, IMAP—সবগুলোতেই ডাটা (যেমন ওয়েবপেজ, ইমেইল) সঠিকভাবে পুরোপুরি না পৌঁছালে কমিউনিকেশন ভেঙে যাবে।
* UDP-তে এই গ্যারান্টি নেই — প্যাকেট হারিয়ে গেলে বা ক্রম বদলে গেলে অ্যাপ্লিকেশন নিজেকে সামলাতে হয়, যা জটিল।

**2. Connection-oriented communication দরকার**

* TCP একটি **connection-oriented** প্রোটোকল — প্রথমে সংযোগ (handshake) তৈরি হয়, তারপর ডাটা আদান-প্রদান হয়।
* HTTP-তে ক্লায়েন্ট ও সার্ভারকে একই কনটেক্সটে রিকোয়েস্ট-রেসপন্স বিনিময় করতে হয়।
* SMTP আর IMAP-এও ইমেইল সার্ভারগুলোর মধ্যে একাধিক ধাপে কমান্ড-রেসপন্স হয়, যেটার জন্য স্থায়ী সংযোগ দরকার।

**3. Flow control ও congestion control**

* TCP স্বয়ংক্রিয়ভাবে sender ও receiver-এর মধ্যে ডাটা ট্রান্সমিশন স্পিড অ্যাডজাস্ট করে।
* UDP-তে এই ব্যবস্থা নেই, ফলে নেটওয়ার্কে কনজেশন বা ডাটা লস বেশি হতে পারে।

**4. Large message handling**

* ইমেইল বা ওয়েবপেজ অনেক বড় ডাটা ট্রান্সফার করতে পারে, যা UDP-তে ফ্র্যাগমেন্টেশন ইস্যু তৈরি করে।
* TCP স্বয়ংক্রিয়ভাবে ডাটা ভাগ করে পাঠায় এবং রিসিভারে আবার জোড়া লাগায়।

**R12. Consider an e-commerce site that wants to keep a purchase record for each of its customers. Describe how this can be done with cookies.**

When the user first visits the site, the server creates a unique identification number, creates an entry in its back-end database, and returns this identification number as a cookie number. This cookie number is stored on the user’s host and is managed by the browser. During each subsequent visit (and purchase), the browser sends the cookie number back to the site. Thus the site knows when this user (more precisely, this browser) is visiting the site.

**R13. Describe how Web caching can reduce the delay in receiving a requested**

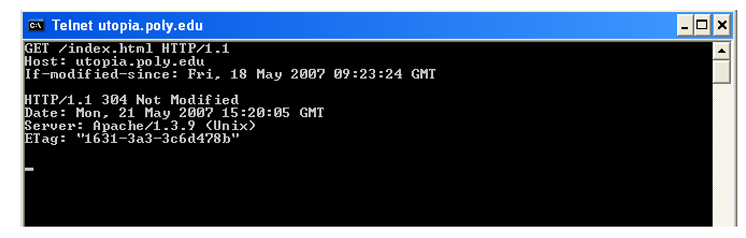
**object. Will Web caching reduce the delay for all objects requested by a user**

**or for only some of the objects? Why?**

Web caching can bring the desired content “closer” to the user, possibly to the same LAN to which the user’s host is connected. Web caching can reduce the delay for all objects, even objects that are not cached, since caching reduces the traffic on links.

**R14. Telnet into a Web server and send a multiline request message. Include in the request message the If-modified-since: header line to force a response message with the 304 Not Modified status code.**

Telnet is not available in Windows 7 by default. to make it available, go to Control Panel, Programs and Features, Turn Windows Features On or Off, Check Telnet client. To start Telnet, in Windows command prompt, issue the following command > telnet webserverver 80 where "webserver" is some webserver. After issuing the command, you have established a TCP connection between your client telnet program and the web server. Then type in an HTTP GET message. An example is given below:

****

Since the index.html page in this web server was not modified since Fri, 18 May 2007 09:23:34 GMT, and the above commands were issued on Sat, 19 May 2007, the server returned "304 Not Modified". Note that the first 4 lines are the GET message and header lines inputed by the user, and the next 4 lines (starting from HTTP/1.1 304 Not Modified) is the response from the web server.

