CS339 Computer Networks Chapter 2 -- Application Layer

1. Two Architecture Models

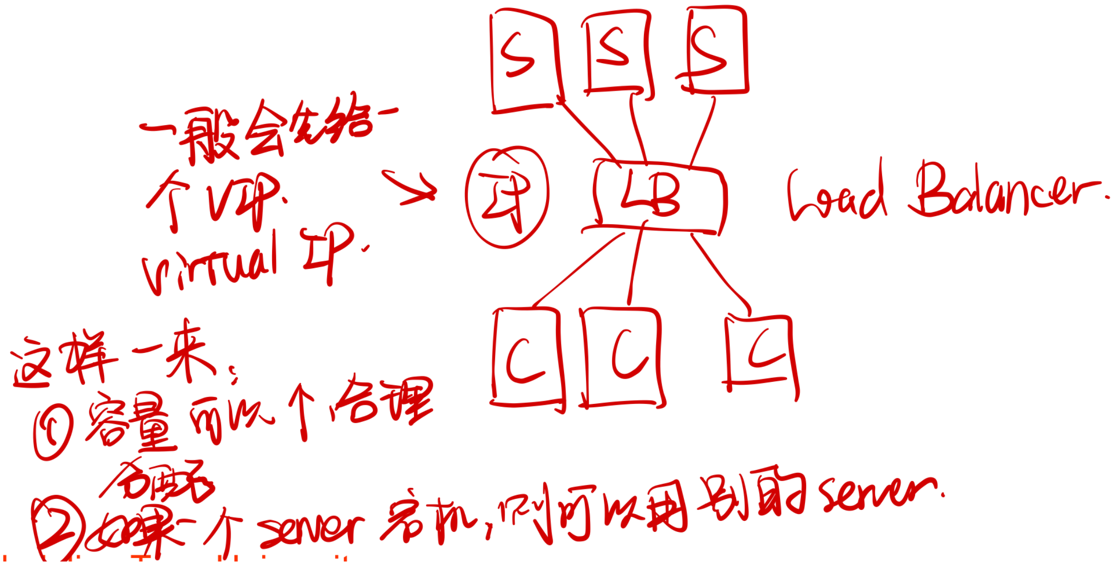
We talk again about the two kinds of Architecture Models:

1. Client-server model: There are always-on servers. The clients do not directly communicate with each other.
2. P2P model: There is no always-on server. Arbitrary end systems directly communicate with each other.

Conclusion: The P2P model makes better use of the upwards bandwidth of end systems. And the pressure on the downwards bandwidth of the host will be distributed. Thus, P2P is more cost effective, highly scalable, but it is also more difficult to manage.

2. Load Balancer for Servers

Just in case we are interested. The load balancer of a data center will balance all the loads to different servers. It will distribute the capacity appropriately. Also, it will ensure the user will switch to another server if one of the server is down.



一般会先给一个VIP. virtual IP.

这样一来，
①容量可以个合理分配
②如果一个server宕机，则可以用别的server.

3. Skype Use Hybrid Model

The clients ask the server for its IP address and its buddies' IP address. Connection is then built based on client-client connection.

4. Socket

The socket provides API from transport layer to application layer. Since the transport layer is controlled by OS, the socket actually offer users some freedom to control some properties.

5. Transport Service Required by Applications

There are three major service properties that applications concern:

1. Data Loss: Some application allows no data loss, like e-mail
2. Delay: Some application allows no delay, like live
3. Throughput: Some application allows no small throughput, like downloading a movie.
4. Security (not major concern)

6. Two Major Services Provided by Transport Layer

6.1 TCP

Feature: Connection based

- Reliable Data Transfer: Guarantee the data will be received by the receiver
- Congestion Control: Avoid fulfilling the network, which results in packet loss
- Flow Control: Avoid sending too fast and the receiver cannot receive the packets, which results in packet loss.

6.2 UDP

Feature: Packet-switching based

- Unreliable Data Transfer

7. Process Indentifier

To determine the process, which means the process should be uniquely marked, we use

$$\text{process identifier} = \text{IP address} + \text{Port Number (e.g. } 202.120.2.102 + 80)$$

8. UDP Based Application Protocol

DNS! DNS is the representative application protocol using UDP as service.

