Challenges for distributed systems:

**Security:-**  
In distributed system critical information travels over network in form of network messages specially in banking web services, information like account number, credit card number, Social security number etc. get transmitted from one server to another while serving the client request. Content of message can be conceal. In addition identifying of the genuine remote users or Service provider is a challenge.

Denial of Service Attack: - Someone by sending huge number of requests without any real purpose to the service provider over web can impact the performance of the service provider.

**Scalability**: - A scalable service will remain effective when number of requests increase, The design of the Distributed system should be expendable and if service become poplar then scale should be expendable. So in web for any, if number of subscriber increased then attaching new hardware/physical resource become a necessity, or implementation design might need to change. For example implementation for file server may need to change for reducing access/ update time.

**Heterogeneity** :- A distributed network may consist of heterogeneous systems like different machine with different OS consist of different applications which may be implemented in  different programming languages, so it become important at multiple levels where two processes or applications should be able to communicate.

**Openness**: - Components of the distributed system should be expandable or reimplementable. So their key interfaces should be published and should comply with uniform communication mechanism.

**Failure Handing**: - few components of a distributed system may stop working at some point of time, in such situation services should not be get affected. Like a bank should have backup server ready in case main server stop working or system should be able to redirect requests.  
**Transparency**: - A distributed system should behave as a whole system, user should not be concern about the physical location of the servers, replication process, failure and design of the system.

**Concurrency:-** Distributed system receive concurrent request hence I should be able to do safe operations in such environment without harming data or loss of request.

Answer 3

**Service A**

**Omission Failure**: As messages can lost in network or not properly handled by Server hence server may exhibit omission Failure.

**Integrity and Validity Failure:** In case of duplication of messages server will execute Integrity Failure and messages can lost hence it may show validity failure.

**Arbitrary Failure**: Distorted Messages can reach to the server and server is not performing checksum on data packets, in such case service can exhibit arbitrary failures.

**Service B   
Omission Failure:** As messages can lost in network or not get properly handled by Server hence service may execute omission Failure.

**Validity Failure**: As messages can lost in network hence service can exhibit validity failure.

Service B is not reliable as it may show validity failure.

Answer 4: -

We will choose the time with 20ms rtt as it will give most accurate estimation of the time according to Cristian's method. Estimated actual time = t+ rtt/2  
10:54:28.342 + 10 = 10:54:28.352. If we know minimum round trip time for message (request) is 8 ms then accuracy will improve by +-2 ms.