**R15. List several popular messaging apps. Do they use the same protocols as SMS?**

**R16. Suppose Alice, with a Web-based e-mail account (such as Hotmail or Gmail),**

**sends a message to Bob, who accesses his mail from his mail server using**

**IMAP. Discuss how the message gets from Alice’s host to Bob’s host. Be**

**sure to list the series of application-layer protocols that are used to move the**

**message between the two hosts.**

**R17. Print out the header of an e-mail message you have recently received. How**

**many Received: header lines are there? Analyze each of the header lines**

**in the message.**

**R18. What is the HOL blocking issue in HTTP/1.1? How does HTTP/2 attempt to**

**solve it?**

**R19. Is it possible for an organization’s Web server and mail server to have**

**exactly the same alias for a hostname (for example, foo.com)? What would**

**be the type for the RR that contains the hostname of the mail server?**

**R20. Look over your received e-mails, and examine the header of a message sent**

**from a user with a .edu e-mail address. Is it possible to determine from the**

**header the IP address of the host from which the message was sent? Do the**

**same for a message sent from a Gmail account.**

**Chapter 2 Problems**

P1. True or false?

**a.** *A user requests a Web page that consists of some text and three images. For this page, the client will send one request message and receive four response messages.*

* **False** ✅
  + For each object (HTML file + 3 images), the browser sends a **separate HTTP request** and gets a separate response.
  + That’s **4 request messages** and **4 response messages**, not 1 request and 4 responses.

**b.** *Two distinct Web pages (e.g., www.mit.edu/research.html and* [*www.mit.edu/students.html*](http://www.mit.edu/students.html)*) can be sent over the same persistent connection.*

* **True** ✅
  + With **persistent HTTP (HTTP/1.1 default)**, multiple requests/responses can be sent sequentially over the same TCP connection.

**c.** *With nonpersistent connections between browser and origin server, it is possible for a single TCP segment to carry two distinct HTTP request messages.*

* **False** ✅
  + In **nonpersistent HTTP**, each TCP connection handles **only one request-response pair**, then closes. So one TCP segment can’t have two different requests.

**d.** *The Date: header in the HTTP response message indicates when the object in the response was last modified.*

* **False** ✅
  + Date: indicates **when the response message was generated by the server**.
  + The header that indicates last modification time is Last-Modified:.

**e.** *HTTP response messages never have an empty message body.*

* **False** ✅
  + Many responses have no body (e.g., status codes **204 No Content**, **304 Not Modified**, or HEAD request responses).

Problem 2

SMS, iMessage, WeChat, and WhatsApp are all smartphone real-time messaging systems. After doing some research on the Internet, for each of these systems write one paragraph about the protocols they use. Then write a paragraph explaining how they differ.

### Answer:

**SMS (Short Message Service)**  
SMS is a cellular network service that enables sending short text messages between mobile devices. It relies on the signaling channels of the GSM (Global System for Mobile Communications) or other cellular networks, specifically using the SS7 (Signaling System No. 7) protocol for message delivery. SMS messages are sent via the cellular carrier's infrastructure and do not require internet connectivity. The protocol is store-and-forward based, with messages routed through the Short Message Service Center (SMSC) before reaching the recipient.

**iMessage**  
Apple’s iMessage is a proprietary messaging service that runs over the Internet using Apple's own protocols. It uses a combination of **XMPP (Extensible Messaging and Presence Protocol)** as a base messaging protocol with Apple-specific extensions for encryption and delivery. Messages are end-to-end encrypted using the **Apple Push Notification Service (APNS)** to wake devices and handle message delivery, and TLS for secure transport. iMessage requires an Apple ID and internet connectivity.