9. DNS

DNS provides the map between name and its IP address:

$$\text{www.sjtu.edu.cn} \rightarrow 202.120.2.119$$

9.1 Use UDP!

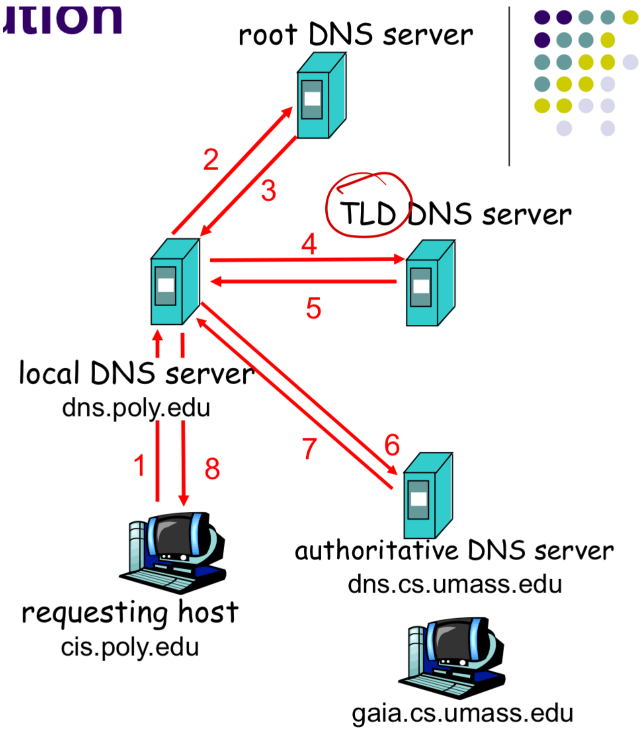
9.2 How to Construct

Add domain suffix to the end of the name: like adding `cn` and then adding `edu`

9.3 DNS Name Resolution

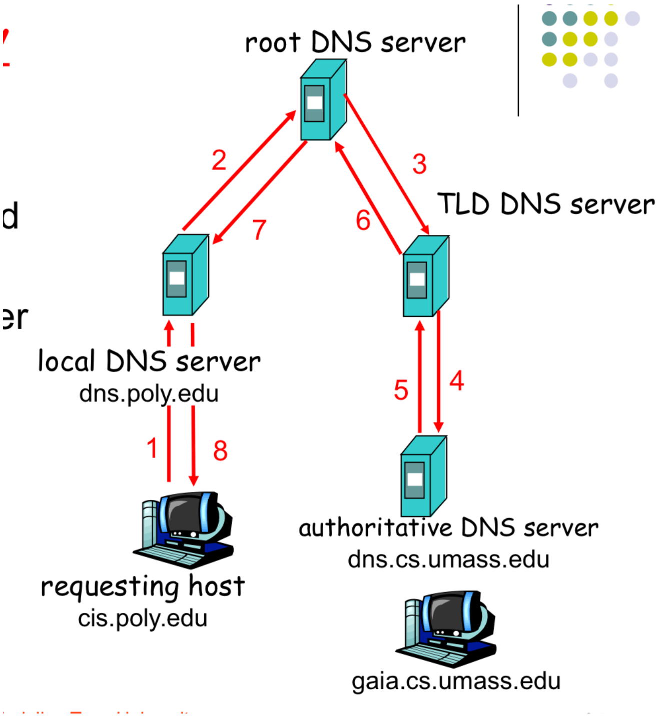
Iterated Query

This means that the local DNS server will keep asking different levels of DNS server and it will finally ask the right server. The responsibility is always kept by the local DNS server.



Recursive Query

This means that the local DNS server will ask the root DNS server, and it will be the root DNS server's responsibility to get the right IP back. Then the root DNS server will ask TLD DNS server, and it becomes TLD DNS server's responsibility...



10. HTTP

10.1 World Wide Web

It uses client-server model, sets TCP connection and then send HTTP request/responce.

10.2 HTTP

HTTP used TCP

It first builds TCP connection, and then send HTTP messages. Finally the TCP closed and the session comes to an end.

HTTP is Stateless

This means that HTTP maintains no message about the past client requests. It will forget everything. This is for simplicity. If we need to store some historical information (like TaoBao does), we need to use **Cookies**.

11. Page Load Time

11.1 Round Trip Time (RTT)

RTT is one of the most important time scale defined to measure the performance of Internet. It is the time for a small packet to travels from sender to the receiver and goes back.

11.2 Simple Case

If we consider there is only one HTML object to be transmitted, then constructing the TCP connection uses one RTT, using HTTP to request the content uses one RTT, and transmitting the content uses some time. Thus, the formula can be written as:

$$\text{Page Load Time} = 2 \times \text{RTT} + \text{Transmit time}$$

11.3 Complex Case

If there are more than 1 HTML object to be transmitted, always repeating the procedure above is not efficient, we can usually transmit several HTML objects concurrently. For example, we can transmit 3 HTML objects at the same time, and there are 5 HTML objects to be transmitted.

1. Then we first use one RTT to construct the TCP connection, then use one RTT to request the HTML file. It then takes some time to transmit this HTML file.
2. We then use one RTT to construct another TCP connection, then use one RTT to request the HTML files. It then takes some time (three times than that in 1) to transmit these three files.
3. Repeat 2 for another 2 files.

$$\text{Total Page Load Time} = 3 \times 2 \times \text{RTT} + 6\sigma \text{ (object transmitting time)}$$

12. FTP

12.1 Use Client-server Model

The client will transfer file to/from the remote server.

12.2 Use TCP

- FTP uses two TCP connections to communicate with the TCP server. One connection is in charge of sending control message and the other connection is in charge of sending data.
- Using TCP means that FTP will first set up a TCP connection for every file. When that file is completely transmitted, the TCP connection will be disconnected.
- Maintain a "state" message.

13. Email

13.1 Two Class of Protocols

Push Protocol

In Email, there is only one push protocol -- SMTP. It pushes the mail to the sender's server, or pushes the mail from sender's server to receiver's server.

Pull Protocol

POP,IMAP,HTTP. These protocols pull message from the receiver's email server.

14. Peer to Peer Model

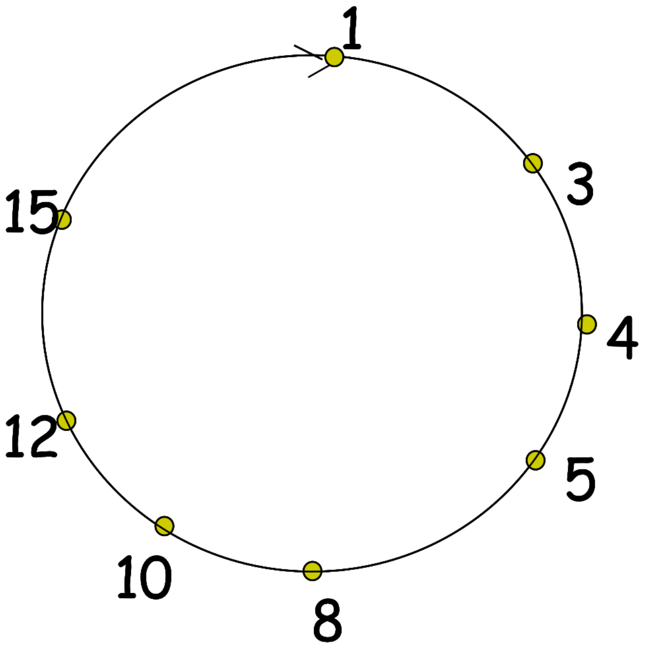
14.1 Distributed Hash Table

To identify each node together with the content it has, we use a key-value pair to represent such a node in a peer-to-peer network.

This is implemented using hash function:

$$\text{key} = \text{hash}(\text{Content Keywords})$$
$$\text{value} = \text{hash}(\text{IP address, Port Number})$$

14.2 Circular DHT



In this circular DHT model, each node is assigned an ID by hash table, and it **only** knows which node is the next node. Thus, finding the corresponding node number of given content takes

$$\mathcal{O}(n)$$

in worst case. For example, let's assume we need to find the node with largest key and the nodes are coincidentally hashed in increasing order (assume this case for following discussion).