**WeChat**  
WeChat uses a proprietary protocol developed by Tencent that runs over the internet on top of TCP and HTTP/2. The app uses a custom messaging protocol for chat, voice, and video calls. For encryption, WeChat uses a combination of TLS for transport security and additional application-layer encryption for message content. It integrates with Tencent’s server infrastructure for message routing and delivery, supporting offline storage and multi-device sync.

**WhatsApp**  
WhatsApp uses the **XMPP protocol** for message exchange but with custom extensions tailored for reliability, presence, and end-to-end encryption. The key security feature is the **Signal Protocol**, which provides forward secrecy and end-to-end encryption for all message types. Messages are sent over TCP secured by TLS. WhatsApp requires internet connectivity and uses WhatsApp servers to relay messages when direct peer-to-peer communication is unavailable.

Problem 3

**Question:**  
Consider an HTTP client that wants to retrieve a Web document at a given URL. The IP address of the HTTP server is initially unknown. What transport and application-layer protocols besides HTTP are needed in this scenario?

**Easy Answer:**  
Besides HTTP, the client needs:

* **Application-layer protocol:**  
  **DNS (Domain Name System)** — to translate the website’s domain name (like www.example.com) into its IP address.
* **Transport-layer protocol:**  
  **TCP (Transmission Control Protocol)** — to create a reliable connection between the client and the web server for HTTP communication.

So, the process is:

1. Use **DNS** to get the IP address from the domain name.
2. Use **TCP** to connect to that IP address on port 80 (for HTTP).
3. Then use **HTTP** to request the web page.

Problem 4

Got it! Let’s answer each question by looking at the HTTP GET message you shared:

**a. What is the URL of the document requested by the browser?**

* From the first line:
* GET /cs453/index.html HTTP/1.1
* The requested **path** is /cs453/index.html.
* From the Host header:
* Host: gaia.cs.umass.edu
* So, the full URL is:  
  [**http://gaia.cs.umass.edu/cs453/index.html**](http://gaia.cs.umass.edu/cs453/index.html)

**b. What version of HTTP is the browser running?**

* From the first line again:
* GET /cs453/index.html HTTP/1.1
* The browser is using **HTTP version 1.1**.

**c. Does the browser request a non-persistent or a persistent connection?**

* Look at the Connection header:
* Connection: keep-alive
* The value keep-alive means the browser is requesting a **persistent connection** (the TCP connection will stay open for multiple requests/responses).

**d. What is the IP address of the host on which the browser is running?**

* **This information is NOT present in the HTTP GET message.**
* The client’s IP address is part of the TCP/IP packet headers, not the HTTP headers or message body.
* So, **you cannot find the client IP in this HTTP GET request**.

**e. What type of browser initiates this message? Why is the browser type needed in an HTTP request message?**

* From the User-Agent header:
* User-Agent: Mozilla/5.0 (Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gecko/20040804 Netscape/7.2 (ax)
* The browser identifies itself as **Netscape 7.2 (Mozilla/5.0) on Windows NT 5.1**.
* **Why is this needed?**
  + Servers use the User-Agent string to tailor content or behavior based on the browser’s capabilities and quirks.
  + For example, the server might send different HTML or scripts to different browsers to ensure proper display and compatibility.

If you want, I can help you write these answers in a neat format for your assignment!

Problem 5

a) The status code of 200 and the phrase OK indicate that the server was able to locate

the document successfully. The reply was provided on Tuesday, 07 Mar 2008

12:39:45 Greenwich Mean Time.

b) The document index.html was last modified on Saturday 10 Dec 2005 18:27:46 GMT.

c) There are 3874 bytes in the document being returned.

d) The first five bytes of the returned document are : <!doc. The server agreed to a

persistent connection, as indicated by the Connection: Keep-Alive field

Problem 6

a) Persistent connections are discussed in section 8 of RFC 2616 (the real goal of this

question was to get you to retrieve and read an RFC). Sections 8.1.2 and 8.1.2.1 of

the RFC indicate that either the client or the server can indicate to the other that it is

going to close the persistent connection. It does so by including the connection-token

"close" in the Connection-header field of the http request/reply.

b) HTTP does not provide any encryption services.

c) (From RFC 2616) “Clients that use persistent connections should limit the number of

simultaneous connections that they maintain to a given server. A single-user client

SHOULD NOT maintain more than 2 connections with any server or proxy.”

d) Yes. (From RFC 2616) “A client might have started to send a new request at the same

time that the server has decided to close the "idle" connection. From the server's point

Downloaded by TuiMor Sala (tuimorsala01@gmail.com)

of view, the connection is being closed while it was idle, but from the client's point of

view, a request is in progress.”