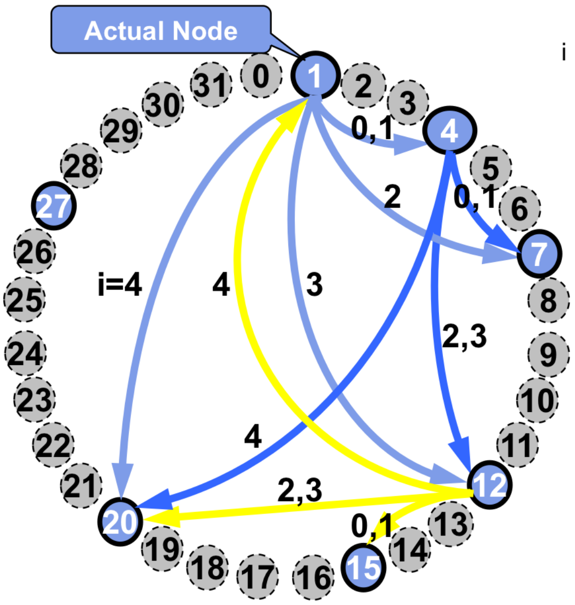
To improve the efficiency, each node can perform the lookup with shortcuts. That is, at probing `i` it probes the

$$1 + 2^i \text{ node after this node}$$

and then record the node in **finger table**. Thus, it takes

$$\mathcal{O}(\log n)$$

time to get the target node in worst case. The diagram for shortcuts is shown as below:



$1+2^0$ $4+2^0$ $1+2^1$ $4+2^1$

i	Noe 1		Node 4		Node 7		Node 12	
0	2	4	5	7	8	12	13	15
1	3	4	6	7	9	12	14	15
2	5	7	8	12	11	12	16	20
3	9	12	12	12	15	15	20	20
4	17	20	20	20	23	27	28	1

i	Node 15		Node 20		Node 27	
0	16	20	21	27	28	1
1	17	20	22	27	29	1
2	19	20	24	27	31	1
3	23	27	28	1	3	4
4	31	1	4	4	11	12

14.3 When Node Leaves/Join

This is easy, just tell the nodes before and let it update its finger table.

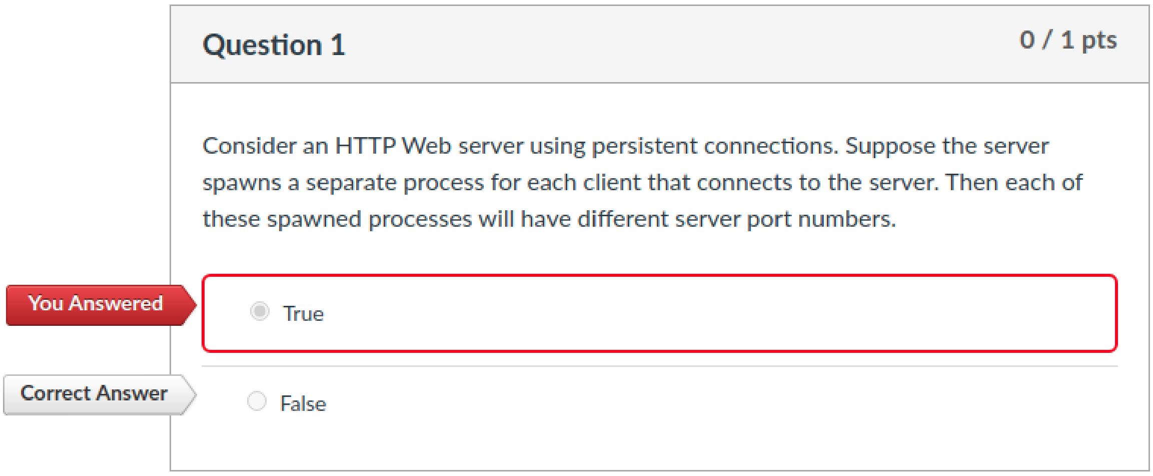
15. Tit-for-Tat Principle

We can slow our upload to the target who contributes little, so that they will have more bandwidth for the uploading part. In this way, we can encourage the contribution of the target. The `tit-for-tat` principle describes a strategy that, you should favor the peers that upload to you rapidly, which means that you should provide the peers who transmit the file to you with high rate with a high rate transmission reciprocally, in order to prevent those peers from thinking you are not "active" and then chock you. On the other hand, you should slow your upload to those peers who are not "active" transmitting file to you, so that they are "chocked" and encouraged to contribute faster.

16. UDP Holing

Knowing what is this is enough. It just enables hosts behind NATs to talk with each other.

Homework and Quiz Review

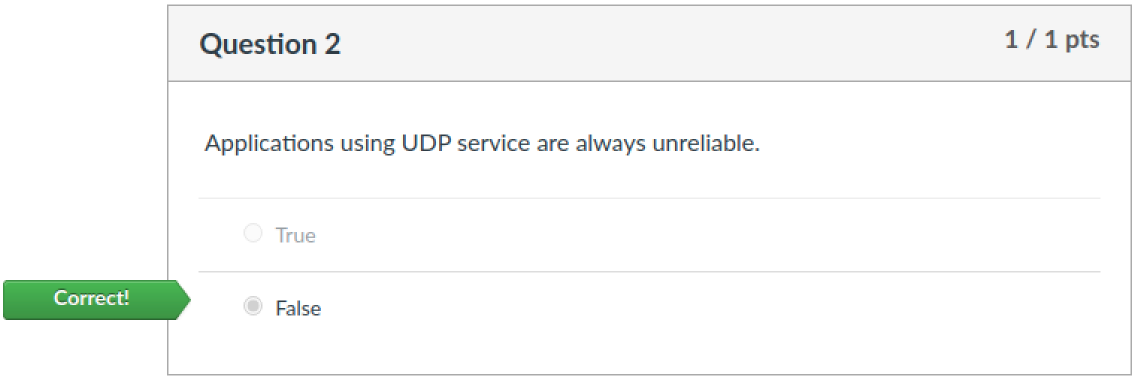

Question 1 0 / 1 pts

Consider an HTTP Web server using persistent connections. Suppose the server spawns a separate process for each client that connects to the server. Then each of these spawned processes will have different server port numbers.

You Answered ● True

Correct Answer ○ False

Explain: A TCP socket is defined as a four-tuple `(source IP, source port, destination IP, destination port)`, since the processes are all HTTP, they will use the same source port number `80`, but they will use different `(destination IP, destination port)` pair.

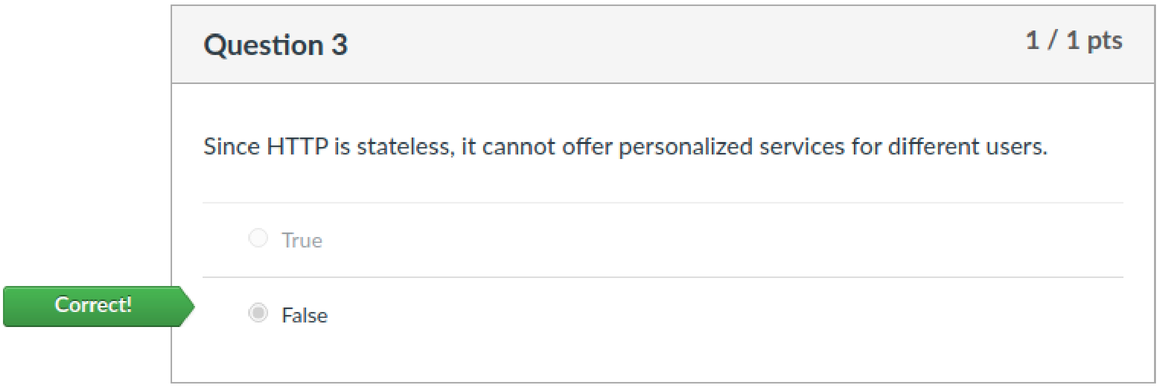

Question 2 1 / 1 pts

Applications using UDP service are always unreliable.

○ True

Correct! ● False

Explain: We can add some application layer features to make UDP service reliable.