Problem 7

The total amount of time to get the IP address is

n

RTT RTT RTT + + +  2 1

.

Once the IP address is known, O

RTT elapses to set up the TCP connection and another

O

RTT elapses to request and receive the small object. The total response time is

n o

RTT RTT RTT RTT + + + +  2 1

2

Problem 8

a)

o o n

RTT RTT RTT RTT 2 8 2 1

 + + + +

n o

RTT RTT RTT + + + =  1

18 .

b)

o o n

RTT RTT RTT RTT 2 2 2 1

 + + + +

n o

RTT RTT RTT + + + =  1

6

c) Persistent connection with pipelining. This is the default mode of HTTP.

o o n

RTT RTT RTT RTT + + + + 2 1



n o

RTT RTT RTT + + + =  1

3 .

Persistent connection without pipelining, without parallel connections.

o o n

RTT RTT RTT RTT 8 2 1

+ + + +

n o

RTT RTT RTT + + + =  1

10 .

Problem 9

a) The time to transmit an object of size L over a link or rate R is L/R. The average time

is the average size of the object divided by R:

 = (850,000 bits)/(15,000,000 bits/sec) = .0567 sec

The traffic intensity on the link is given by =(16 requests/sec)(.0567 sec/request) =

0.907. Thus, the average access delay is (.0567 sec)/(1 - .907)  .6 seconds. The total

average response time is therefore .6 sec + 3 sec = 3.6 sec.

Downloaded by TuiMor Sala (tuimorsala01@gmail.com)

lOMoARcPSD|19239111

lOMoARcPSD|19239111

b) The traffic intensity on the access link is reduced by 60% since the 60% of the

requests are satisfied within the institutional network. Thus the average access delay

is (.0567 sec)/[1 – (.4)(.907)] = .089 seconds. The response time is approximately

zero if the request is satisfied by the cache (which happens with probability .6); the

average response time is .089 sec + 3 sec = 3.089 sec for cache misses (which

happens 40% of the time). So the average response time is (.6)(0 sec) + (.4)(3.089 sec)

= 1.24 seconds. Thus the average response time is reduced from 3.6 sec to 1.24 sec.

Problem 10

Note that each downloaded object can be completely put into one data packet. Let Tp

denote the one-way propagation delay between the client and the server.

First consider parallel downloads using non-persistent connections. Parallel downloads

would allow 10 connections to share the 150 bits/sec bandwidth, giving each just 15

bits/sec. Thus, the total time needed to receive all objects is given by:

(200/150+Tp + 200/150 +Tp + 200/150+Tp + 100,000/150+ Tp )

+ (200/(150/10)+Tp + 200/(150/10) +Tp + 200/(150/10)+Tp + 100,000/(150/10)+ Tp )

= 7377 + 8\*Tp (seconds)

Now consider a persistent HTTP connection. The total time needed is given by:

(200/150+Tp + 200/150 +Tp + 200/150+Tp + 100,000/150+ Tp )

+ 10\*(200/150+Tp + 100,000/150+ Tp )

=7351 + 24\*Tp (seconds)

Assuming the speed of light is 300\*106 m/sec, then Tp=10/(300\*106)=0.03 microsec. Tp

is therefore negligible compared with transmission delay.

Thus, we see that persistent HTTP is not significantly faster (less than 1 percent) than the

non-persistent case with parallel download.

Problem 11

a) Yes, because Bob has more connections, he can get a larger share of the link

bandwidth.

b) Yes, Bob still needs to perform parallel downloads; otherwise he will get less

bandwidth than the other four users.