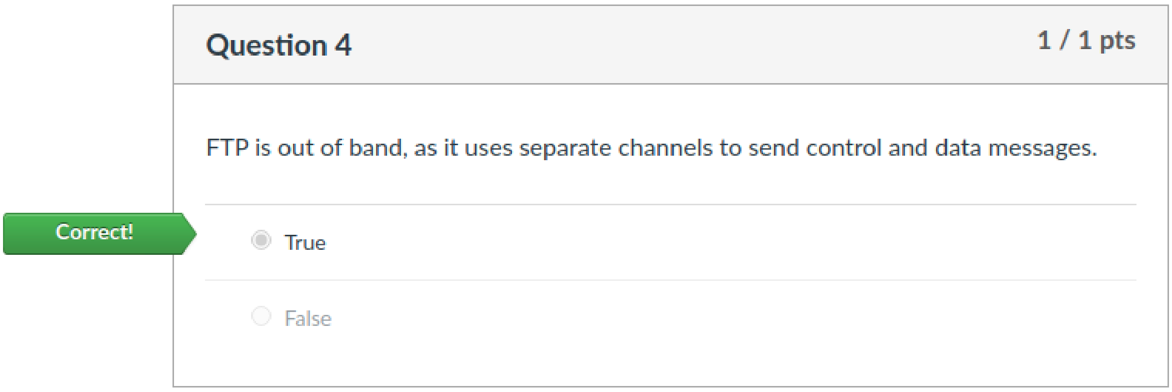

Question 3 1 / 1 pts

Since HTTP is stateless, it cannot offer personalized services for different users.

○ True

Correct! ● False

Explain: Use **cookies** to modify the services.

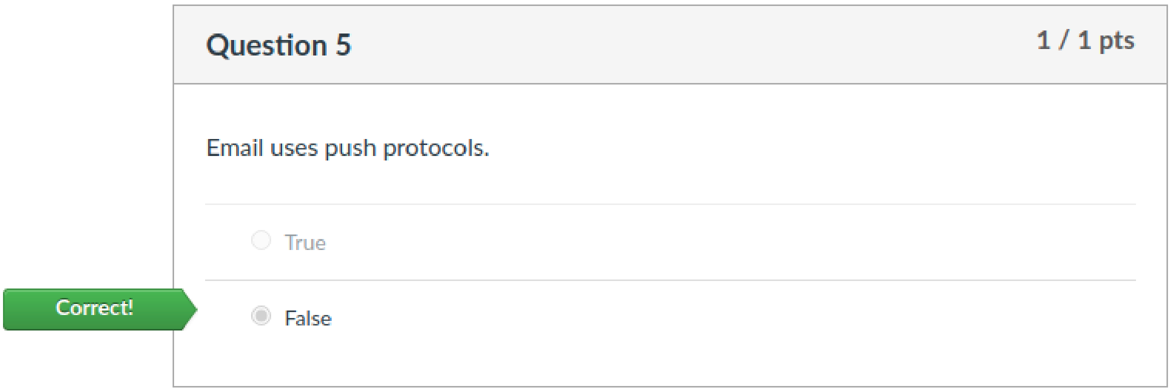

Question 4 1 / 1 pts

FTP is out of band, as it uses separate channels to send control and data messages.

Correct! ● True

○ False

Explain: In deed, FTP has two connections, one for control and one for data.



Question 5

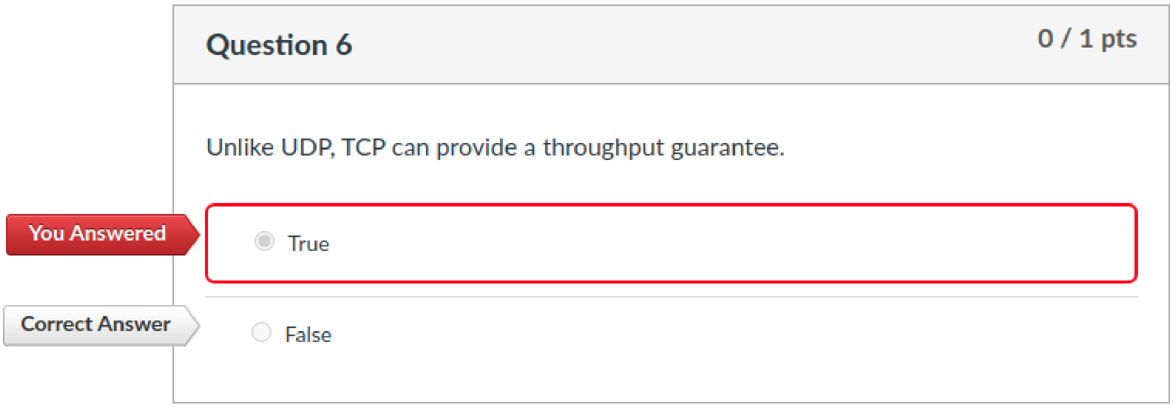
1 / 1 pts

Email uses push protocols.

○ True

Correct! ● False

Explain: Indeed, Email is based on push protocols (SMTP) and pull protocols (POP, IMAP, HTTP)



Question 6

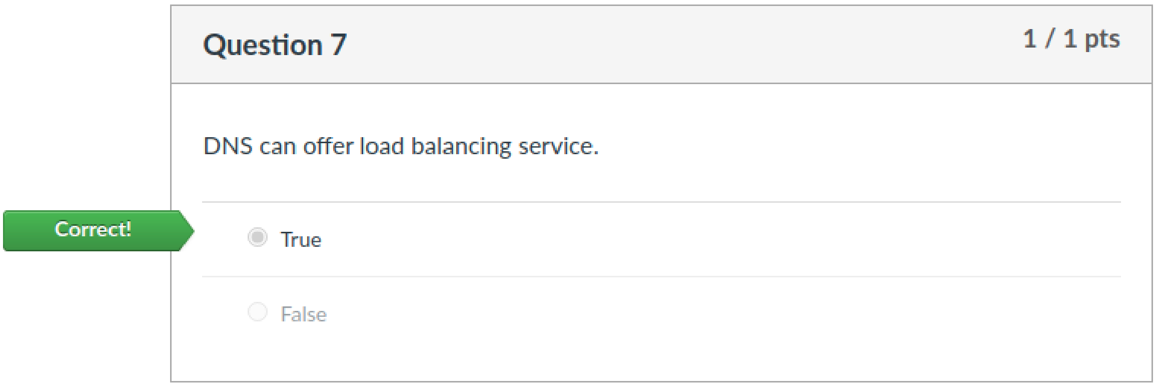
0 / 1 pts

Unlike UDP, TCP can provide a throughput guarantee.

You Answered ● True

Correct Answer ○ False

Explain: No transport layer protocols can guarantee throughput. TCP can only guarantee reliable data transfer. It is still possible (actually a large possibility) that TCP packets are kicked out by UDP packets.



Question 7

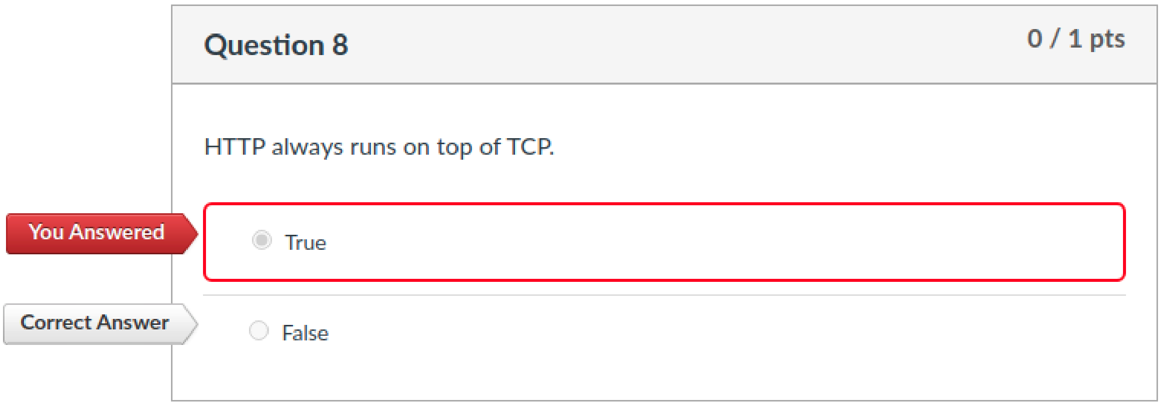
1 / 1 pts

DNS can offer load balancing service.

Correct! ● True

○ False

Explain: DNS can give different IP to different users in order to balance the load. Consider the GitHub has different IPs worldwide and these IPs can balance the load.



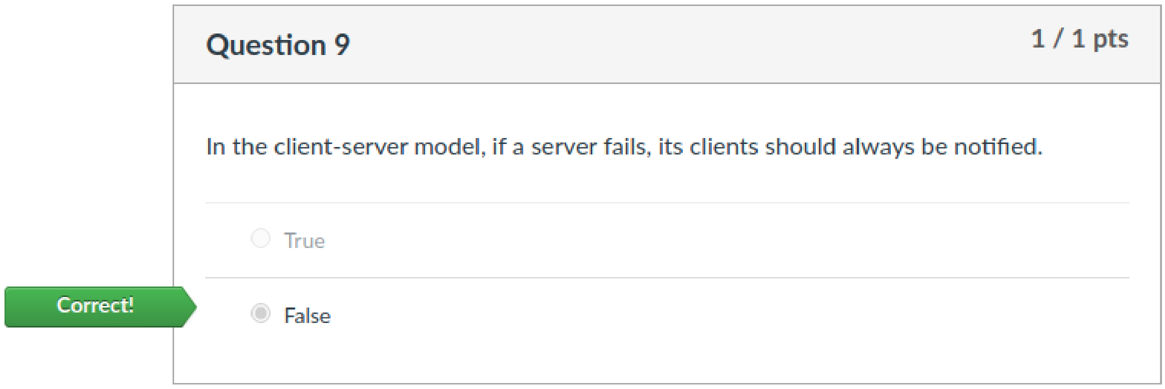
Question 8 0 / 1 pts

HTTP always runs on top of TCP.

You Answered ● True

Correct Answer ○ False

Explain: Google's HTTP3.0 (QUIC) uses UDP.



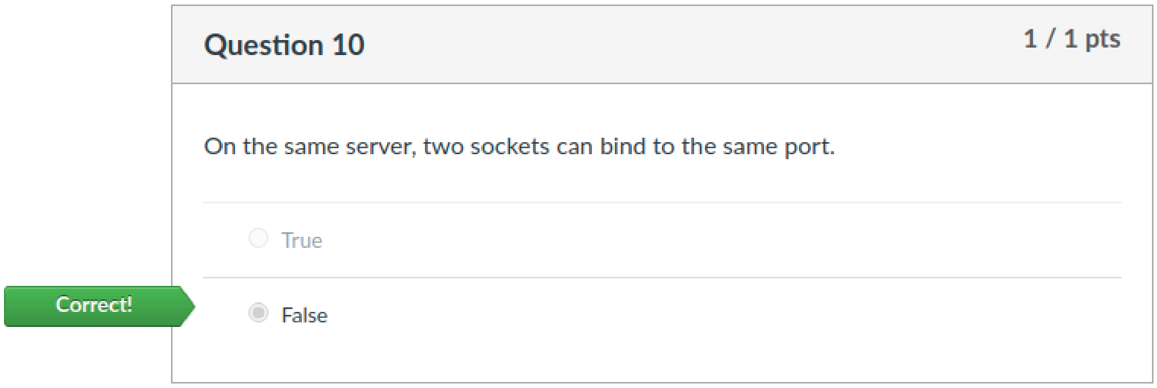
Question 9 1 / 1 pts

In the client-server model, if a server fails, its clients should always be notified.

○ True

Correct! ● False

Explain: Consider the existence of a load balancer. In this case, the client will be allocated with a new server and the client does not know this change.



Question 10 1 / 1 pts

On the same server, two sockets can bind to the same port.

○ True

Correct! ● False

Explain: In UDP, each process will have its own socket, and each socket has different server port number so that they can identify themselves. In TCP, each process will have its own socket. For any two sockets, either the source port numbers are different, or the destination port numbers are different. So they can not bind to the same port.